Emergent Communication and Decision-Making in Multi-Agent Teams

Seth Karten

CMU-RI-TR-23-32

July 6th, 2023

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Katia Sycara, Chair
Fei Fang
Benjamin Freed

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Robotics.

Keywords: multi-agent reinforcement learning, emergent communication, human-agent teaming, information theory

To the windowless chamber, my muse and my cell,
Where Sylvia's echoes of darkness befell,
This dedication, etched with both pain and grace,
I offer, a testament to my battle's embrace.

Abstract

Explicit communication among humans is key to coordinating and learning. In multi-agent reinforcement learning for partially-observable environments, agents may convey information to others via learned communication, allowing the team to complete its task. However, agents need to be able to communicate more than simply referential messages about their observations. Agents must use communication to coordinate their actions to effectively accomplish their goals. This thesis argues that sparse emergent communication in multi-agent teams is essential for agents to encompass general decision-making prowess and fully reach potential in decentralized and social settings. First, I show the previous issues with emergent communication through the lens of interfacing between humans and groups of agents. Through human experiments, I find that humans learn to work best with agent partners which use discrete communication tokens with continuous encodings and a sparse message rate. An interpretability analysis shows that the tokens that work best with humans have the best representation capacity. Then, I investigate the usage of autoencoders to increase the representational capacity of observations. These results further confirm that sparser communication can be enabled without any loss of performance strictly based on intrinsic messaging objectives through mutual information and the information bottleneck. Lastly, I explore the development of language and communication through a social learning lens. In order to understand the minimal amount of communication, one needs to understand how communication may arise, especially in decentralized systems and teams where new agents are added without prior experience. Together, these techniques allow for sparse, intelligent communication between agents and groups of agents with a human partner with strong representational properties that allow for low empirical sample complexity and the potential to learn in social scenarios.

Acknowledgments

I am very grateful to my advisor Professor Katia Sycara for her encouragement and the opportunity to explore research topics from multiple perspectives, through a human-agent teaming lens as well as empirical and theoretical. I would also like to acknowledge Mycal Tucker, Siva Kailas, Soumith Udatha, Huao Li, Dana Hughes, Benjamin Freed, Michael Lewis, Julie Shah, and Fei Fang.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Machine learning is currently being applied, with generous results, to everything from healthcare to manufacturing, with various perspectives occasionally crossing from one discipline to another. Multi-agent systems are intriguing in the fact that they abstract individual decision-making such that they are able to be applied to many real problems, including microprocessor design [41], multi-robot pathfinding [20], vehicle rescheduling and multi-agent pathfinding [33], and internet packet routing [90]. Although, in order to fully test the capabilities and limitations of these systems, there is preference to apply testing in "toy" environments such as StarCraftII [69, 15]. One may easily argue that all problems are, in fact, multi-agent in formulation. Though, most practitioners abstract this and consider other agents in the system as part of the environment or deal with the combinatorial aspect directly, which have scaling issues. There are a set of disjoint communities that have studied multi-robot systems for communication: End-to-end observation-to-action systems have recently seen increasing success [47], The goal of this thesis is to address the combination of challenges in multi-robot systems, machine learning for end-to-end observation to action and communication policies, and provide enough generality to apply to arbitrary decision-making artificial intelligence problems.

The current machine learning paradigm requires the use of explicit supervision in order to maximize performance. Supervised learning requires carefully hand-labeled data, usually from a human expert-an expensive operation. Traditional planning shows that using explicit supervision in order to learn a complex action control it is unclear how to generate data for high-dimensional action spaces [36]. Due to combinatorial explosion, it is unfeasible for a human to collect this data. Rather, it is better to learn using hierarchical methods that use reinforcement learning [74] in order to learn how to perform complex multi-step tasks directly from experience. These methods allow for emergent behaviors directly from purposefully explored trajectories in the environment. Unsupervised learning has started to play a large role in reinforcement learning when the usage of direct reward is insufficient [17]. Rather, these methods use information theory [79] in order to provide a mutual information intrinsic reward between the latent spaces of the agent and encodings of desired future states.

How can we use recent advances in machine learning and artificial intelligence to allow agents to perform arbitrary decision-making with the help of tabula rasa communication in order to coordinate cooperative teams of agents, and in some cases, humans? Previous research has enabled cooperative multi-agent teams which may operate in partially observable environments through the use of multi-agent reinforcement learning [19, 45]. These agents learn to communicate through backpropagation in a team setting. Their messages are grounded in their observations. The key is to enable communication that is more than just communicating their observations. These messages need to provide insight regarding what action should be performed in order to induce cooperative behavior for arbitrary decision-making. We evaluate whether it is better learn an emergent communication language directly through task reward or in combination with recent advances in unsupervised learning and information theory.

A core area of this work is to understand interfacing between humans and groups of agents. Learning interpretable communication is essential for multi-agent and human-agent teams (HATs). In multi-agent reinforcement learning for partially-observable environments, agents may convey information to others via learned communication, allowing the team to complete its task. Inspired by human languages, recent works study discrete (using only a finite set of tokens) and sparse (communicating only at some time-steps) communication. However, the utility of such communication in human-agent team experiments has not yet been investigated. In chapter 3, we analyze the efficacy of sparse-discrete methods for producing emergent communication that enables high agent-only and human-agent team performance. We develop agent-only teams that communicate sparsely via our scheme of Enforcers that sufficiently constrain communication to any budget. Our results show no loss or minimal loss of performance in benchmark environments and tasks. In human-agent teams tested in benchmark environments, where agents have been modeled using the Enforcers, we find that a prototype-based method produces meaningful discrete tokens that enable human partners to learn agent communication faster and better than a one-hot baseline. Additional HAT experiments show that an appropriate sparsity level lowers the cognitive load of humans when communicating with teams of agents and leads to superior team performance.

As soon as one adds communication among agents, the most prevalent concern is to remove it. In multi-robot systems, there is latency between sending and receiving messages. Additionally, there are other constraints that warrant the prevention of communication to that which is only necessary. There many be adversaries listening. Learning *when* to communicate, i.e., *sparse* (in time) communication, and *whom* to message is particularly important when bandwidth is limited. However, recent work in learning sparse individualized communication suffers from high variance during training, where decreasing communication comes at the cost of decreased reward, particularly in cooperative tasks. We use the information bottleneck to reframe sparsity as a representation learning problem, which we show naturally enables lossless sparse communication at lower budgets than prior art. In chapter 4, we propose a method for **true lossless sparsity** in communication via *Information Maximizing Gated Sparse Multi-Agent Communication* (IMGS-MAC). Our model uses two individualized regularization objectives, an information maximization autoencoder and sparse communication loss, to create informative and sparse communication. We evaluate the learned communication 'language' through direct causal analysis of messages in non-sparse runs to determine the range of lossless sparse budgets, which allow **zero-shot sparsity**, and the range of sparse budgets that will inquire a reward loss, which is minimized by our learned gating function with **few-shot sparsity**. To demonstrate the efficacy of our results, we experiment in cooperative multi-agent tasks where communication is essential for success. We evaluate our model with both continuous and discrete messages. We focus our analysis on a variety of ablations to show the effect of message representations, including their properties, and lossless performance of our model.

Lastly, we explore the development of language and communication through a social learning lens. In order to understand the minimal amount of communication, we need to understand how communication may arise, especially in decentralized systems and teams where new agents are added without prior experience. Explicit communication among humans is key to coordinating and learning. Social learning, which uses cues from experts, can greatly benefit from the usage of explicit communication to align heterogeneous policies, reduce sample complexity, and solve partially observable tasks. Emergent communication, a type of explicit communication, studies the creation of an artificial language to encode a high task-utility message directly from data. However, in most cases, emergent communication sends insufficiently compressed messages with little or null information, which also may not be understandable to a third-party listener. Finally, chapter 5 proposes an unsupervised method based on the information bottleneck to capture both referential complexity and task-specific utility to adequately explore sparse social communication scenarios in multi-agent reinforcement learning (MARL). We show that our model is able to i) develop a natural-language-inspired lexicon of messages that is independently composed of a set of emergent concepts, which span the observations and intents with minimal bits, ii) develop communication to align the action policies of heterogeneous agents with dissimilar feature models, and iii) learn a communication policy from watching an expert's action policy, which we term 'social shadowing'.

1.1 Contributions

The goal of this work is to establish a reliable means of developing robust minimally communicating methods between groups of agents, humans and agents, as well as learning to communicate to coordinate when agents have already learned about their environments. This thesis makes the following contributions:

- Explore the interpretability of emergent communication and show its link to human workload in human-agent teams.
- Novel techniques to enable agents to learn to communicate tabula rasa through self-play.
- Show that the learned representation of emergent communication has a strong relationship with the sample complexity and overall performance of multi-agent teams.
- Show how to minimize communication to only causal messages (of the total messages) and reduce the learned size of the emergent communication messages to the least number of bits.
- Novel techniques for combining unsupervised learning with reinforcement learning in order to learn the best emergent messaging representation.
- Novel techniques for adapting prior message and action policies as part of social learning for emergent communication.

In addition to these conceptual contributions, the work has led to several research artifacts [38, 37, 35, 34] as well as open-source software available at `https://github.com/sethkarten`.

Chapter 2

Background

In this chapter, we provide relevant background on learning methods and problem setups used through the thesis.

2.1 Reinforcement Learning

Reinforcement learning is a learning paradigm in which an agent learns an action policy from interactions with the environment. Each interaction at a time-step produces a new state and reward. The series of states $s_t \in \mathcal{S}$, actions $a_t \in \mathcal{A}$ and rewards $r_t \in \mathcal{R}$ in an episode form a trajectory τ that makes up the episode. Reinforcement learning is not a learned fundamental law of the universe. Rather, the Bellman Equation [78] recursively defines the expected return of a state s to be the maximum for any action a of the expected reward for taking an action a in the state s plus a discounted value of the next state s'.

$$V(s) = \max_a (R(s,a) + \gamma V(s'))$$

In deep reinforcement learning, the idea is to perform informed random permutations to the action policy (from backpropagation). A deep neural network parameterizes the policy with network parameters θ. In order to maximize reward, the agent learns a policy $\pi(s_t;\theta)$, which is a mapping from state to actions.

2.1.1 Markov Decision Processes

Standard single-agent reinforcement learning abstracts other agents as part of the environment. Instead, multi-agent reinforcement learning understands the the necessity to model each agent individually in order to model their theory of mind [50, 51]. In cooperative multi-agent settings, we can model this as a decentralized, partially observable Markov Decision Process with communication (Dec-POMDP-Comm). Formally, our problem is defined by the tuple,
$\langle \mathcal{S}, \mathcal{A}, \mathcal{M}, \mathcal{T}, \mathcal{R}, \mathcal{O}, \Omega, \gamma \rangle$. We define $\mathcal{S}$ as the set of states, $\mathcal{A}^i$, $i \in [1, N]$ as the set of actions, which includes task-specific actions, and $\mathcal{M}^i$ as the set of communications for N agents. $\mathcal{T}$ is the transition between states due to the multi-agent joint action space $\mathcal{T} : \mathcal{S} \times \mathcal{A}^1, ..., \mathcal{A}^N \rightarrow \mathcal{S}$. Ω defines the set of observations in our partially observable setting. Partial observability requires

communication to complete the tasks successfully. $\mathcal{O}^i : \mathcal{M}^1, ..., \mathcal{M}^N \times \hat{\mathcal{S}} \to \Omega$ maps the communications and local state, $\hat{\mathcal{S}}$, to a distribution of observations for each agent. $\mathcal{R}$ defines the reward function and γ defines the discount factor. At every time-step, agents have the option to both send a message and execute an action in the environment.

2.1.2 Policy Gradient

One method of determining the gradient for backpropagation to train our deep neural network within the reinforcement learning paradigm is to use the policy gradient algorithm. The network outputs a probability distribution (usually via outputting a score and then normalizing it into a probability distribution through the softmax function $\phi(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$) and then acts in the environment by sampling from this distribution $a_t \sim \pi_\theta(a_t|s_t)$. The goal of any policy gradient algorithm is to maximize the total expected future reward during an episode given our policy:

$$\mathbf{E}_{\pi_\theta}[R(\tau)] = \mathbf{E}_{\pi_\theta}[\gamma^t r_t]$$

The first policy gradient algorithm, REINFORCE [91], uses the gradient of this function in order to determine the gradient update for the policy network:

$$\nabla_\theta \mathbf{E}_{\pi_\theta}[R(\tau)] = \mathbf{E}_{\pi_\theta}[\sum_{t=0}^{T} r_t \nabla_\theta \log \pi_\theta(a_t|s_t)]$$

2.2 Mutual Information

Not all reward functions may be specified. Our work considers using unsupervised objectives in order to create an intrinsic reward. Our work uses mutual information as the basis of deriving intrinsic rewards. Mutual information, denoted as $I(X;Y)$, looks to measure the relationship between random variables,

$$I(X;Y) = \mathbb{E}_{p(x,y)}\left[\log \frac{p(x|y)}{p(x)}\right] = \mathbb{E}_{p(x,y)}\left[\log \frac{p(y|x)}{p(y)}\right]$$

which is often measured through Kullback-Leibler divergence [42],
$I(X;Y) = D_{KL}(p(x,y)||p(x) \otimes p(y))$. By bounding the dependence between random variables, we can derive a Langrangian term for our loss function to optimize the intrinsic reward.

2.3 Communication and Lewis Games

A Lewis game [48] is a referential game between a speaker and a listener. In emergent communication, a speaker and a listener must learn to share a common referential language strictly through backpropagation. The difficulty of the Lewis game lies in the decentralized nature of the game. Without a common training signal to rate the utility of the language tokens, it may be difficult to converge to the same language. With this in mind, in the reinforcement learning setup, we either use a shared critic to act as a mediator to learn a similar language or use decentralized critics with the same reward function.

Chapter 3

Human Agent Teaming

3.1 Introduction

Multi-agent reinforcement learning (MARL) has been successfully applied in a variety of multiplayer games [25, 94], but trained agents often only collaborate well with humans in specific settings. For example, in StarCraft [65] and Dota 2 [7], teams of agents may be trained to compete against humans; or in Hanabi [73] and Overcooked [8], only a single agent may cooperate with a human. Prior art in training agents to collaborate in more complex settings that require communication among teammates (e.g., blindly crossing a traffic junction) has considered agent-only teams [77, 72], but translating such techniques to support human teammates, especially without human data during training, remains challenging [6].

Teaming with artificial agents is generally novel for humans; therefore, teaming requires a learning process for humans to adapt to new protocols for communication and collaboration. The dissimilarity of decision-making and lack of co-training [63] introduces a significant challenge for learning a shared 'language' for both humans and agents. A large body of research investigates the problems of translating natural language into a form usable by agents and generating intelligible replies in return. Our work takes the complementary approach of searching for ways to make it easier for humans to communicate with agents in their own language. Biases in human cognition are a product of both neural substrates [76] and evolutionary processes [40], which predispose humans to perceive and think in particular ways and not in others. Policies learned in the absence of human data, such as unsupervised interaction among agents, are less likely to conform to human biases than approaches that incorporate human inputs. Developers of AlphaGo, a family of Go playing programs that outperform human experts, found that while versions trained through supervised learning were superior in predicting the moves of human experts, versions trained exclusively through self-play were much better players [71]. These self-trained programs had perfect records against human opponents, found new joseki (corner sequences) unknown to human players [71], and were described by human players as "amazing, strange or alien" [9]. A sister StarCraft playing program, AlphaStar [89], followed the opposite tack by learning from replays of human matches and limiting speed and observations to human-like values [89]. In this case, despite reaching Grandmaster status (99.8%), the AlphaStar agents team could be defeated and was described as "like it is playing a 'real' game of StarCraft and doesn't completely throw

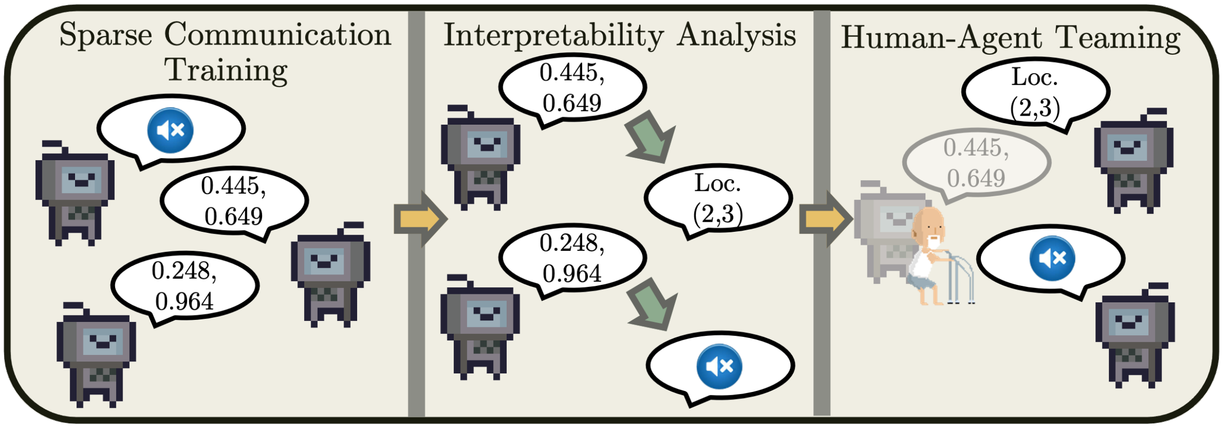


Figure 3.1: Our method has three phases: During sparse communication training, we train agents through self-play to learn an emergent message paradigm and to communicate infrequently according to a budget. In the interpretability analysis, messages are analyzed to determine the observations that they encode. Additional messages are removed in case they contain no information. Finally, in human-agent teaming, a ghost agent decodes the information and sends encoded messages for the human, who chooses the action.

the balance off by having unrealistic capabilities" [89]. Human difficulty in understanding behavior developed through self-play can be seen in even very simple games. Despite providing saliency maps and/or reward decomposition graphics for offering extra insight into the factors contributing to an action, observers could not predict the next action in Ms. Pac-Man [30] or a drastically simplified real-time strategy game [5] at better than chance levels.

During human-agent teaming, humans often struggle to understand the intent of agent partners, which is necessary for an effective partnership. Recent work has shown that bidirectional human-agent communication is sometimes required for peak performance in human-agent teaming [58]. While prior work in emergent communication establishes how agents may learn to communicate to accomplish their goals, the learned communication conventions often exhibit undesirable properties and fail to conform to human cognitive mechanisms. For example, emergent communication is often continuous, while humans communicate in natural language, which consists of discrete linguistic tokens or prototypes. Furthermore, humans learn from few examples and use compositional language that encompasses more complex meanings than was demonstrated to them [43]. Prior art has taken initial steps towards closing the gap between emergent and human communication: in agent-only teams in specific scenarios, discrete communication can perform as well as continuous communication [53], and in our prior work, we showed how learning discrete prototypes promotes robustness in noisy communication channels, as well as human interpretability and zero-shot generalization [81]. However, learning suitable discrete communication protocols that exhibit high performance in achieving team goals in sequential team decision-making remains a big challenge.

A recent meta-analysis [59] found that communication quality (rating as "effective and clear") had a significantly more substantial relationship with performance than the frequency of communication in human teams. In fact, excessive communication can present problems both in recognizing critical information and in remembering it. Recognizing an actionable message from a background of irrelevant ones is a classic signal detection problem [24] for which misses are known to rise as the number of irrelevant messages increases. Cognitive load theory [84] suggests that a large volume of communication may also interfere with the process of transferring information from working memory to long-term memory, which creates difficulty in remembering previously received information accurately. Thus, high frequency/high redundancy communication is likely to harm humans' abilities to recognize and learn from 'high quality' communication.

As an attempt to make agent communication akin to human communication in agent self-play, sparsity constraints have been shown to reduce the total amount of communication. Sparsity aims to minimize the number of unnecessary communications between agents to adhere to bandwidth constraints. Previous methods attempt to minimize the total communication while minimizing the loss of performance [90]. However, achieving this is very challenging: works that use gating (a function that determines whether to pass a message or not) often exhibit high variance and tend not to converge to the optimal communication budget [3]. Our Enforcers scheme builds upon gating methods but reduces variance sufficiently to converge to the optimal gating value by using a soft threshold.

To the best of our knowledge, this is the first work investigating the intelligibility of agent-generated communication models and their effects on performance in human-agent teams. We train our agents to learn human interpretable, discrete prototypes to interpret a message's intent by the receiver. The agent self-play method also constrains communication and considers the effects of different communication budgets on communication robustness and team performance. We develop an emergent communication interpretability scheme to transition from an agent-only communication space to a human-interpretable interface. In human-agent communication experiments, we explore the efficacy of humans learning to communicate with agents using discrete prototypes compared to a discrete 1-hot representation.[1]

Human-agent teaming experiments test two hypotheses. The first is whether humans can learn to communicate in Human-Agent Teams (HATs) using the emergent discrete prototypes from agent self-play. This is tested with single-agent HATs in the Parent and Lost Child environment, a variant of Predator-Prey. The second hypothesis is that an appropriate level of sparsity in communication reduces the cognitive load of human teammates and leads to better team performance. This is tested with multi-agent HATs in a blind Traffic Junction environment.

Our work additionally analyzes the effects of sparsity at various communication budgets to find the optimal minimum budget for human-agent teams through cognitive load and task performance.

Our contributions are as follows:

- We propose a novel MARL method that uses an interpretability analysis to produce sparse-discrete communication in HAT. We evaluate its efficacy in human subject experiments in HATs with (a) single human-single agent in different roles and (b) single human-multiple

[1] A one-hot is a group of bits among which the legal combinations of values are only those with a single high (1) bit and all the others low (0). In statistics, dummy variables represent a similar technique for representing categorical data.

agents.

- Our results indicate that humans can learn the emergent discrete prototype ‘language’ generated by agent self-play faster as compared to a baseline.
- This work is the first that focuses not only on learning human interpretable communication but, most crucially, on the effects of learned sparse communication in MARL on HAT *performance and human workload*. In particular, we expand previous findings about communication sparsity from human-human teams to human-agent teams. Our results show that an appropriate frequency level leads to the best team performance and the lowest cognitive load of human teammates.

3.2 Related Work

Our work studies multi-agent reinforcement learning at the intersection of research in sparse and discrete emergent communication, evaluated in the context of human-agent teams.

3.2.1 Multi-Agent Reinforcement Learning

Multi-agent reinforcement learning studies a team of agents working to maximize shared performance on a task. By learning to communicate through backpropagation, agents can learn task-specific coordination directly from error derivatives. Typically, a centralized training, decentralized execution paradigm enables agents to learn from privileged information but act independently [19]. Instead, we use a fully decentralized training setup with shared parameters to foster faster training in cooperative tasks [77].

3.2.2 Sparse Communication

In the context of multi-agent systems, sparse communication necessitates limiting the total communication exhibited between agents, measured by the number of bits sent. Sparse communication has been explored by learning a communication gate, learning whom to target, and compressing communication tokens. Gating methods use a neural network layer to learn a gating function to decide whether to pass through the message [57, 88]. Targeting methods learn who to send messages and/or who would be receiving messages [11, 1, 23, 39]. Information bottleneck methods attempt to minimize the entropy of messages between agents to learn whom to target communications with a centralized communication targeting system [68, 90]. These methods try only to remove unnecessary communication, but they have been shown to decrease the overall reward, which leads to suboptimal task performance. Targeting methods work as a complementary model to gating and information bottleneck methods. Thus, this may be used as an additional model to reduce communication further. All these methods only have one suboptimal budget to reduce communication. Compression is enabled by limiting the size of communication tokens to a fixed size or a bitwise encoding [70, 20] though recent work has shown that continuous encodings can contain more information [81]. Our work builds upon gating methods but reduces variance sufficiently to converge to the optimal gating value by using a soft threshold. We are able to converge to various communication budgets, allowing our models to learn communication conventions that complement human preferences while maintaining multi-agent system performance. This chapter also analyzes the performance of sparsity in human trials.

3.2.3 Interpretable Communication Formats

Regardless of communication frequency, researchers have developed a variety of communication formats in an effort to improve human interpretability. For example, inspired by the discrete nature of words in human language, one work discretizes emergent communication by forcing agents to communicate via one-hot or binary vectors [55]. Other works focus on the compositionality of tokens to create simplified "sentences" [61] or train agents to communicate directly via natural language [66, 12, 2]. Unfortunately, agents trained by such techniques often perform worse in human trials than in multi-agent teams, indicating that the communication interpretability remains limited [45]. Other work decides the interpretation of messages based on the effect on a human listener [6]. In our work, we train agents to communicate via a discrete set of tokens in a continuous space [81]. While maximally informative to the agents, these sets of tokens may conflate intent with location or relations among items in correlated but indirect ways. Our experiments test the degree to which discretization and sparsity can overcome potential cognitive incompatibilities and allow humans to exploit the semantic richness of agent token sets. In our single-trial experiments, participants must learn the semantics of tokens while using them to accomplish their joint tasks. The one-hot encoding provides an isomorphic mapping, known to be compatible with human cognition, but must be learned token by token as a paired-associate task. By contrast, embedding spaces learned by the agents provide a relational structure among tokens, reducing the learning that must occur [80].

3.2.4 Human-Agent Teaming

Most human-agent teaming has focused on adapting policies to allow humans and agents to understand their partners' intent, which is necessary for effective partnership. Both human and agent adaptation benefits team performance [49]. Some agents are able to recognize changes in human intent given observations of their actions. The agents can then adapt their policy to coordinate better with the human [29]. Agents have also been trained to learn the effects of their actions on other agents. When combined with communication, this has led to increased coordination of agent-only teams [31]. These works assume agents will adapt to their teammates' policies by observing their actions. Instead, given the importance of bidirectional human-agent communication in team performance, we focus on enabling humans and agents to also adapt based on *communication* [58].

Recent works have begun to explore communication in human-agent teams. In agent self-play and human-agent teaming, sharing the intent or goal was most useful towards increasing performance [52]. However, this study only involved tasks with simplistic agents who follow hard-coded randomized paths, in which communication was not required to successfully complete the task. Other work has built upon this by introducing tasks that may require communication. They found that communicating both beliefs and goals, while minimizing communication frequency, improved human-agent teaming [86], but worked with simplistic agents that are given

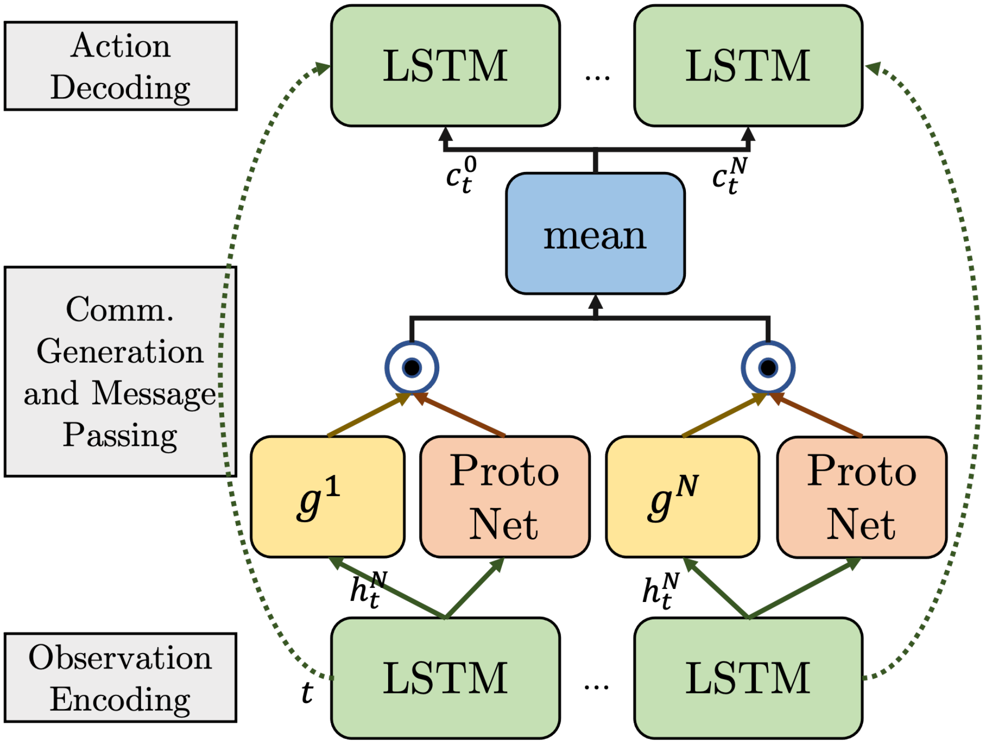


Figure 3.2: Above is the Enforcers architecture, including the MARL with communication pipeline.

human-designed observation information to complete their task. They then learn to recognize when they need additional information and learn to communicate from a list of predefined goals. In our work, we use state-of-the-art agents to team with humans using *emergent communication*. Our agents learn both communication and action policies. In our benchmarks, communication is necessary to behave optimally, including cases when a single human must coordinate with multiple agents.

3.3 Agent Self-Play

In this section, we introduce the Enforcers, a curriculum of constraints necessary for enabling stable multi-agent sparse-discrete emergent communication. Agents are trained exclusively with other agents. In agent-only experiments, we show that our models can perform competitively with respect to other agent-only models on benchmark multi-agent tasks and environments that utilize communication. Additionally, our method can train agents to adhere to various communication budgets from maximum communication (100% of the time) to the optimal communication budget for a task (e.g., only communicating 10% of the time), which is assessed in benchmark environments: Predator-Prey and Traffic Junction.

	Traffic Junction			
	Easy		Medium	
Model	Convergence Epoch	Success %	Convergence Epoch	Success %
Fixed-Cts	101	.993	675	.997
Fixed-Proto	199	.993	927	.959
Gated-Cts	652	.968	1320	.926
Gated-Proto	1262	.977	1518	.920
Enforcer-b^*	+35	.983	+196	.947

Table 3.1: **Traffic Junction:** In two Traffic Junction environments, we compared the convergence epoch and success rate for fixed (at all time-steps) vs. gated (sparse) communication and continuous vector vs. prototype-based (discrete) tokens. Prototype-based agents achieve a similar success rate to continuous communication agents. The Enforcer method can stably maximize success while using optimal communication b^* and discrete prototypes with a few additional epochs from Fixed-Proto.

	Predator-Prey Cooperative	
	5×5, N=3	**10×10, N=5**
Model	Average Rewards	
Fixed-Cts	2.25	7.64
Fixed-Proto	2.18	7.13
Gated-Cts	1.38	6.51
Gated-Proto	0.94	5.69
Enforcer-b^*	2.07	7.22

Table 3.2: **Predator-Prey:** In two cooperative predator-prey environments, we measured the team reward for agents using fixed vs. gated communication and continuous vector vs. prototype-based (discrete) tokens. Unlike naive gating approaches, the Enforcer method stably maximizes reward while using optimal communication b^* and discrete prototypes.

3.3.1 Problem Setup

We formulate our multi-agent problem as a decentralized, partially observable Markov Decision Process with communication (Dec-POMDP-Comm). Each agent or human receives a partial observation of the environment, so as a team, they must learn to communicate essential information to complete the task adequately. Additionally, we require *sparse* communication: each agent must minimize total communications according to a communication budget b. First, define B as the total number of bits communicated if an agent emits a communication vector at each time-step. We define $b = \frac{B}{t_\delta}$ as the average number of bits communicated over any contiguous subset of the episode t_δ. For ease of analysis, we define $B = |\mathcal{T}| * |c|$ as the length of the episode times the size in bits of the communication vector, which makes $b \in [0, 1]$. We encode this into the reward function in section 3.3.2. Our problem is formulated by the tri-objective of discovering the optimal constrained communication-action policy. That is, the agents must learn to

Soft Enforcer (Comm % \ Success %)		
Budget	**Easy**	**Medium**
100	- \ .993	- \ .959
90	.889 \ .983	.883 \ .955
70	.781 \ .986	.682 \ .946
50	.588 \ .980	.445 \ .947
30	.282 \ .983	.261 \ .931
20	.195 \ .959	- \ -

Table 3.3: **Traffic Junction:** The table above shows the training results using the Enforcers with various budgets b. The fraction of total communication \ success rate is compared for each imposed budget. The method is able to yield consistent performance while an optimal budget is observed.

1) communicate effectively, 2) act effectively, and 3) obey communication sparsity constraints. Communications occur at discrete, uniform time-steps.

Formally, our problem is defined by the 8-tuple, $(\mathcal{S}, \mathcal{A}, \mathcal{C}, \mathcal{T}, \mathcal{R}, \mathcal{O}, \Omega, \gamma)$. We define $\mathcal{S}$ as the set of states, $\mathcal{A}_i$, $i \in [1, N]$ as the set of actions, which includes task-specific actions, and $\mathcal{C}_i$ as the set of communications for N agents. $\mathcal{T}$ is the transition between states due to the multi-agent joint action space $\mathcal{T} : \mathcal{S} \times \mathcal{A}_1, ..., \mathcal{A}_N \rightarrow \mathcal{S}$. Ω defines the set of observations in our partially observable setting. Partial observability requires communication to complete the tasks successfully. $\mathcal{O}_i : \mathcal{C}_1, ..., \mathcal{C}_N \times \mathcal{S} \rightarrow \Omega$ maps the communications and state to a distribution of observations for each agent. $\mathcal{R}$ defines the reward function and γ defines the discount factor. We build on the objective in [81], in which we aim to maximize the total expected reward of all agents, as follows,

$$\max_{\pi:\mathcal{S}\rightarrow\mathcal{A}\times\mathcal{C}} \mathbb{E} \sum_{t\in\mathcal{T}} \sum_{i\in N} \gamma\mathcal{R}(s_t, a_t)$$

such that, $(a_t, c_t) \sim \pi$, $s_t \sim \mathcal{T}(s_{t-1})$.

We use REINFORCE [91] to train both the gating function and policy network subject to the previous constraints. In order to calculate the information similarity, we compute loss by comparing each agent's decoded state against the entire state but enforce decentralized execution and testing.

3.3.2 Methodology

The MARL with communication pipeline consists of an observation encoding step, a message passing step, and an action decoding step. Similar to IC3Net [72], each agent uses a recurrent encoding and decoding step. However, our model novelly addresses the communication generation and message passing step. As depicted in Fig. 3.2, during each time-step, each agent receives an observation, which is passed into each agent's LSTM[2]. The hidden state is passed to the gating function and the prototype network. Rather than a probabilistic gate in IC3Net,

[2] Long Short-term Memory is a Recurrent Neural Network (RNN) architecture

the Enforcers' gating function decides whether to pass a message to other agents based on the latent state information. The prototype network receives continuous hidden state information. The prototype network encodes and compresses the relevant observation and intent/coordination information into one of a discrete set of emergent prototype vectors. We then take the Hadamard product between the gating value and discrete prototype vectors, which masks communication output according to the gating value. These communication messages are passed to other agents, where each agent takes the mean of the messages received. This value is passed to an agent's LSTM, which, finally, produces an action.

The construction of the communication message is similar to an unsupervised learning objective, which determines useful latent information to pass to other agents based on the policy loss gradient. Due to the high variance of dual communication and action policy learning, a communication curriculum is applied. A success criterion must be achieved at each level of the curriculum to advance to the next phase. Let λ define some hyperparameter. The communication curriculum changes the reward,

$$R^i = R^i_{env} - \lambda R^i_{comm},$$

at each phase, where R^i_{comm} is defined during each phase. During the "Open Gate" phase, communication is hard-coded to always occur and $R^i_{comm} = 0$. After achieving the success criterion, the "Positive Communication Reward" phase rewards communication with $R^i_{comm} = |1 - c|$. This phase reduces instability when transitioning from an open gate to a learned gating function. After achieving the same success criterion with the new constraint, the Enforcers phase is implemented.

The Enforcers apply a soft constraint to the reward function to ensure that the communication is within a budget. The communication reward penalty's main purpose is to constrain communication for the total budget and prevent communication bursts. This helps the reward-based sparse method achieve a particular sparse budget. Define a reward penalty proportional to a scaled distance to the proposed fraction of the total budget b for some observed fraction of total communication c,

$$R^i_P = \begin{cases} \frac{b-c}{b} & c \leq b \\ \frac{b-c}{1-b} & c > b \end{cases},$$

We also incorporate a first-order derivative term,

$$R^i_D = R^i_P - R^{i-1}_P,$$

and an integral term,

$$R^i_I = \sum_{j=0}^{i} R^j_P,$$

to the communication penalty,

$$R^i_{commSoft} = \lambda_P R^i_P + \lambda_D R^i_D + \lambda_I R^i_I.$$

The hyperparameters are tuned empirically. Note that the method limits the integral term such that $|R^i_I| < K$ for some hyperparameter K for stability. Even with our optimizations, we find reward-based sparsity methods to be high variance. We find that the hyperparameter ranges are binary in that only carefully determined ranges work. Outside these ranges, there is no convergence to the sparse budgets. We use $\lambda_P = 1., \lambda_D = 1.6, \lambda_I = 0.026, K = 50$.

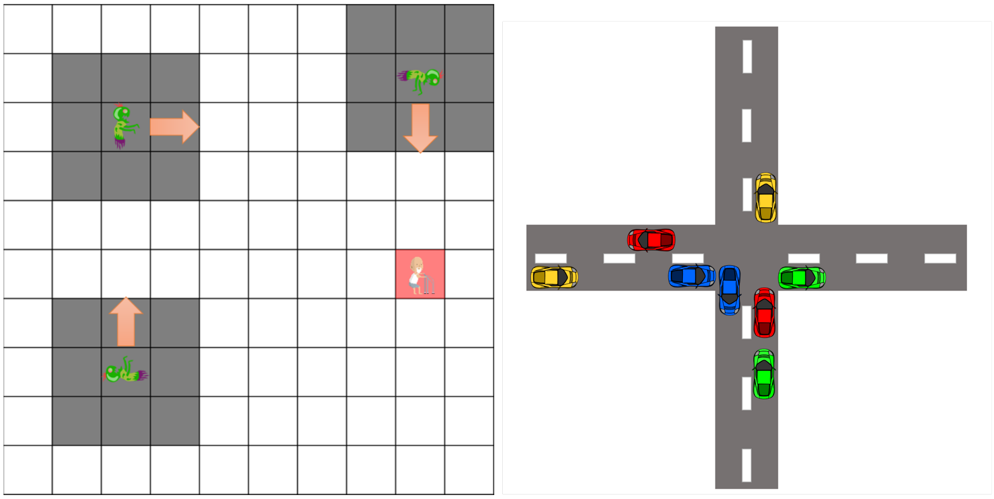


Figure 3.3: **Left:** 10×10 Predator-Prey. The human in the red square denotes the prey, while the remaining entities denote the predators. The arrows denote the actions taken by each predator, and the gray shading denotes the vision of the predators. **Right:** Two-lane bidirectional traffic junction. The agents enter the junction, driving on the right lane from any of the four entrances randomly. The agent's path entails either going straight at the intersection, turning left at the intersection, or turning right at the intersection. The agents then directly proceed to exit the intersection.

3.3.3 Benchmark Agent-Only Experiments

Experimental Setup

We tested the Enforcers scheme in Predator-Prey and Traffic Junction benchmark environments [72]. In Predator-Prey, a precursor to the Human and Lost Child scenario, N predator-agents search for one prey agent and then travel to its location. This is a fully-cooperative scenario, so ideally, the prey learns to communicate its location to allow for optimal navigation of the predators. Predator and prey agents each have a uniformly random chance of spawning in any cell in the grid at the beginning of the episode before searching for T time-steps (5×5: $T = 20$; 10×10: $T = 40$).

In Traffic Junction, up to 10 agents navigate a two-lane bidirectional traffic junction as shown in Fig. 3.3. The agents are unable to observe each other; they are *"blind"*, so they must communicate to avoid collisions both in the junction and within a given lane (e.g., if the front agent brakes). Agents are spawned in the environment with probability p at each time-step for a fixed total number of time-steps T (easy: $p = 0.1$, $T = 20$; medium: $p = 0.05$, $T = 40$).

Results

Setting a budget with the Enforcers, the model can converge to various communication budgets, including the optimal communication budget b^*. Performance is evaluated over three seeds. The Enforcer method is able to yield performance equivalent to unconstrained methods, i.e., continuous communication, while exhibiting discrete-sparse properties. Table 3.1 and table 3.2 show

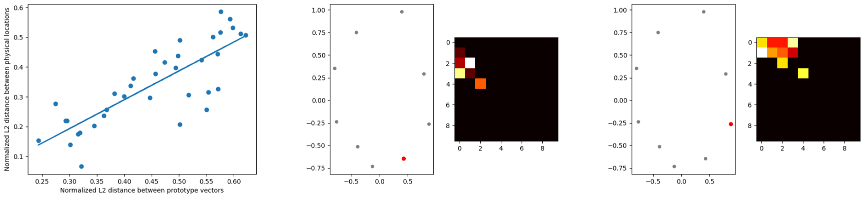


Figure 3.4: The distance between a pair of prototypes was highly correlated with the distance between the locations the prototypes referred to (left plot). A single prototype referred to a cluster of nearby locations (middle and right plots), with lighter colors denoting more frequent locations. Here, the two prototypes were both in the bottom right of the 2D PCA of the communication space and both referred to locations in the upper left of the grid.

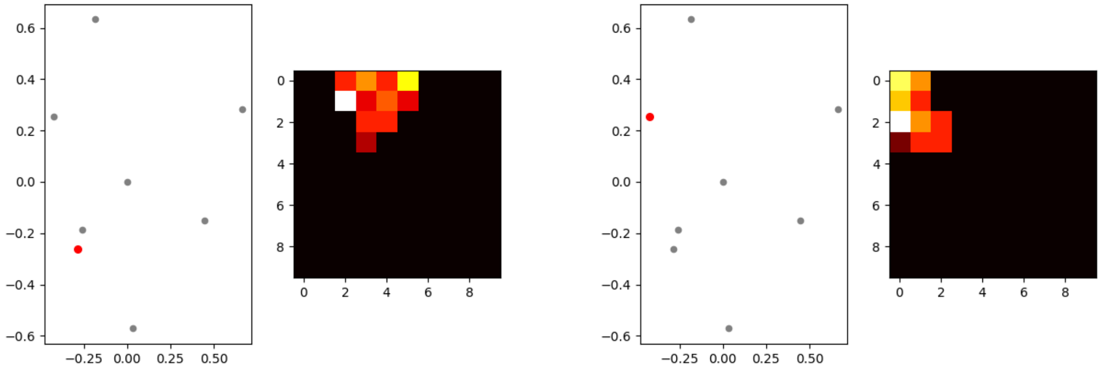


Figure 3.5: 1-hot communication vectors show no correlation between prototype locations and environment locations.

that our method, **Enforcer**-b^*, is able to converge to a high success rate with few additional training epochs after introducing the Enforcer constraints on the Fixed-Proto method. Table 3.3 shows that the method can constrain communication until an optimal budget threshold with nearly no change in reward. Overall, the Enforcer method is able to provide high task performance at various communication budgets with limited additional training.

3.4 Agent Interpretability

This section discusses the analysis of the properties that we observed in the agent training experiments and deem useful for learning the prototypes and sparsity of communication. Using domain knowledge of the task, we can deconstruct the prototypes to understand latent space communication. We performed a 2D Principal Component Analysis (PCA) of the prototypes for each environment and task based on the agents' observations while communicating their respective messages.

3.4.1 Setup

Our purpose of interpretability is to find a post-hoc analysis of the emergent message paradigm learned in self-play. We analyze the degree to which we can determine the white box meaning of learned messages. We compare 1-hot encodings with prototype encodings. We analyze the learned messages in the Medium Traffic Junction scenario as described in section 3.3.3 and the Parent and Lost Child scenario, which is a single predator and single prey subcase of Predator-Prey as described in section 3.3.3. Simply, or analysis aims to solve the follow question: What are the properties of the different emergent communication policies that we foresee being most useful in HAT?

3.4.2 Methodology

From intuitive and mathematical perspectives, we analyze how agents convey observation information using prototype or one-hot messages. First, we visualize associations between prototypes and agent locations (a subset of observation information), as shown in the rightmost figures in Fig. 3.4. The eight gray dots and one red dot denote the 2D principal component analysis (PCA) of the 9 prototypes that the agent used. The red dot denotes a particular prototype; the heatmap to the right of the PCA plot shows which agent locations were observed when the agent emitted the denoted prototype. Intuitively, these heatmaps show the meaning of each prototype. Additionally, we analyze the relationship between the Euclidean distance between vectors and the information that they contain, e.g., in Parent and Lost Child the information is environment locations, in order to determine any correlation. Ideally, we want to see structured interpolation between information and messages in the latent space to prevent null messages, which contain ambiguous or no information.

3.4.3 Results

Parent and Lost Child

Property 1: Relational Observation Encodings. We found that the learned prototypes exhibited several desirable characteristics. First, only 9 prototypes were used, indicating that the agents divided the 9×9 grid into coarser areas. Second, each prototype referred to a distinct patch of the grid. Lastly, and most interestingly, prototypes that were close together in communication space referred to nearby locations in the grid. For example, the two visualized prototypes in Fig. 3.4 are both in the bottom right of the communication space, and both refer to grid locations in the upper right. We confirmed this anecdotal evidence by finding that the distance between prototypes and the distance between the locations the prototypes referred to was tightly correlated (with an $r^2 \approx 0.5$).

While one-hot based communication also successfully represented grid locations, as shown in Fig. 3.5, it was fundamentally unable to exhibit a similar correlation as found between environment location and prototype location with prototypes. All one-hot vectors are equally far apart by definition, so one cannot predict grid distances as a function of distance between communication vectors ($r^2 = 0$). Furthermore, given that all one-hot vectors are orthogonal, conducting PCA on the vectors does not provide any information on distance relations.

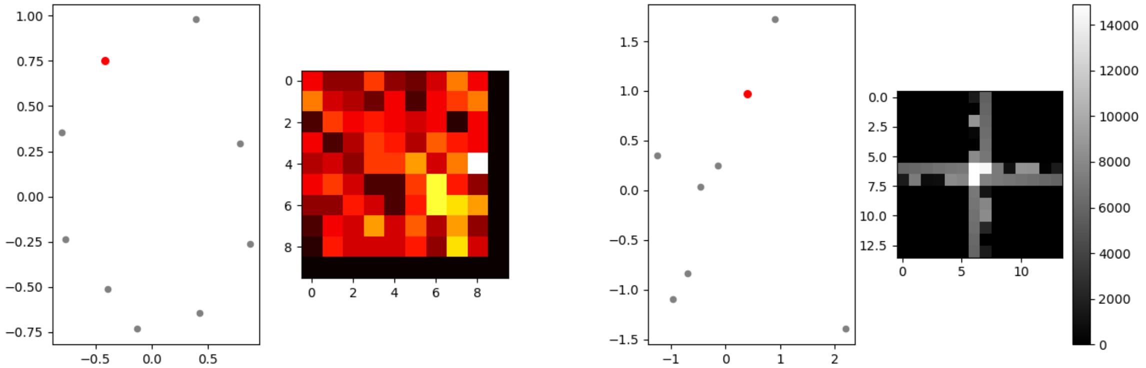


Figure 3.6: Recurrence with learned prototypes leads to null prototypes for communication. That is, it conveys no information to the other agent. These communications are unnecessary and show that sparsity is feasible. Left: Human and Lost Child. Right: Traffic Junction.

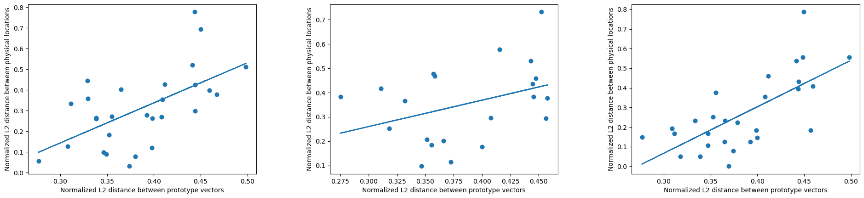


Figure 3.7: Above shows the correlation for prototypes in Traffic Junction with full communication (left), medium sparsity at 50% communication (middle), and optimal sparsity at 30% communication (right).

Traffic Junction

Property 2: Sparse Communication Through Information Content. Results from prototype-based communication vectors yield that they encode intrinsic information of the environment locations. Fig. 3.7 shows the correlation of the Euclidean distance between each prototype vector to each other and the corresponding Euclidean distance in the environment. The slope for the full continuous communication, 50% communication (mid-sparse), and 30% communication (max sparsity) are 1.94, 1.09, and 2.36, respectively. The emergent prototypes tend to have multiple location mappings per distinct prototype. This results from the prototypes often representing information similarly per lane instead of uniquely for each environment location.

Since navigating the traffic junction "blind" is a complex task, communicating the location alone is insufficient information to avoid a collision. One may think that the agent's intent is also necessary to interpret. A car may communicate its location, but it is useful to know if they are going to accelerate (move to the next cell) or brake (stay in the same cell). However, the agents communicate the same prototype token regardless of the action that they take, which implies that the agents rely on recurrent and coordination information within their communications. This follows from homogeneous action policies among agents.

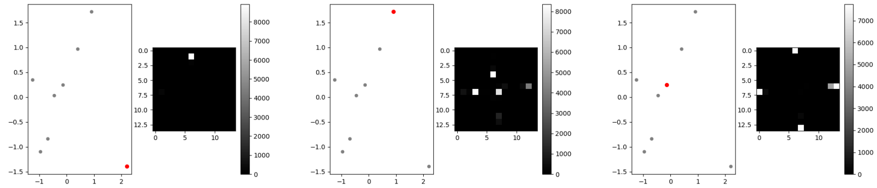


Figure 3.8: Above shows the Traffic Junction locations (right figures) represented by their corresponding prototype as shown in a principal component analysis (left figures).

Recurrence and other coordination information play a large role for the agents in Traffic Junction. The agents tend to announce themselves when they enter the environment, but afterwards they communicate only at key points depending on the direction that they are traveling. Additionally, the agents repeat prototypes irrespective of the lane. See Fig. 3.8. This means that agents send less information later in their trajectory, implying that agents can remember important details from previous messages. In Fig. 3.6, the last figure shows that a null prototype is often used. This communication vector shows that, while the agents will still optimally traverse the junction, at least 30% of all communications may be gated (prevented from being communicated).

3.5 Human-Agent Teaming

This section discusses the setup and results of the human-agent teaming experiments. In previous sections, we introduced our multi-agent sparse-discrete emergent communication method and showed its effectiveness in agent self-play experiments. The next intuitive question is: would the communication method work when a human is swapped with one of the agents in the original problem setup? Although humans and MARL agents use entirely different communication systems, we deliberately design two human subject experiments with reasonable translations and approximations in order to evaluate our method in multiple human-agent teaming task scenarios. Specifically, two hypotheses are tested:

- **H1**: Humans can learn to communicate in a human-agent team using the emergent prototypes from agent self-play.
- **H2**: Sparsity in communication reduces the cognitive workload on a human teammate such that they are able to perform better at the task.

3.5.1 Human Experiment Design

Interpretability

The core of **H1** lies in the interpretability of communication tokens learned by MARL agents. If the emergent “language” is interpretable, then the human can effectively develop a mental model and use it to collaborate with agent teammates. Participants in the first human-agent teaming experiment teamed with MARL agents to solve the Parent and Lost Child task. Half of the participants use prototype-based communication, and the other half use one-hot communication. The first experiment aims to test whether humans can learn the mapping relationship between

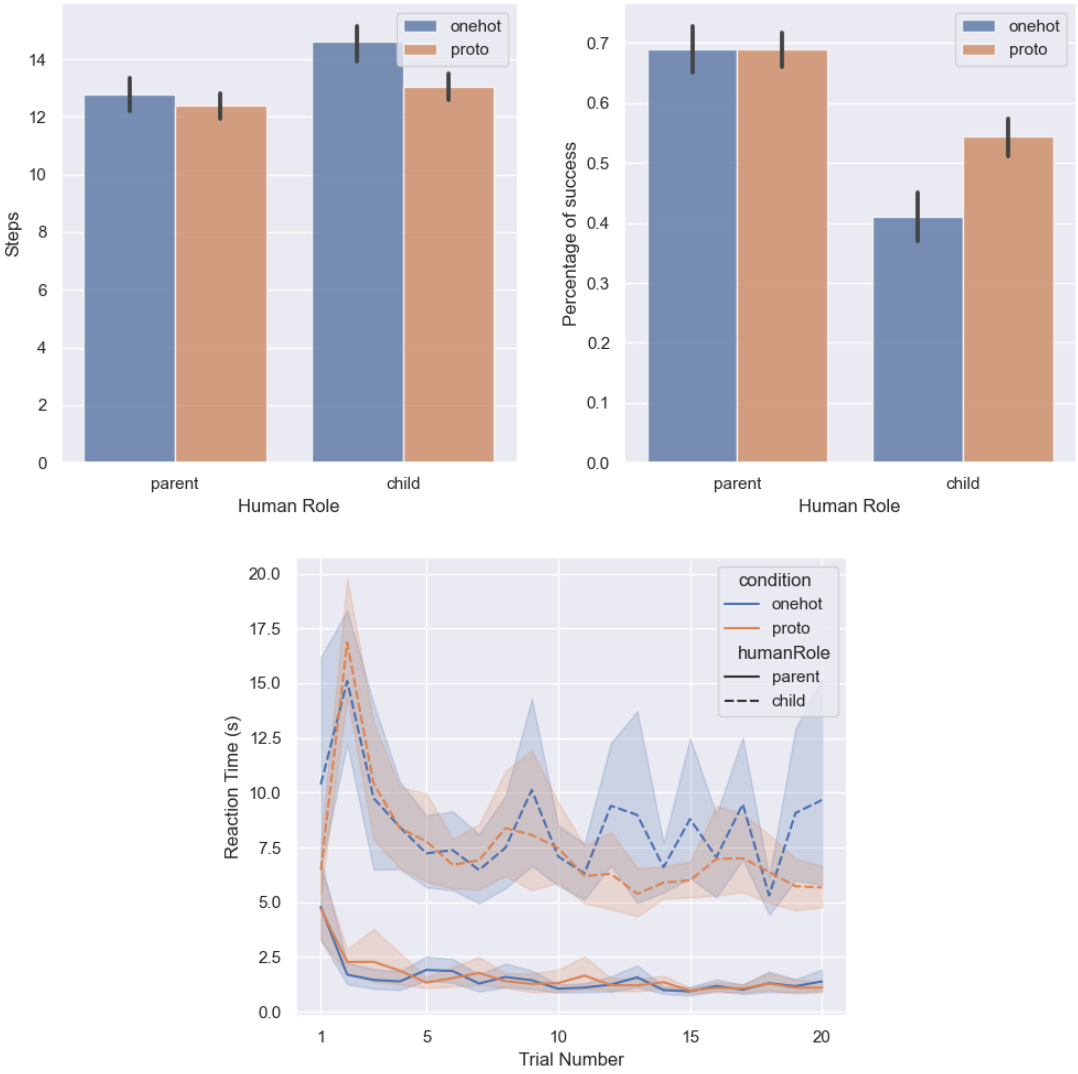
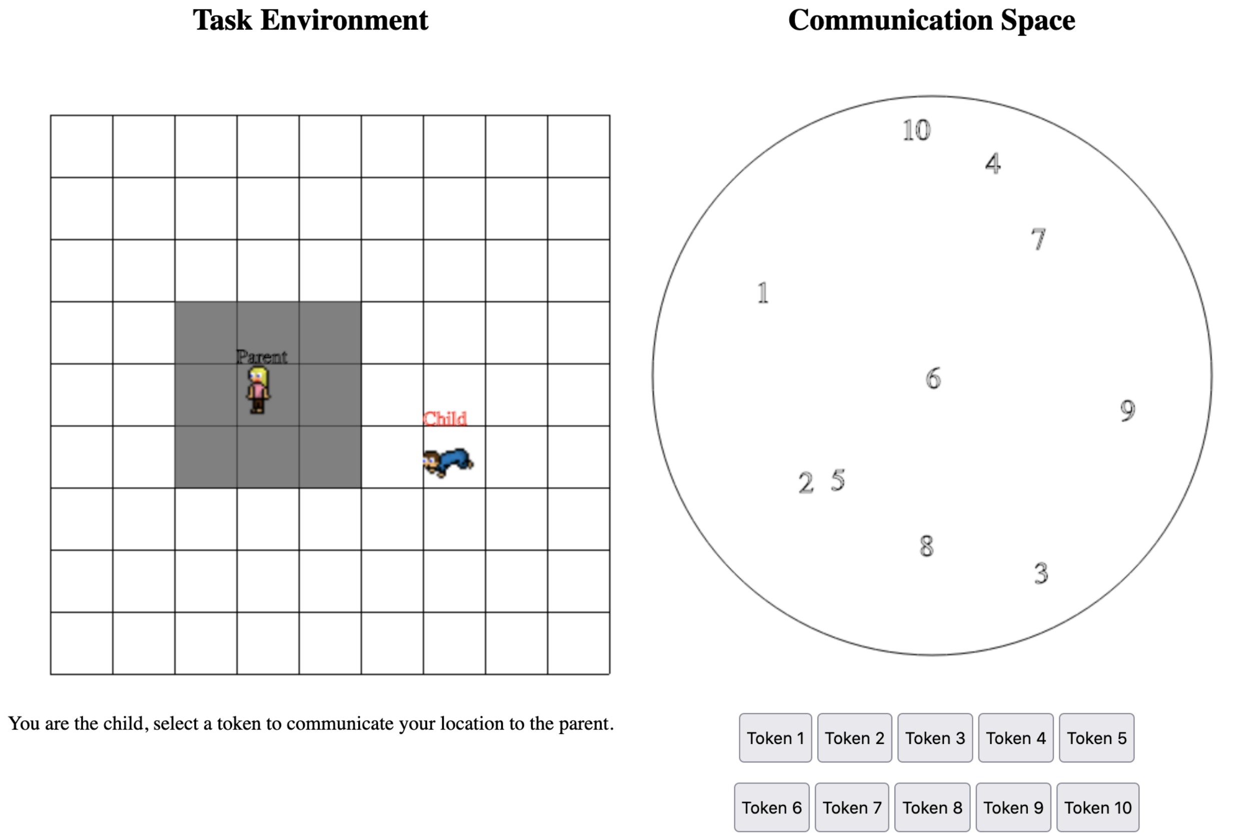


Figure 3.9: Learning curves of human-agent communication in the parent and lost child scenario. Top Left: Average steps taken per trial in one-hot and prototype conditions. Top Right: Percentages of completed trials by roles and conditions. Bottom: Reaction time learning curves of human participants in the child (dashed lines) and parent roles (solid lines). Shaded areas indicate 95% confidence intervals.



You are the child, select a token to communicate your location to the parent.

Figure 3.10: User interface of the human-agent communication experiment in the Parent and Lost Child scenario. The left panel displays the game environment state, including the locations and vision of the parent and child. The right panel consists of a communication space and selection buttons. Communication tokens are arranged in the 2D space based on PCA analysis to indicate potential semantic meanings.

tokens and child locations better than rote memorization and use it in a human-agent team task. The prototype vectors are translated into communication tokens and plotted on a 2D space using PCA as described in section 3.4. For 1-hot vectors, we use the same spatial arrangement with prototype tokens but randomize the token mapping between subjects. By doing so, we eliminate the confounding effect of spatial structure in human learning and can better focus on the core comparison between prototype-based and 1-hot communication.

Sparsity

Human-agent communication is usually limited by the divergent cognitive capacity of humans and agents in information processing. That is, humans are good at spatial, heuristic, and analogical reasoning while autonomous agents process high-frequency information continuously [18]. **H2** assesses if introducing sparse messaging helps humans process the information sent by agents and lower the overall cognitive workload. We evaluate this hypothesis in the Traffic Junction experiment, where a human participant is swapped with one of the cars in lane. Other cars are MARL agents trained with a budget only to send messages when necessary, as described in section 3.3. A "ghost" agent processes incoming messages to facilitate human understanding, which is translated into a graphic user interface based on the prior PCA analysis in section 3.4. Since the analyzed communication method does not create an injection with environment locations, the car's location is represented as a probability distribution. A caution triangle visualizes the likelihood of a cell being occupied by other agents on the user interface. Messages from the human are handled by a "ghost" agent, which acts as an interface between the human and messages from/to artificial agents. Previous literature has shown this method to be effective in studying communication understanding in HATs [52]. By implementing the experiment in this way, we directly present the meaning of the prototype communication to the human to speed up the learning process. Thus, the experiment can better focus on the influence of sparsity on human understanding of agent communication.

3.5.2 Methodology

Parent and Lost Child

Each participant is teamed up with an AI agent in the Parent and Lost Child environment. The humans complete 20 trials in each parent and child role, with corresponding communication tasks. We counterbalance the sequence of two experimental sessions between subjects. At the beginning of the trial, the parent and child spawn at random locations. A trial ends when the parent reaches the child or exceeds the maximum step limitation (20). The human-agent team performance is measured by 1) the number of steps the parent takes to find the child, 2) the percentage of trials in which the parent successfully found the child, and 3) the time it takes the human to make decisions.

The task interface, shown in Fig. 3.10, consists of two panels: the task environment and the communication panel. The task environment displays the true game state, the parent's vision radius, and the parent's trajectories. Interface visibility is subject to change according to the human's role. The communication panel presents numbered tokens for the child to send its

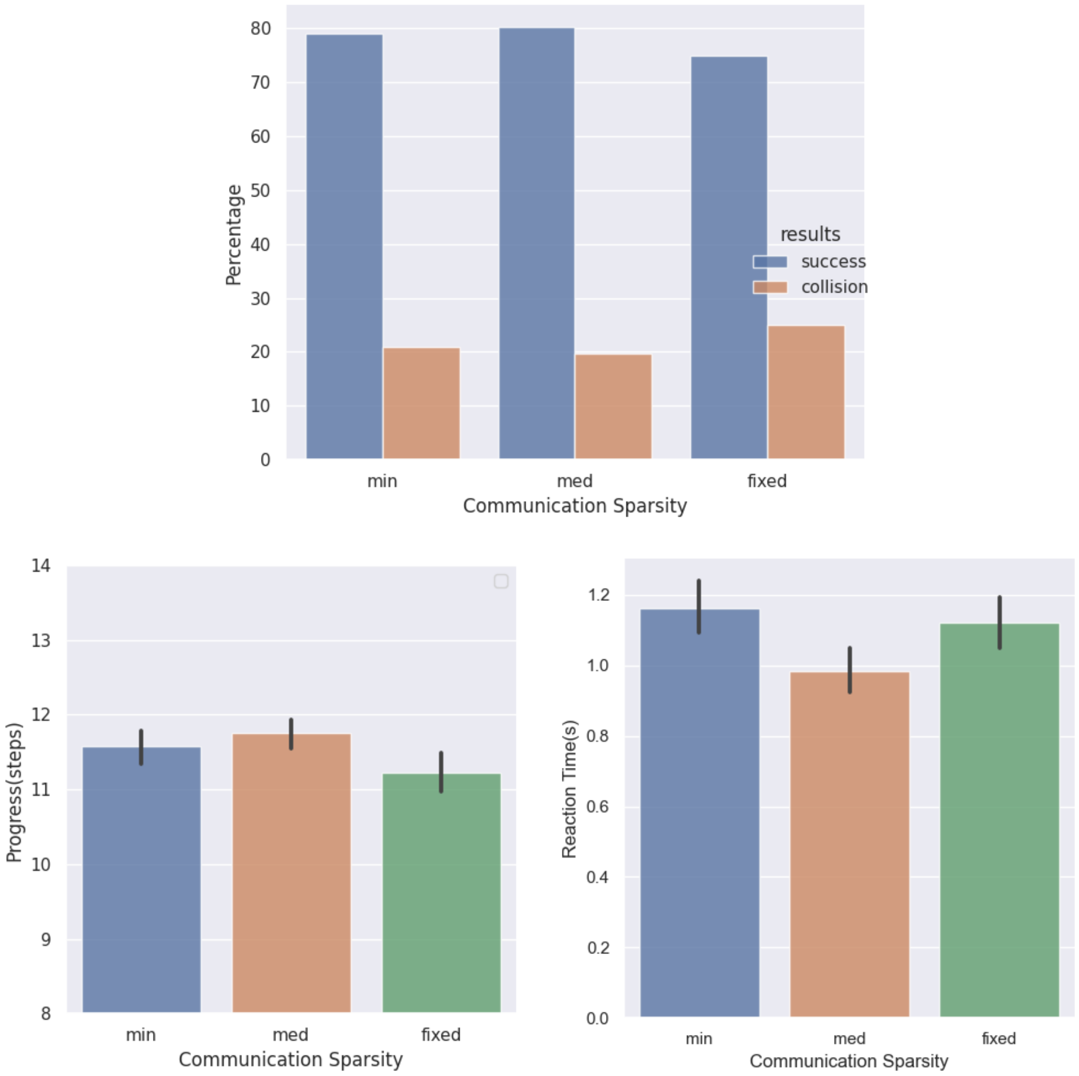


Figure 3.11: Learning curves of human-agent communication in Traffic Junction. Top left: Percentages of success in trials with three communication sparsity conditions. Top Right: Average progress made per trial. Bottom: Average reaction time. Error bars indicate 95% confidence intervals.

location. The number and arrangement of tokens are determined by the 2D PCA analysis from section 3.4. Once the child selects a token, it is highlighted on the panel.

When the human is playing the parent role, one fixed token is highlighted on the communication panel throughout the trial to signal to the human what the child agent is communicating. The human then must select a movement action to move using arrow keys on their keyboard. The participants' task as the parent is to learn to understand the meaning of the communication tokens in order to find the child in the least number of steps. When the human is in the child role, she may select only one communication token to send to the parent agent per trial but can use a different token in different trials. The human then views a replay of the entire trajectory

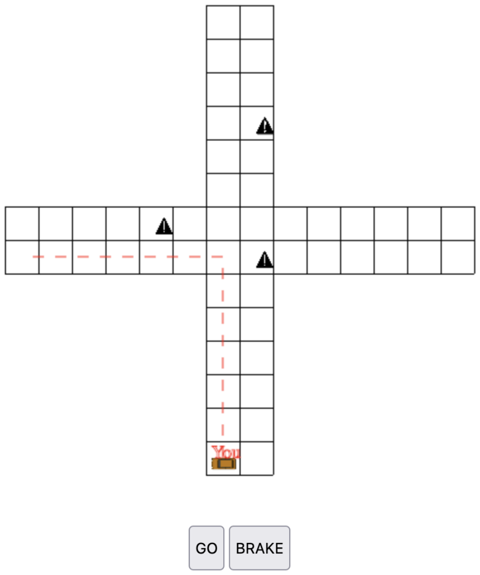


Figure 3.12: The human interface for Traffic Junction is shown above. The participants' task is to move the yellow car along the designated path (red dashed line) using two actions: go or brake. Black caution triangles refer to potential locations of other cars as revealed by agent communication.

of the parent agent as feedback to determine if the communication is helpful. Similarly, participants in the child role do not know the semantic meanings of each communication token at the beginning. Therefore, they must learn to develop the mapping between tokens and locations via repeated interactions.

Traffic Junction

In order to evaluate the effect of sparsity, human participants were recruited to team up with multiple agents in Traffic Junction. Fig. 3.12 shows the user interface of the Traffic Junction experiment. Humans act as one of the cars traversing through the junction. The goal of this task is to traverse through one's lane as far as possible without collision. All cars, including both the human and the agents, are 'blind' and have predetermined goals. The human car will enter the environment while between 5 and 9 agents are currently navigating the junction. Each trial lasts for 40 steps. The human can choose whether to accelerate or brake on the designated path to navigate the intersection without collision before the end of the trial.

3.5.3 Results

Human and Lost Child

106 participants were recruited from Amazon MTurk and Prolific, 30 were removed due to an incomplete session or failure to pass the attention check. Participants were randomly divided into two conditions, receiving either one-hot communication or prototype communication tokens.

The average number of steps taken per trial is 12.74 in the prototype condition and 13.64 in the one-hot condition. This measurement quantifies the collaboration outcome of the human-agent team. The fewer steps taken, the better the team's performance. We conducted a mixed-ANOVA in which the human role is the within-subject variable and the experimental condition is the between-subject variable. Results show that the main effects of the condition are significant $(F(1, 74) = 5.95,\ p = .017)$, indicating the proposed prototype communication indeed leads to better team performance as compared to the one-hot baseline. In addition, the main effect of the role and interaction effect between the condition and role are both significant $(p < .05)$ for each one. Paired t-tests show that prototype communication significantly outperforms one-hot baseline in the human child condition (13.09 vs. 14.54, $t(52.4) = 3.43,\ p = .001,\ cohen\ d = .83$), but not in the human parent condition. The other measurement of human-agent team performance is the completion rate shown in Fig. 3.9-middle. Chi-squared analysis shows similar patterns as above: prototype communication leads to significantly better team performance, and this difference is mainly due to the human child condition.

Another research question is how can one reveal a human's learning process of a communication language generated by reinforcement learning agents. To answer this question, we measure the reaction time of humans to decide what action to take or what communication token to send as an indicator of participants' cognitive workload. As shown in Fig. 3.9-right, the required reaction time decreases along the course of interaction between the human and agent teammates. Trial number is negatively correlated with reaction time in both child ($r = -.17,\ p < .001$) and parent ($r = -.24,\ p < .001$) roles. Specifically, this learning effect is more substantial when the human child was using prototype-based communication ($r = -.24,\ p < .001$) as compared to one-hot communication ($r = -.08,\ p = .045$). While the learning effect of team performance is not observed directly, participants actually learn the communication language during the interaction with reinforcement learning agents and complete the task faster. Our proposed prototype communication can speed up this learning process by allowing the human to bear a lower cognitive load and select the correct communication token more quickly in the child role.

In summary, **H1** confirmed that 1) prototype communication is interpretable to humans and leads to better HAT performance, and 2) prototype communication speeds up the learning process of humans by lowering the cognitive load. Our proposed method is especially effective when the human sends messages as the child. A possible explanation is the different levels of initiative between human (exploratory) and agent (exploitative) parent's searching strategies. When receiving an invalid communication, unlike the agent, the human will continue exploration.

Traffic Junction: Evaluating Sparsity

142 human participants were recruited from Amazon MTurk and Prolific, but 16 were removed due to incomplete sessions. Participants were divided into three conditions, either use prototype communication with fixed non-sparse $b = 1$, medium sparsity $b = 0.5$, or minimum sparsity $b = b^* = 0.3$. This condition division is based on agent training results, in which b^* was shown to be the minimum frequency without sacrificing performance. Each participant completed 20 trials of the experiment. Team performance in the Traffic Junction task is measured by the percentage of trials in which a human's car successfully arrived at its destination without collision. The average success rate for the three conditions is min: 79.1%, med: 80.1%, fixed: 74.6% (as

shown in Fig. 3.11-left). A chi-squared test shows that the relationship between task success and communication sparsity is significant ($\chi^2(2, N = 2408) = 7.28,\ p = .026$). Human participants who received sparse communication (i.e., min and med conditions) from agents are more likely to succeed in passing through the junction.

Recording how far each human managed to proceed before collision is also a good measure of a human's task progress. Fig. 3.11-middle shows the average progress Similar to the above analysis, one-way ANOVA shows a significant difference between conditions ($F(2, 2405) = 5.27,\ p = .005$). T-tests indicate that both min (11.57) and med (11.74) conditions outperform the fixed baseline (11.23) in making more progress ($p < .05$) for each condition.

To reveal the reasons of performance improvement brought by sparse communication, we plot the average reaction time of participants in three conditions in Fig. 3.11-right. ANOVA analysis shows a significant difference in reaction time between conditions ($F(2, 2405) = 7.49,\ p = .001$). Paired t-tests show that participants in the medium sparsity communication condition had a significantly shorter reaction time (0.98s) than those in either minimum (1.16s) and full communication (1.12s) conditions. Considering reaction time as an indicator of human cognitive workload, we found an inverted 'U' shape relationship between communication sparsity and human cognitive load. Our proposed sparse method with appropriate configuration was shown to reduce human cognitive load by sending fewer communication messages; that is, messages are sent only at necessary moments, confirming **H2**.

3.6 Conclusion

Recent work in MARL has aimed to develop sparse-discrete communication paradigms. In this work, we have analyzed a sparse-discrete method to determine its interpretability and performance when used in human-agent teaming. Through the Enforcers we can train sparse-discrete models to perform competitively with unconstrained alternatives. The results show that the intent of the communication can be trained to correlate with human-understandable observations of information necessary to complete tasks.

We conducted two human-agent teaming experiments to explore the relationship between the human learning effect and various properties of agent communication. In the first experiment, we evaluated whether humans are able to learn the communication "language" generated by reinforcement learning agents. We compared our proposed interpretable prototype-based communication against a one-hot encoding communication baseline. Results validate the superiority of prototype-based communication in better overall team performance and a faster learning effect over the baseline, confirming our proposed method's interpretability. Based on the analysis in section 3.4, we attribute these results to the correlation relationship between tokens' locations in the communication space and referenced child locations in the task environment. Prototype-based messages have a structured latent space, allowing for learning ease. Our results verify this hypothesis since there is a steeper learning curve of reaction time to decide what communication token to send.

The results of the traffic junction experiment support our hypothesis about communication sparsity in human-agent teaming with multiple agents. Both minimum and medium sparse communication enable better team performance than the full communication baseline. Measurements of reaction time as a proxy for human cognitive workload [85] provided further corroborating evidence. Reaction times followed an inverted 'U' shape, in which medium communication sparsity led to lower cognitive loads than minimum or maximum frequency; one explanation for this trend is that overly-frequent communication overloads humans, while overly-sparse communication is hard to recall. These findings align with previous literature [59] and imply that introducing an appropriate communication frequency budget is essential in supporting human-agent interaction. As emphasized by our agent self-play and HAT experiments, adapting communication to many sparse budgets is critical for effective human-agent teaming. An additional interesting discovery was that an adaptive function might be necessary to find the appropriate communication sparsity for each individual human teammate or population of humans. We will pursue this research avenue in future work.

Chapter 4

Sparse Communication

4.1 Introduction

In multi-agent teams, communication is necessary to successfully complete tasks when agents have partial observability of the environment. Multi-agent reinforcement learning (MARL) has recently seen success in scenarios that require communication [19, 77, 22, 55, 45]. Sparse multi-agent communication (wherein agents communicate during only some time-steps) has been shown to be an effective solution to internet packet routing [57], multi-robot navigation [21], complex multiplayer online games such as StarCraft [77, 72], and human-agent teaming [38]. In particular, these successes have been achieved using neural network architectures in conjunction with a reinforcement learning framework. Simultaneously, research in individualized multi-agent communication [77, 72, 11, 1] has solved sparse cooperative-competitive multi-agent problems where adversaries are listening, and sparsity is built into their competitive objective. But such research is unable to provide sparse individualized communication in fully-cooperative settings, where there is no built-in incentive. This is particularly unreasonable in real-world settings where multiple robots may need to adhere to bandwidth/budget restrictions. A budget (or bandwidth) b defines the maximum percentage of the time an agent may communicate.

Emergent communication enables agents to learn a set of communications vectors apt for solving a particular task; however, learning emergent communication simultaneously with an action policy is highly unstable. Agents often converge to undesirable policies in which communication is ignored, unless special training terms are used [14, 54]. Enforcing sparse communication, i.e., limiting the number of messages over time or communicating within a bandwidth/budget, only worsens this problem due to the additional constraint. Using the information bottleneck framework [79] may adequately address sparsity constraints [90], but due to their objective, exhibit a trade-off between the total bandwidth and task performance. In these scenarios, the agents fail to send necessary messages and transmit unnecessary messages, which we dub *null communications*. In fact, many papers on sparsity suggest lossless sparsity, but in actuality, have a non-trivial decrease in reward.

In this chapter, we propose a novel framework, *Information Maximizing Gated Sparse Multi-Agent Communication* (IMGS-MAC), which aims to learn a communication-action policy and then enforce a sparse communication budget (learning *when* and *whom* to send messages) with

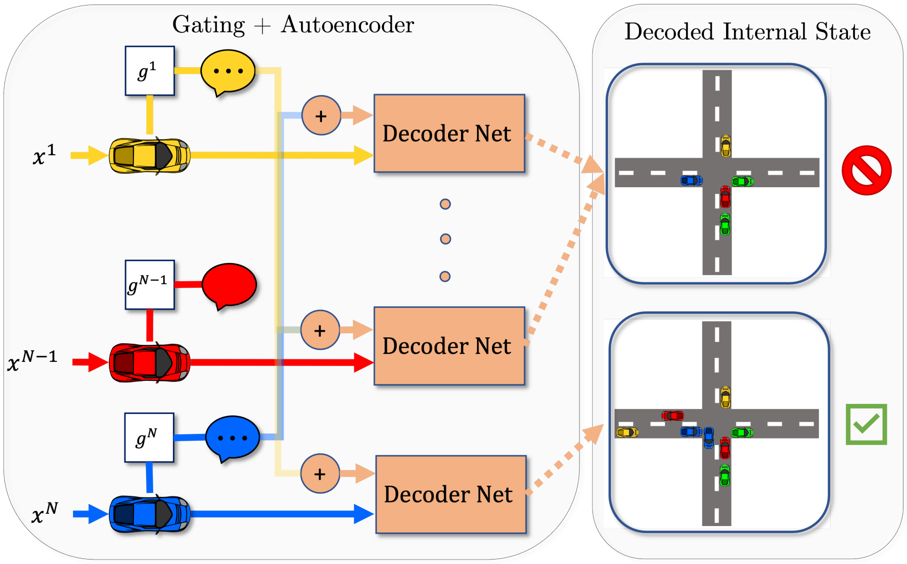


Figure 4.1: Overview of our multi-agent architecture with gated sparse, informative communication. At every timestep, each agent receives an occluded observation x. Each agent creates a communication message, which is passed to the learned gating function g as well as the `Decoder`. The gating function determines whether to communicate the message to the other agents. The `Decoder` receives all messages and attempts to reconstruct the full state of the environment.

lossless performance. Our key insight in IMGS-MAC is reframing the sparse multi-agent communication problem as a representation learning problem. The use of an information maximizing autoencoder prevents shortcut solutions in order to structure the latent communication space to allow for high reward with little communication. After learning a non-sparse communication policy, we analyze the direct causal effect of choosing to send each token to any other agent to determine null messages. Then, IMGS-MAC uses a table of these null messages to prevent them from being emitted, enabling sparsity with lossless performance without additional reinforcement learning, which we call zero-shot sparsity. To further promote sparsity for over-constrained budgets, we finetune our model using an individualized communication regularization term for a learned gating/targeting function g, which we call few-shot sparsity.

4.2 Related Work

4.2.1 Emergent Communication Vectors

Prior art in emergent communication establishes how agents may learn to communicate to accomplish their goals with continuous communication vectors [32, 77, 72]. Motivated by human communication in which people speak only when necessary and using only a discrete set of words, we wish for agents to learn sparse (in number of listeners over time) and discrete communication. While previous work has been successful in learning discrete communication vectors [44, 53, 19, 3, 21], the learned communication conventions often exhibit undesirable properties. Learning discrete prototypes has been shown to promote robustness in noisy communication channels, as well as human interpretability and zero shot generalization [81]. Similar to word embeddings in natural language processing, they capture the relationship between vectors. However, many of these methodologies only try to learn token meanings through rewards. In our work, we show that grounding messages in reproducing the concatenated state of all agents with an autoencoder creates desirable representations regardless of continuous or discrete settings.

4.2.2 Sparsity: Gating Total Messages

In this work, we attempt to reduce communication in MARL problems through gating total messages. Gating methods learn a function that dictates whether an agent will communicate to each other agent at any given timestep. Some methods try to learn a gating probability to decide whether to broadcast a budget, but these are unable to follow a communication budget [72, 28]. In reward-based sparse communication [38, 88], by penalizing communication reward during training, gating/targeting methods have reduced communication. However, this method is not able to adequately choose a budget (what maximum percentage of the time to communicate). Overall, gating methods are high variance and often unstable [3]. Rather than building the objective into the reward, I2C [13] tries to measure the causal effect of an individualized message through a learned Q-value. However, I2C only tries to address sparse targeting in the lossless sparsity case and fails to account for the effect of message representation. In our work, we measure the actual effect of each token and mask the emergent vocabulary accordingly.

4.3 Preliminaries

We formulate our setup as a centralized training, decentralized execution (CTDE) [19], partially observable Markov Decision Process with individualized communication (POMDP-Comm). Formally, our problem is defined by the tuple, $(\mathcal{S}, \mathcal{A}, \mathcal{M}, \mathcal{T}, \mathcal{R}, \mathcal{O}, \Omega, \gamma)$. We define $\mathcal{S}$ as the set of states, $\mathcal{A}_i$, $i \in [1, N]$ as the set of actions, which includes task specific actions, and $\mathcal{M}_i$ as the set of communications for N agents. $\mathcal{T}$ is the transition between states due to the multi-agent joint action space $\mathcal{T} : \mathcal{S} \times \mathcal{A}_1, ..., \mathcal{A}_N \to \mathcal{S}$. Ω defines the set of observations in our partially observable setting. The partial observability requires communication to complete the tasks successfully. $\mathcal{O}_i : \mathcal{M}_1, ..., \mathcal{M}_N \times \mathcal{S} \to \Omega$ maps the communications and state to a distribution of observations for each agent. $\mathcal{R}$ defines the reward function and γ defines the discount factor.

4.3.1 The Sparsity Objective

The multi-agent emergent communication problem is phrased as a combination of a Lewis game [48] and the information bottleneck [79]. We seek to develop a message representation M, which contains sufficient referential and ordinal information to successfully complete a task. Notably, the information bottleneck defines a trade-off between referential (X) mutual information, $I(X; M)$, which is observable to an agent, and ordinal (Y) mutual information, $I(M; Y)$, which requires coordination between agents.

The communication graph $G_t = (V, E)$ is a set of agents (vertices) and active communication edges between them, where connectivity changes at each timestep. Messages flow through the edges from agents to agents, $E : v_i \rightarrow v_j$. We aim to learn a masking function g to dynamically modify the graph to prevent messages from flowing along the graph. The total number of bits communicated, $s(M)$ can be defined in terms of vertices (gating), $v \in V$, $s(M) = \sum_{m \in \mathcal{M}} v_m$ or in terms of edges (targeting), $e \in E$, $s(M) = \sum_{m \in \mathcal{M}} e_m$, over an episode. One can see that gating is a special form of targeting in which a vertex is disjoint from the graph. We will use gating and targeting interchangeably, but in terms of sparsity, limit the total number of message edges during an episode.

In MARL, the objective of sparse communication is to minimize the total number of bits communicated while maximizing team task performance,

$$\begin{aligned}
&\max_{\pi:\mathcal{S}\rightarrow\mathcal{A}\times\mathcal{M}} \mathbb{E}\left[\sum_{t\in\mathcal{T}}\sum_{i\in N}\gamma\mathcal{R}(s_t, a_t)\right] \\
&\text{s.t. } (a_t, m_t) \sim \pi, s_t \sim \mathcal{T}(s_{t-1}) \\
&\text{subject to} \\
&\min \mathbb{E}_{M\sim\pi}\left[s(M)\right]
\end{aligned} \tag{4.1}$$

That is, to achieve this objective, first one maximizes task performance; then one reduces total communication while keeping task performance fixed.

Definition 4.3.1 (Lossless Sparse Communication)**.** A communication policy π_m is lossless and sparse iff it satisfies the objective in equation 4.1. A lossless sparse communication policy defines the minimum sparse budget (fraction of total messages) b^*.

Most sparse communication work rephrases the $\min\max$ problem to a single objective by introducing a Lagrangian,

$$\begin{aligned}
&\max_{\pi:\mathcal{S}\rightarrow\mathcal{A}\times\mathcal{M}} \mathbb{E}\left[\sum_{t\in\mathcal{T}}\sum_{i\in N}\gamma\mathcal{R}(s_t, a_t) - \lambda s(m_t)\right] \\
&\text{s.t. } (a_t, m_t) \sim \pi, s_t \sim \mathcal{T}(s_{t-1}), m_{AVG} < b
\end{aligned} \tag{4.2}$$

However, depending on the Lagrange multiplier, the objective in equation 4.1 is not always the same as equation 4.2. Due to the dual-objective, equation 4.2 also introduces the possibility of suboptimal sparse communication even when lossless sparse communication is possible. It also explains the high variance of lossless sparsity in prior art [3].

Definition 4.3.2 (Sub-Optimal Sparse Communication)**.** A communication policy π_M is suboptimal and sparse iff there exists a trade-off between task performance and messaging constraints as defined in equation 4.2.

Thus, in our methodology, we cannot directly optimize equation 4.2. Recall that in emergent communication, messages are generated based on their observations. This implies that, in terms of the information bottleneck, messages represent a combination of referential, $I(X; M)$, and ordinal, $I(M; Y)$, information. That is, observations help guide ordinal (task-specific) information. Suppose, we have a Lagrangian objective (see section 4.4.1), which allows for our messages to have independent referential information. Then, given a communication policy which adequately

Algorithm 1 IMGS-MAC

```
$\theta \leftarrow$ randomly initialized network parameters
useDiscreteMessaging $\leftarrow \{true|false\}$
while not converged do
    for $i \leftarrow 1$ to $N$ {simultaneously} do
        $x^i \sim \mathcal{S}$
        $h^i \leftarrow$ GRU$(x^i)$
        if useDiscreteMessaging then
            $m^i \leftarrow$ DiscreteProtoNet$(h^i)$
        else
            $m^i \leftarrow h^i$
        end if
        SendMessages$(m^i \odot g(h^i))$
        $\bar{m}^i \leftarrow$ AggregateMessages$()$
        $\tilde{h}^i \leftarrow$ GRU$(\{h^i, \bar{m}^i\})$
        $a^i, v^i, \tilde{x}^i \leftarrow \pi(\tilde{h}^i), V(\tilde{h}^i),$ DecoderNet$(\tilde{h}^i)$
        $L \leftarrow \pi$Loss$(a^i, v^i) + \mathcal{L}_1(x, \tilde{x}^i) + \mathcal{L}_2(m^i_{AVG})$
    end for
end while
```

solves the task, one can determine the ordinal utility of each token. By removing unnecessary tokens, we can satisfy the objective in equation 4.1. Thus, in our methodology, we emphasize learning emergent communication with properties that enable sparse communication with lower optimal budgets b^* (lossless sparsity).

4.4 Proposed Methodology

In this section, we introduce the IMGS-MAC architecture as well as two types of individualized regularization. The first is an autoencoder, which is used to stabilize the dual training of the communication-action policy. The latter is an individualized communication penalty to enforce each agent individually follows a fixed communication budget/bandwidth. Note that it is important to provide individualized regularization, as otherwise the gradient signal will not be adequately recognized. Our model builds on related art [72, 3], but our technique can be easily applied to any individualized MARL communication module. Below, we introduce our information maximization autoencoder and individualized communication regularization. Overall, the combined framework can be observed in Alg. 1.

4.4.1 Sparsity through Information Maximization

The information bottleneck principle [79] is naturally encoded into any communication module that uses deep learning. By creating a latent representation, any nontrivial solution enforces the network to provide the relevant information within the communication vector. Rather than requiring centralized execution to maintain sparsity through the information bottleneck, we provide a form of information regularization that allows for individualized communication. Additionally, we enforce a structured representation for message tokens, ensuring that tokens represent independent referential and ordinal information from their observations.

We define the autoencoder as follows: The communication module of our network serves as the encoder. Each agent produces their own hidden state h^i and receives communication vectors m^j such that $i \neq j$. For each agent, the model feeds $h^i + m^j$ into the decoder. We then calculate the $l2$ loss $U(s_t, s_t^{i,\texttt{decoded}})$ between the state of all agents $s_t = \{x_t^1, \dots, x_t^N\}$ and the decoded state $s_t^{i,\texttt{decoded}}$, which effectively measures the similarity between the latent communication and the concatenated state of all agents.

$$\mathcal{L}_1(\theta) = \lambda_1 U(s_t, s_t^{i,\texttt{decoded}}) \tag{4.3}$$

To enable sparsity, we first train IMGS-MAC with the autoencoder module and non-sparse communication ($b = 1$). Afterwards, we run evaluation episodes while collecting data regarding each message token to detect null messages.

Definition 4.4.1 (Null Communication Vector)**.** A null communication vector from agent i provides a lack of information to another agent j. That is, in terms of the information bottleneck, $I(m^i; y^j) = 0$.

To determine the mutual information between a message m and the task specific information y, we measure if there is a change in the reward within a small $\epsilon \approx 1e - 3$. If there is no significant change, we consider this token a null message.

While simple, in our experiments, we show that by combining this trick with strong latent representations, our model can remove larger amounts of unnecessary communication, or null communication vectors, without impacting the performance. In fact, our lossless sparsity method requires no additional reinforcement learning training, which we define as zero-shot sparsity.

Similar to zero-shot learning, which requires no additional data to satisfy an objective, zero-shot sparsity enables satisfaction of sparse communication constraints from non-sparse training through careful analysis of the emergent communication policy. Our methodology exhibits zero-shot sparsity since no additional reinforcement learning training is required to enforce sparsity given our non-sparse model with informative communication, which is shown in section 4.6.

4.4.2 Sparsity through Individualized Regularization

In the overconstrained bandwidth case, $b < b^*$, which implies that we will not be able to maximize task performance, inducing the suboptimal sparsity case. However, we can use the properties of lossless sparsity to maximize performance such that $m_{AVG} <= b$. We combine previous techniques with a second regularization term, a per-agent communication penalty $\mathcal{L}_2$. The penalty depends on the nature of the communication budget. At each discrete time-step t, each agent has the opportunity to choose to emit a message. Thus, we define our budget b as a fraction of the total agents multiplied by the time-steps in which we measure communications. We let m_{AVG} define the actual fraction of messaging. Finally, we can define the regularization penalty,

$$\mathcal{L}_2(\theta) = \lambda_2 \left\| m^i_{AVG} - (b + (1 - b^*)) \right\|_2^2 \tag{4.4}$$

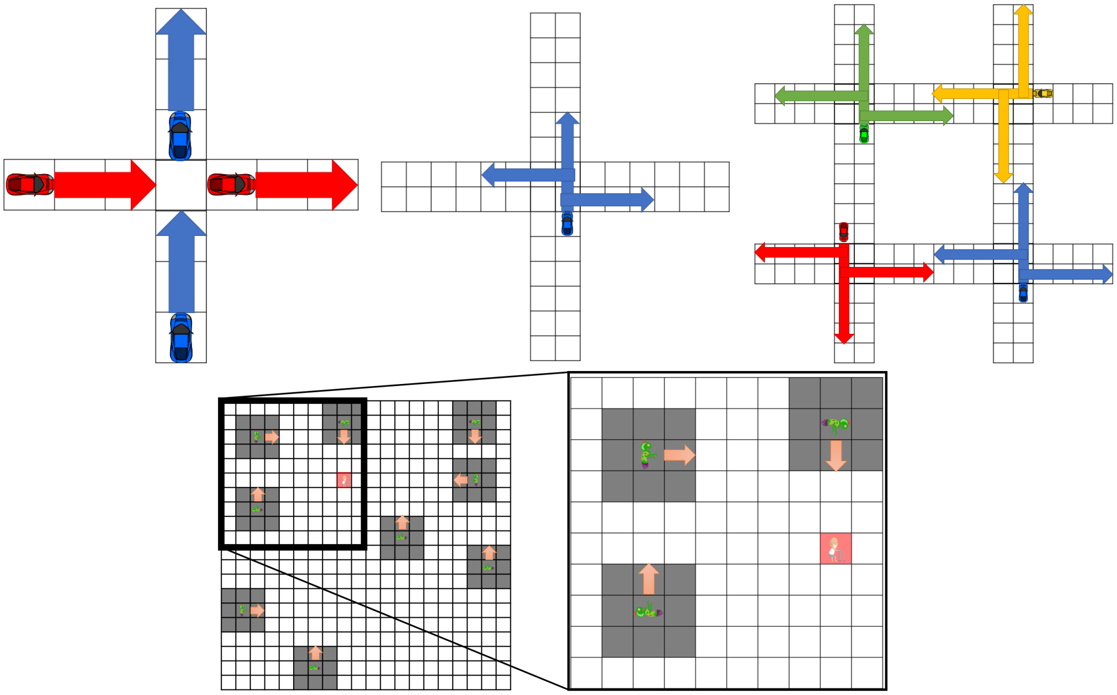


Figure 4.2: Above are the easy, medium, and hard traffic junction environments. Visibility is limited to the cell the car is located, so agents are effectively blind. The bottom shows a zoomed-in view of the 20×20 predator-prey environment. The predators are denoted by green aliens, while the prey is denoted by a human (in a red square).

where we penalize messages when $b < m_{AVG} < b^*$.

Similar to few-shot learning where a limited amount of data, we define few-shot sparsity as enabling the satisfaction of sparse communication constraints from non-sparse training through limited additional MARL training. We quantify the amount of data in our experiments, notably Fig. 4.4. We finetune our model using the regularization penalty in Eq. 4.4 to observe overconstrained budgets, thus exhibiting few-shot sparsity.

4.5 Experimental Setup

We train and evaluate our model in a blind traffic junction and predator-prey environment settings following prior benchmarks [72, 77, 13]. For each of these variants, we train on 10 random seeds and one epoch uses 5000 samples. We used an RMSProp optimizer with a learning rate of 0.003. See Figure 4.2.

The blind traffic junction scenario involves multiple agents navigating a discretized narrow intersection with no observability regarding the locations of the other agents. Clearly, this necessitates informative communication in order to avoid collisions in the environment. Note that both communication and action occur in a single time-step. We study three variants of the blind traffic junction and report results on the easiest and hardest environments which converge for continuous and discrete communication.

The predator-prey scenario involves multiple agents, where one agent is denoted as the prey and the remaining agents are denoted as predators. The predator agents move and search the environment for the prey agent. The predator agents can only observe its current cell and the adjacent cells (limited visibility to 1 cell around itself). The episode terminates when all predator agents reach the prey agent or when the maximum episode length is hit.

Predator-prey does not necessarily require communication to solve the task. However, in the fully-cooperative predator-prey environment, predators are rewarded for maximizing the number of predators who reach the discovered prey. Thus, there is no built-in incentive for fully-cooperative teams to decrease total communication. In our experiments, we show that our method, IMGS-MAC, is able to decrease messaging to a minimum sparse budget b^* with lossless performance.

Overall, our proposed method is trained ("pretraining") using the autoencoder in Eq. 4.3. We then analyze to determine if our model will follow a lossless sparse budget. If not, we finetune our model for a suboptimal sparse budget (Def. 4.3.2) using the message penalty in Eq. 4.4.

We use REINFORCE [91] to train both the gating function and policy network subject to the previous constraints. In order to calculate the information similarity, we compute loss, using Eq. 4.3, between each agent's decoded state $s_t^{i,\texttt{decoded}}$ and the concatenation of all agents' states s_t.

4.6 Experiments

In this section, we first describe the benchmark environment. Then, we present ablations showing the efficacy of our sparse model with informative communication. As stated in section 4.2, IC3Net and I2C provide close framework compatibility. We compare IMGS-MAC with IC3Net with non-sparse ($b = 1$) communication to understand the effect of our information maximizing autoencoder in developing independent referential (based in observations x) representations, $I(m_j; m_k) = 0$. We evaluate with both continuous and discrete messages to show the necessity of using our methodology to develop structured latent tokens (messages m). Then we show the few-shot sparsity benefits of finetuning sparse budgets when $b < b^*$ as compared with solving the tri-objective (1: communicate effectively, 2: act effectively, and 3: obey communication sparsity constraints), which is akin to trying to satisfy the objective in Eq. 4.2 when $b \geq b^*$. We analyze our model's communication vectors to find zero-shot sparsity $b = b^*$. We show that our method can provide lower optimal budgets b^* than I2C. Finally, we verify that IMGS-MAC has lossless performance at $b = b^*$ as compared with its non-sparse performance $b = 1$, and show the optimized trade-off between suboptimal budgets $b < b^*$ and task performance, e.g., reward. We detail our experimental setup in Appendix 4.5.

4.6.1 Information Maximization Analysis

To show the benefits of the autoencoder for information maximization, we first show comparison with IC3Net with a fixed gate, i.e., non-sparse communication ($b = 1$). In Figure 4.3, our results show that our method has much lower variance. Note that IC3Net may have a shaded area higher than IMGS-MAC, but it never actually performs that well. Rather, the variance comes from very

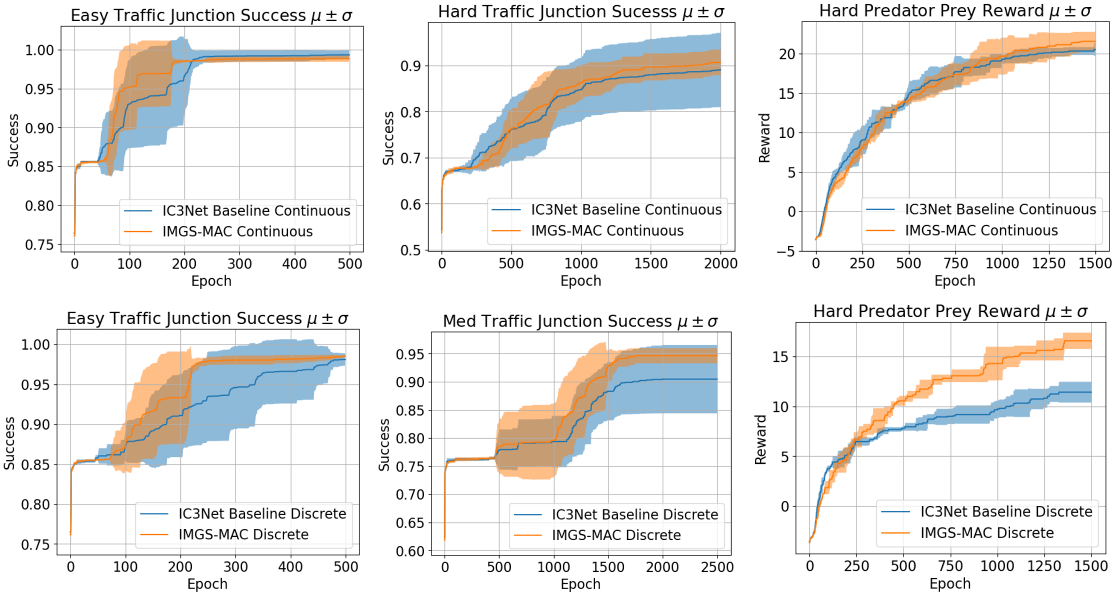


Figure 4.3: Left and middle figures compare the training IC3Net (blue) vs. our IMGS-MAC (orange) with non-sparse communication ($b = 1$) in Traffic Junction. Our method converges to higher success earlier and with less variance. Right figures compare in Predator-Prey. Our method converges to higher success earlier and with less variance. Top figures use continuous communication vectors while bottom figures use discrete.

low performing runs. In the simple, easy setting, our method is able to find solutions of equivalent quality as IC3Net. However, in hard settings, and in all discrete communication vector settings, our method outperforms IC3Net in terms of performance and the number of epochs required to find the solution. Particularly, in the more difficult discrete communication vector scenarios, the autoencoder drastically outperforms IC3Net. Note that the decreased variance results in much more stable solutions.

Our hypothesis is that decreasing the training epochs to converge to high task performance implies that we have more informative communication. Our results show that communication tokens which represent information more independently allow for lower b^*. This is found by analyzing the number of states in which the same message is emitted. Overall, this strengthens our hypothesis that a structured latent space naturally allows for lower b^* for lossless sparsity. We analytically study the performance of the autoencoder in Table 4.1.

Percent null communication vectors determines the number of null tokens in the emergent ‘vocabulary’, i.e., all possible messages. The number of observations per vector reports the independence of a token or mutual information between any two distinct tokens, $I(m_j; m_k)$. We want to minimize $I(m_j; m_k)$ in order to decouple information into independent messages, so that we can later promote stronger sparsity through the analysis of the utility of each token in determining optimal actions. The percent of null communications emitted reports the percentage of null messages that were communicated to other agents over 500 episodes. We aim to minimize these unnecessary null messages. We see that the IC3Net method uses more null vectors on average and has high mutual information between tokens. Further, using our IMGS-MAC,

Table 4.1: Average $\mu \pm \sigma$ for quality and performance of null communication vectors. IMGS-MAC (ours) provides significantly more informative communication, as recognized by its low usage of null communications. Lower is better.

Environment	Method	% Null Comm. Vectors	# Observations per Vector	% Null Comms. Emitted
TJ Easy Cts.	IC3Net	0.59 ± 0.107	3.81 ± 0.304	0.529 ± 0.112
	IMGS-MAC	0.0550 ± 0.198	1.785 ± 0.507	0.0565 ± 0.196
TJ Hard Cts.	IC3Net	0.404 ± 0.0753	26.892 ± 6.662	0.543 ± 0.0999
	IMGS-MAC	0.0334 ± 0.107	16.928 ± 10.113	0.0310 ± 0.167
TJ Easy Discrete	IC3Net	0.589 ± 0.265	3.39 ± 1.09	0.846 ± 0.263
	IMGS-MAC	0.0194 ± 0.0394	1.390 ± 0.220	0.0320 ± 0.0719
TJ Med. Discrete	IC3Net	0.724 ± 0.139	15.944 ± 8.127	0.964 ± 0.0424
	IMGS-MAC	0.0857 ± 0.172	5.105 ± 3.154	0.201 ± 0.322
PP Hard Cts.	IC3Net	0.784 ± 0.0445	73.148 ± 12.099	0.497 ± 0.0887
	IMGS-MAC	0.284 ± 0.160	17.523 ± 6.231	0.300 ± 0.173
PP Hard Discrete	IC3Net	0.482 ± 0.145	104.803 ± 6.0713	0.719 ± 0.312
	IMGS-MAC	0.380 ± 0.0909	82.809 ± 6.507	0.141 ± 0.114

we effectively remove null messages and decrease mutual information between tokens, further improving performance. In fact, IMGS-MAC removes almost all null messages. We will later further see that it does so without any reduction in performance.

4.6.2 Sparsity Analysis

Few-shot Sparsity

In the case where $b < b^*$ we require a small amount of additional training data to enable sparse communication. We introduce an autoencoder to include independent referential communication in order to ease the dual communication-action policy learning. When we introduce the sparsity constraint (and the corresponding individualized communication regularization), our model must additionally learn a gating function, which further increases the complexity. In order to avoid requiring more data, we introduce a pretraining and finetuning paradigm. First, we pretrain dual communication-action policy with a fixed open gate (non-sparse $b = 1$). Then, we apply finetuning to train the gating function (with the rest of the network) at any $b < b^*$. In Figure 4.4, we see that the total number of epochs required for task success convergence under a budget is about half as many for the pretraining+finetuning paradigm than for the tri-objective, which aims to solve the objective in Eq. 4.2 directly. Note that the variance entirely comes from the dual objective pretraining. The sparsity finetuning requires less than 10% of the total training epochs. In fact, we can apply finetuning for any budget b rather than having to train the tri-objective from scratch, further decreasing training time. In Figure 4.5, we observe that our model only needs a few dozen epochs to converge to a communication budget and is able to safely reduce

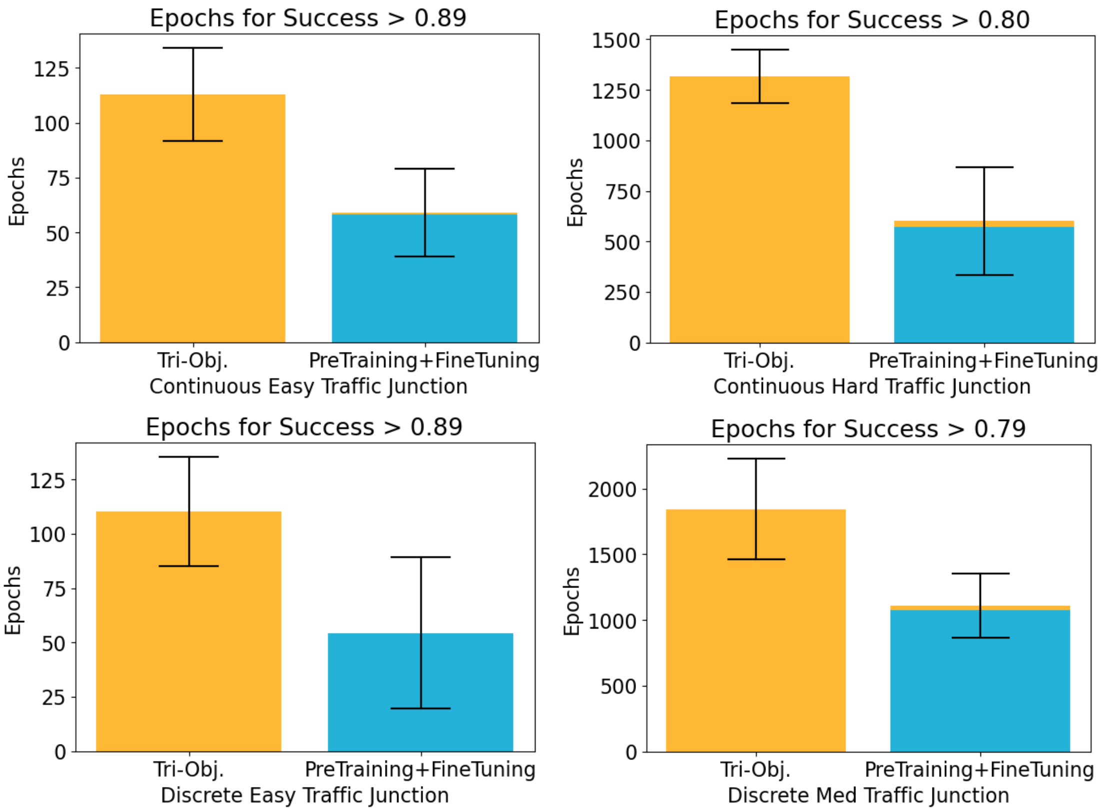


Figure 4.4: Average success and 95% confidence interval for Tri-objective (left bar, orange) vs. Pretraining with non-sparse $b = 1$ (blue), then Finetuning (orange) with $b = 0.7$. The Pretraining+Finetuning paradigm takes half the amount of training as the Tri-objective.

Table 4.2: Minimum sparse budget b^* with lossless performance, $\mu \pm \sigma$. Observe that our model can reduce 20-60% without a loss in task performance.

Environment	**IMGS-MAC** b^*	I2C-Cts. b^*
TJ Easy Cts.	0.610 ± 0.191	-
TJ Hard Cts.	0.462 ± 0.249	0.63
TJ Easy Discrete	0.815 ± 0.00469	-
TJ Med Discrete	0.519 ± 0.140	0.66
PP Hard Cts.	0.244 ± 0.0644	0.48
PP Hard Discrete	0.263 ± 0.00757	0.48

total communication below the allowed budget. Overall, our objective exhibits *few-shot sparsity* ($b < b^*$). The performance of few-shot sparsity is analyzed in Figure 4.6.

Zero-shot Sparsity

We use sparsity through information maximization in section 4.4.1 to reduce the number and usage of null prototypes. In Table 4.1, one can see that through our analysis, we are able to remove

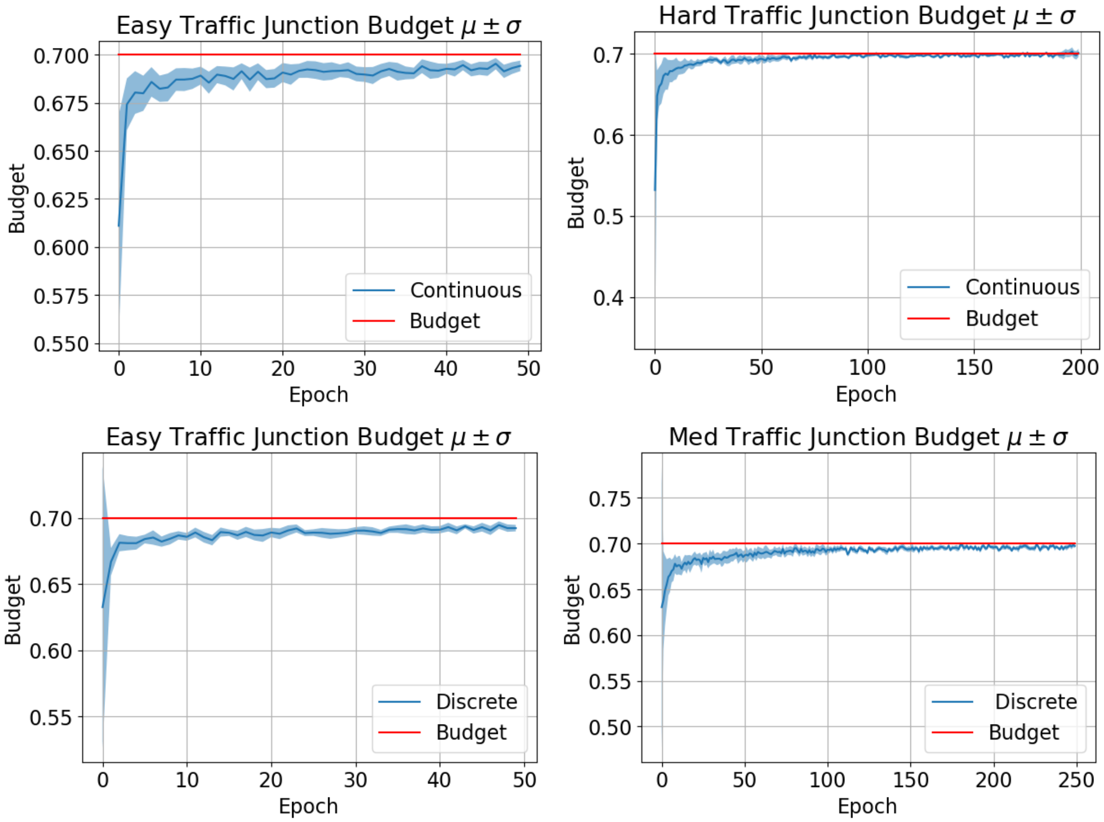


Figure 4.5: Above, the model follows the budget $b = 0.7$ average over each episode. Observe that the model (in blue) only needs to run for a few dozen epochs before adequately following the budget (in red).

significant usage of null communication vectors, which allow our model to only use informative communication, enabling **true lossless sparsity**. That is, the task performance, or success in our case, will not decrease at all by decreasing the budget within the true lossless range. Otherwise, enforcing a budget requires the learned gating function g to determine whether an agent should communicate, which may induce a loss in task performance. Of course, this is dependent on how well the initial communication model is learned, i.e., the range is dependent on the learned model. Each model has its own minimum lossless budget b^*, which depends on the emergent communication model. In Table 4.2, we report the lossless budget b^* for each environment. We are able to reduce communication by 20-75% with no additional training. Interestingly, we are able to reduce communication more when we have continuous communication vectors instead of discrete communication vectors. This implies that our continuous vectors have more informative communication. Though, it most likely follows from the fact that discrete communication is a harder problem than continuous communication, confirming results from [82]. Additionally, we are able to find lower optimal budgets b^* than I2C, even without specific reinforcement learning training to reduce the communication overhead.

Finally, we analyze the lossless, $b = b^*$, and suboptimal, $b < b^*$, performance for sparse budgets for our model in Figure 4.6, which uses the lossless budget b^* as reported in Table 4.2. We

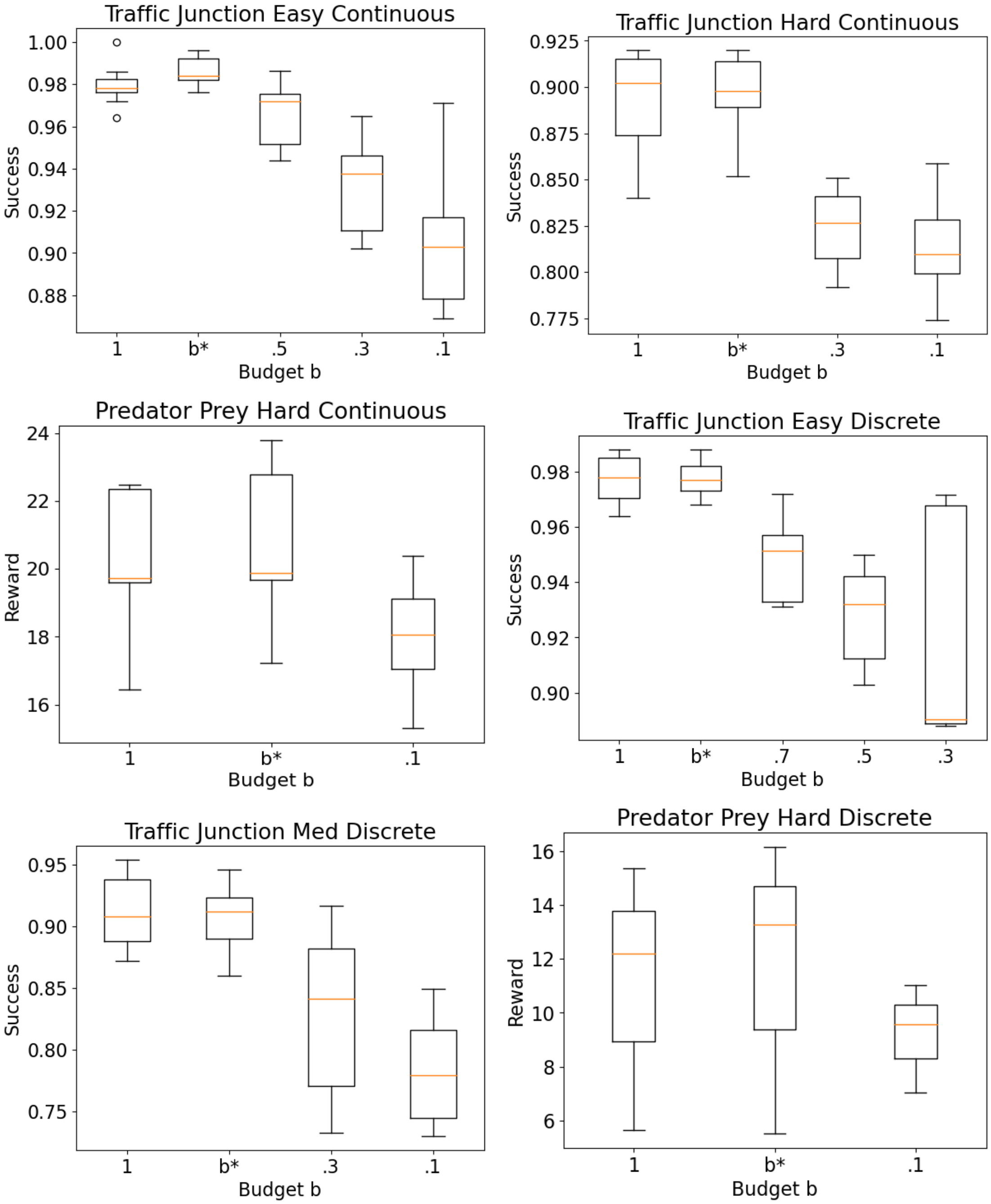


Figure 4.6: Success versus budget for IMGS-MAC at baseline non-sparse $b = 1$, lossless $b = b^*$, and suboptimal $b < b^*$. Our model provides lossless performance for $b = b^*$ for b^* in Table 4.2 as compared with the baseline non-sparse $b = 1$. Our performance tapers for smaller budgets until it approaches the no communication performance. Top: continuous communication vectors; Bottom: discrete; Left, middle: Traffic Junction; Right: Predator-Prey.

find that the lossless budget b^* provides true lossless performance. Unsurprisingly, for overconstrained budgets $b < b^*$, there is a small task performance tradeoff for adherence to the budget.

4.7 Conclusion and Future Work

In this paper, we have proposed a method for multi-agent individualized sparse communication. We reframed sparsity as a representation learning problem through the information bottleneck problem. We have shown that through training a communication-action policy grounded with an autoencoder and analysis during execution of non-sparse messaging, one can exhibit lossless zero-shot sparsity. That is, the sparsity objective may be achieved without any cost of performance with no additional reinforcement learning training. Additionally, we produce individualized regularization to limit performance loss with few-shot sparsity. This allows our model to adhere to messaging constraints in over-constrained bandwidth scenarios. In a limitation of our work, once the 'vocabulary' is restricted by removing some null messages, other messages are discovered later that could be removed and mutual information between tokens is nonzero. Stronger theoretical bounds on message content independence will further allow sparser communication. In our future work, we aim to create an overarching framework that combines gating/targeting sparsity and communication compression. This will remove the need for tuning message sizes, but still opt for a decoupled training scenario. That is, first learn an emergent language. Then adhere to sparsity constraints. Additionally, further increases to the unsupervised representation learning will allow for sparser performance.

Chapter 5

Compositional Messages and Social Learning

5.1 Introduction

Social learning [31, 62] agents analyze cues from direct observation of other agents (novice or expert) in the same environment to learn an action policy from others. However, observing expert actions may not be sufficient to coordinate with other agents. Rather, by learning to communicate, agents can better model the intent of other agents, leading to better coordination. In humans, explicit communication for coordination assumes a common communication substrate to convey abstract concepts and beliefs directly [60], which may not be available for new partners. To align complex beliefs, heterogeneous agents must learn a message policy that translates from one theory of mind [50] to another to synchronize coordination. Especially when there is complex information to process and share, new agent partners need to learn to communicate to work with other agents.

Emergent communication studies the creation of artificial language. Often phrased as a Lewis game, speakers and listeners learn a set of tokens to communicate complex observations [48]. However, in multi-agent reinforcement learning (MARL), agents suffer from partial observability and non-stationarity (due to unaligned value functions) [64], which aims to be solved with decentralized learning through communication. In the MARL setup, agents, as speakers and listeners, learn a set of tokens to communicate observations, intentions, coordination, or other experiences which help facilitate solving tasks [37, 38]. Agents learn to communicate effectively through a backpropagation signal from their task performance [19, 55, 45, 77, 72]. This has been found useful for applications in human-agent teaming [38, 58, 43, 44], multi-robot navigation [21], and coordination in complex games such as StarCraft II [69]. Communication quality has been shown to have a strong relationship with task performance [59], leading to a multitude of work attempting to increase the representational capacity by decreasing the convergence rates [14, 54, 37, 90, 82]. Yet these methods still create degenerate communication protocols [38, 37, 21], which are uninterpretable due to joined concepts or null (lack of) information, which causes performance degradation.

In this work, we investigate the challenges of learning a messaging lexicon to prepare emergent communication for social learning (EC4SL) scenarios. We study the following hypotheses: **H1)** EC4SL will learn faster through structured concepts in messages leading to higher-quality solutions, **H2)** EC4SL aligns the policies of expert heterogeneous agents, and **H3)** EC4SL enables social shadowing, where an agent learns a communication policy while only observing an expert agent's action policy. By learning a communication policy, the agent is encouraged to develop a more structured understanding of intent, leading to better coordination. The setting is very realistic among humans and many computer vision and RL frameworks may develop rich feature spaces for a specific solo task, but have not yet interacted with other agents, which may lead to failure without alignment.

We enable a compositional emergent communication paradigm, which exhibits clustering and informativeness properties. We show theoretically and through empirical results that compositional language enables independence properties among tokens with respect to referential information. Additionally, when combined with contrastive learning, our method outperforms competing methods that only ground communication on referential information. We show that contrastive learning is an optimal critic for communication, reducing sample complexity for the unsupervised emergent communication objective. In addition to the more human-like format, compositional communication is able to create variable-length messages, meaning that we are not limited to sending insufficiently compressed messages with little information, increasing the quality of each communication.

In order to test our hypotheses, we show the utility of our method in multi-agent settings with a focus on teams of agents, high-dimensional pixel data, and expansions to heterogeneous teams of agents of varying skill levels. Social learning requires agents to explore to observe and learn from expert cues. We interpolate between this form of social learning and imitation learning, which learns action policies directly from examples. We introduce a 'social shadowing' learning approach where we use first-person observations, rather than third-person observations, to encourage the novice to learn latently or conceptually how to communicate and develop an understanding of intent for better coordination. The social shadowing episodes are alternated with traditional MARL during training. Contrastive learning, which works best with positive examples, is apt for social shadowing. Originally derived to enable lower complexity emergent lexicons, we find that the contrastive learning objective is apt for agents to develop internal models and relationships of the task through social shadowing.

The idea is to enable a shared emergent communication substrate (with minimal bandwidth) to enable future coordination with novel partners. Our contributions are deriving an optimal critic for a communication policy and showing that the information bottleneck helps extend communication to social learning scenarios. In real-world tasks such as autonomous driving or robotics, humans do not necessarily learn from scratch. Rather they explore with conceptually guided information from expert mentors. In particular, having structured emergent messages reduces sample complexity, and contrastive learning can help novice agents learn from experts. Emergent communication can also align heterogeneous agents, a social task that has not been previously studied.

5.2 Related Work

5.2.1 Multi-Agent Signaling

Implicit communication conveys information to other agents that is not intentionally communicated [26]. Implicit signaling conveys information to other agents based on one's observable physical position [26]. Implicit signaling may be a form of implicit communication such as through social cues [31, 62] or explicit communication such as encoded into the MDP through "cheap talk" [75]. Unlike implicit signaling, explicit signaling is a form of positive signaling [53] that seeks to directly influence the behavior of other agents in the hopes that the new information will lead to active listening. Multi-agent emergent communication is a type of explicit signaling which deliberately shares information. Symbolic communication, a subset of explicit communication, seeks to send a subset of pre-defined messages. However, these symbols must be defined by an expert and do not scale to particularly complex observations and a large number of agents. Emergent communication aims to directly influence other agents with a learned subset of information, which allows for scalability and interpretability by new agents.

5.2.2 Emergent Communication

Several methodologies currently exist to increase the informativeness of emergent communication. With discrete and clustered continuous communication, the number of observed distinct communication tokens is far below the number permissible [81]. As an attempt to increase the emergent "vocabulary" and decrease the data required to converge to an informative communication "language", work has added a bias loss to emit distinct tokens in different situations [14]. More recent work has found that the sample efficiency can be further improved by grounding communication in observation space with a supervised reconstruction loss [54]. Information-maximizing autoencoders aim to maximize the state reconstruction accuracy for each agent. However, grounding communication in observations has been found to easily satisfy these input-based objectives while still requiring a myriad more samples to explore to find a task-specific communication space [37]. Thus, it is necessary to use task-specific information to communicate informatively. This will enable learned compression for task completion rather than pure compression for input recovery. Other work aims to use the information bottleneck [79] to decrease the entropy of messages [90]. In our work, we use contrastive learning to increase representation similarity with future goals, which we show optimally optimizes the Q-function for messages.

5.2.3 Natural Language Inspiration

The properties of the tokens in emergent communication directly affect their informative ability. As a baseline, continuous communication tokens can represent maximum information but lack human-interpretable properties. Discrete 1-hot (binary vector) tokens allow for a finite vocabulary, but each token contains the same magnitude of information, with equal orthogonal distance to each other token. Similar to word embeddings in natural language, discrete prototypes are an effort to cluster similar information together from continuous vectors [81]. Building on the continuous word embedding properties, VQ-VIB [82], an information-theoretic observation grounding based on VQ-VAE properties [83], uses variational properties to provide word embedding properties for continuous emergent tokens. Like discrete prototypes, they exhibit a clustering property based on similar information but are more informative. However, each of these message types determines a single token for communication. Tokens are stringed together to create emergent "sentences".

5.3 Preliminaries

We formulate our setup as a decentralized, partially observable Markov Decision Process with communication (Dec-POMDP-Comm). Formally, our problem is defined by the tuple, $\langle \mathcal{S}, \mathcal{A}, \mathcal{M}, \mathcal{T}, \mathcal{R}, \mathcal{O}, \Omega, \gamma \rangle$. We define $\mathcal{S}$ as the set of states, $\mathcal{A}^i$, $i \in [1, N]$ as the set of actions, which includes task-specific actions, and $\mathcal{M}^i$ as the set of communications for N agents. $\mathcal{T}$ is the transition between states due to the multi-agent joint action space $\mathcal{T} : \mathcal{S} \times \mathcal{A}^1, ..., \mathcal{A}^N \rightarrow \mathcal{S}$. Ω defines the set of observations in our partially observable setting. Partial observability requires communication to complete the tasks successfully. $\mathcal{O}^i : \mathcal{M}^1, ..., \mathcal{M}^N \times \hat{\mathcal{S}} \rightarrow \Omega$ maps the communications and local state, $\hat{\mathcal{S}}$, to a distribution of observations for each agent. $\mathcal{R}$ defines the reward function and γ defines the discount factor.

5.3.1 Architecture

The policy network is defined by three stages: Observation Encoding, Communication, and Action Decoding. The best observation encoding and action decoding architecture is task-dependent, i.e., using multi-layer perceptrons (MLPs), CNNs [46], GRUs [10], or transformer [87] layers are best suited to different inputs. The encoder transforms observation and any sequence or memory information into an encoding H. The on-policy reinforcement learning training uses REINFORCE [91] or a decentralized version of MAPPO [93] as specified by our experiments.

Our work focuses on the communication stage, which can be divided into three substages: message encoding, message passing (often considered sparse communication), and message decoding. We use the message passing from [37]. For message decoding, we build on a multi-headed attention framework, which allows an agent to learn which messages are most important [1]. Our compositional communication framework defines the message encoding, as described in section 5.4.

5.3.2 Objective

Mutual information, denoted as $I(X; Y)$, looks to measure the relationship between random variables,

$$I(X; Y) = \mathbb{E}_{p(x,y)} \left[\log \frac{p(x|y)}{p(x)} \right] = \mathbb{E}_{p(x,y)} \left[\log \frac{p(y|x)}{p(y)} \right]$$

which is often measured through Kullback-Leibler divergence [42],
$I(X; Y) = D_{KL}(p(x, y) || p(x) \otimes p(y))$. The message encoding substage can be defined as an

information bottleneck problem, which defines a trade-off between the complexity of information (compression, $I(X, \hat{X})$) and the preserved relevant information (utility, $I(\hat{X}, Y)$). The deep variational information bottleneck defines a trade-off between preserving useful information and compression [4, 79]. We assume that our observation and memory/sequence encoder provides an optimal representation H^i suitable for sharing relevant observation and intent/coordination information. We hope to recover a representation Y^i, which contains the sufficient desired outputs.

In our scenario, the information bottleneck is a trade-off between the complexity of information $I(H^i; M^i)$ (representing the encoded information exactly) and representing the relevant information $I(M^{j\neq i}; Y^i)$, which is signaled from our contrastive objective. In our setup, the relevant information flows from other agents through communication, signaling a combination of the information bottleneck and a Lewis game. We additionally promote complexity through our compositional independence objective, $I(M^i_1; \ldots ; M^i_L | H^i)$. This is formulated by the following Lagrangian,

$$\begin{aligned}\mathcal{L}(\ p(m^i|h^i)\ ) = \beta_u \hat{I}(M^{j\neq i}; Y^i)\ &- \beta_c \hat{I}(H^i; M^i) \\ &- \beta_I \hat{I}(M^i_1; \ldots ; M^i_L | H^i)\end{aligned}$$

where the bounds on mutual information $\hat{I}$ are defined in equations 5.1, 5.2, and 5.7. Overall, our objective is,

$$\begin{aligned}J(\theta) = \max_{\pi} \mathbb{E}\left[\sum_{t\in T}\sum_{i\in N} \gamma_t \mathcal{R}(s_t, a_t) + \mathcal{L}(\ p(m_t|h_t)\ )\right] \\ \text{s.t.}(a_t, m_t, h_t) \sim \pi^i, s_t \sim \mathcal{T}(s_{t-1})\end{aligned}$$

5.4 Complexity through Compositional Communication

We aim to satisfy the complexity objective, $I(H^i, M^i)$, through compositional communication. In order to induce complexity in our communication, we want the messages to be as non-random as possible. That is, informative with respect to the input hidden state h. In addition, we want each token within the message to share as little information as possible with the preceding tokens. Thus, each additional token adds *only informative* content. Each token has a fixed length in bits W. The total sequence is limited by a fixed limit, $\sum_l^L W_l \leq S$, of S bits and a total of L tokens.

We use a variational message generation setup, which maps the encoded hidden state h to a message m; that is, we are modeling the posterior, $\pi^i_m(m_l|h)$. We limit the vocabulary size to K tokens, $e_j \in \mathbb{R}^D, j \in [1, K] \subset \mathbb{N}$, where each token has dimensionality D and $l \in [1, L] \subset \mathbb{N}$. Each token m_l is sampled from a categorical posterior distribution,

$$\pi^i_m(m_l = e_k|h) = \begin{cases} 1 & \text{for } k = \arg\min_j ||m_l - e_j||_2 \\ 0 & \text{otherwise} \end{cases}$$

such that the message m_l is mapped to the nearest neighbor e_j. A set of these tokens makes a message m. To satisfy the complexity objective, we want to use m^i to well-represent h^i and consist of independently informative m^i_l.

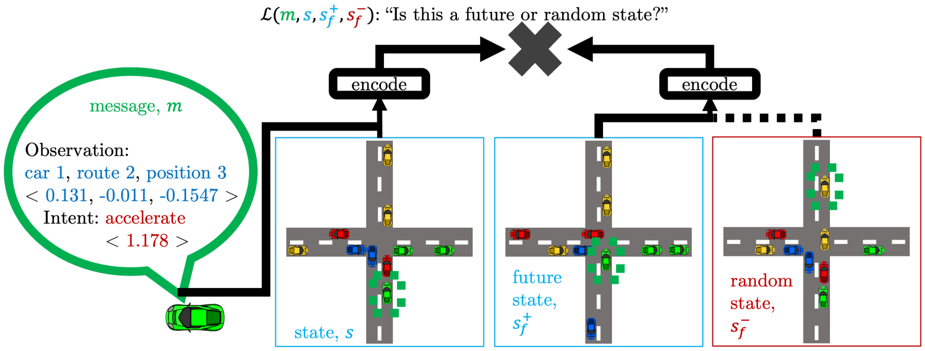


Figure 5.1: By using contrastive learning, our method seeks similar representations between the state-message pair and future states while creating dissimilar representations with random states. Thus satisfying the utility objective of the information bottleneck. The depicted agents are blind and cannot see other cars.

5.4.1 Independent Information

We derive an upper bound for the interaction information between all tokens.

Proposition 5.4.1. *For the interaction information between all tokens, the following upper bound holds:* $I(m_1; \dots; m_L|h) \leq \mathbb{E}_{h \sim p(h)} \left[D_{KL}\left(q(\hat{m}|h) || \pi_m^i(m_1|h) \otimes \cdots \otimes \pi_m^i(m_L|h)\right)\right]$.

The proof is in Appendix 5.8.

Since we want the mutual information to be minimized in our objective, we minimize,

$$\begin{aligned}&\hat{I}(m_1; \dots; m_L|h) = \\ &\mathbb{E}_{h \sim p(h)} \left[D_{KL}\left(q(\hat{m}|h) || \pi_m^i(m_1|h) \otimes \cdots \otimes \pi_m^i(m_L|h)\right)\right]\end{aligned} \tag{5.1}$$

5.4.2 Input-Oriented Information

In order to induce complexity in the compositional messages, we additionally want to minimize the mutual information $I(H; M)$ between the composed message $\hat{m}$ and the encoded information h. We derive an upper bound on the mutual information that we use as a Lagrangian term to minimize.

Proposition 5.4.2. *For the mutual information between the composed message and encoded information, the following upper bound holds:* $I(H; M) \leq \sum_l^L \mathbb{E}_{h \sim p(h)} \left[D_{KL}\left(q(m_l|h) || z(m_l)\right)\right]$.

The proof is in Appendix 5.8. Thus, we have our Lagrangian term,

$$\hat{I}(H^i, M^i) = \sum_l^L \mathbb{E}_{h \sim p(h)} \left[D_{KL}\left(q(m_l|h) || z(m_l)\right)\right] \tag{5.2}$$

Conditioning on the input or observation data is a decentralized training objective.

5.4.3 Sequence Length

Compositional communication necessitates an adaptive limit on the total length of the sequence.

Algorithm 2 Compositional Message Gen.(h_t)

```
$T \leftarrow$ num_tokens
$m = \mathbf{0}$ $\{T \times d_m, d_m \leftarrow$ token_size$\}$
$Q \leftarrow$ Q_MLP$(h_t)$
$V \leftarrow$ V_MLP$(h_t)$
for $i \leftarrow 1$ to $T$ do
    $K \leftarrow$ K_MLP$(m)$
    $\hat{h} = \texttt{softmin}(\frac{Q^\intercal \texttt{mean}(K,1)}{\sqrt{d_k}})^\intercal V$
    $m_i \sim \mathcal{N}(\hat{h}; \mu, \sigma)$
end for
return $m$
```

Corollary 5.4.3. *Repeat tokens, w, are redundant and can be removed.*

Suppose one predicts two arbitrary tokens, w_k and w_l. Given equation 5.1, it follows that there is low or near-zero mutual information between w_k and w_l.

A trivial issue is that the message generator will predict every available token as to follow the unique token objective. Since the tokens are imbued with input-oriented information (equation 5.2), the predicted tokens will be based on relevant referential details. Thus, it follows that tokens containing irrelevant information will not be chosen.

A nice optimization objective that follows from corollary 5.4.3 is that one can use self-supervised learning with an end-of-sequence (EOS) token to limit the variable total length of compositional message sequences.

$$H(m_{\texttt{EOS}}, m_l) = -\pi(m_{\texttt{EOS}}) \log(\pi(m_l)) \tag{5.3}$$

5.4.4 Message Generation Architecture

Now, we can define the pipeline for message generation. The idea is to create an architecture that can generate features to enable independent message tokens. We expand each compressed token into the space of the hidden state h (1-layer linear expansion) since each token has a natural embedding in $\mathbf{R}^{|h|}$. Then, we perform attention using a `softmin` to help minimize similarity with previous tokens and sample the new token from a variational distribution. See algorithm 2 for complete details. During execution, we can generate messages directly due to equation 5.1, resolving any computation time lost from sequential compositional message generation.

5.5 Utility through Contrastive Learning

First, note that our Markov Network is as follows: $H^j \rightarrow M^j \rightarrow Y^i \leftarrow H^i$. Continue to denote i as the agent identification and j as the agent ID such that $j \neq i$. We aim to satisfy the utility objective of the information bottleneck, $I(M^j; Y^i)$, through contrastive learning as shown in figure 5.1.

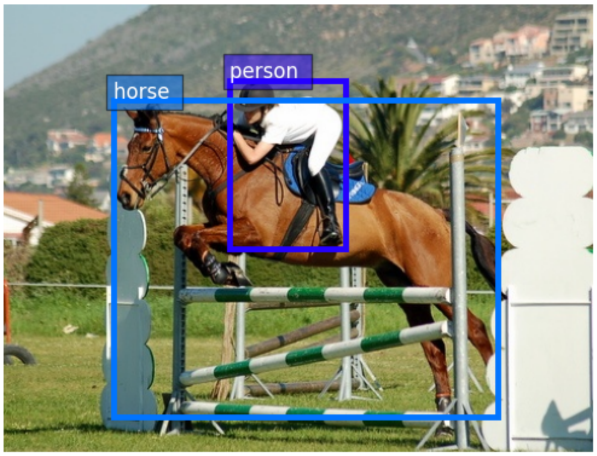


Figure 5.2: An example of two possible classes, person and horse, from a single observation in the Pascal VOC game.

Proposition 5.5.1. *Utility mutual information is lower bounded by the contrastive NCE-binary objective,* $I(M, Y) \geq \log\sigma(f(s, m, s_f^+)) + \log\sigma(1 - f(s, m, s_f^-))$.

The proof is in Appendix 5.8.

This result shows a need for gradient information to flow backward across agents along communication edge connections.

5.6 Experiments and Results

We condition on inputs, especially rich information (such as pixel data), and task-specific information. When evaluating an artificial language in MARL, we are interested in referential tasks, in which communication is *required* to complete the task. With regard to intent-grounded communication, we study ordinal tasks, which require coordination information between agents to complete successfully. Thus, we consider tasks with a team of agents to foster messaging that communicates coordination information that also includes their observations. To test **H1**, structuring emergent messages enables lower complexity, we test our methodology and analyze the input-oriented information and utility capabilities. Next, we analyze the ability of heterogeneous agents to understand differing communication policies (**H2**)). Finally, we consider the effect of social shadowing (**H3**), in which agents solely learn a communication policy from an expert agent's action policy. We additionally analyze the role of offline reinforcement learning for emergent communication in combination with online reinforcement learning to further learn emergent communication alongside an action policy. We evaluate each scenario over 10 seeds.

5.6.1 Environments

Blind Traffic Junction We consider a benchmark that requires both referential and ordinal capabilities within a team of agents. The blind traffic junction environment [72] requires multiple

Table 5.1: Beta ablation: Messages are naturally sparse in bits due to the complexity loss. Redundancy measures the capacity for a bijection between the size of the set of unique tokens and the enumerated observations and intents. Min redundancy is 1.0 (a bijection). Lower is better.

β	Success	Message Size in Bits	Redundancy
0.1	1.0	64	1.0
0.01	.996	69.52	1.06
0.001	.986	121.66	2.06
0	.976	147.96	2.31
non-compositional	.822	512	587

agents to navigate a junction without any observation of other agents. Rather, they only observe their own state location. Ten agents must coordinate to traverse through the lanes without colliding into agents within their lane or in the junction. Our training uses REINFORCE [91].

Pascal VOC Game We further evaluate the complexity of compositional communication with a Pascal VOC [16]. This is a two-agent referential game similar to the Cifar game [54] but requires the prediction of multiple classes. During each episode, each agent observes a random image from the Pascal VOC dataset containing exactly two unique labels. Each agent must encode information given only the raw pixels from the original image such that the other agent can recognize the two class labels in the original image. An agent receives a reward of 0.25 per correctly chosen class label and will receive a total reward of 1 if both agents guess all labels correctly. See figure 5.2. Our training uses heterogeneous agents trained with PPO (modified from MAPPO [93] repository). For simplicity of setup, we consider images with exactly two unique labels from a closed subset of size five labels of the original set of labels from the Pascal VOC data. Furthermore, these images must be of size 375×500 pixels. Thus, the resultant dataset comprised 534 unique images from the Pascal VOC dataset.

5.6.2 Baselines

To evaluate our methodology, we compare our method to the following baselines: (1) `no-comm`, where agents do not communicate; (2) `rl-comm`, which uses a baseline communication method learned solely through policy loss [72]; (3) `ae-comm`, which uses an autoencoder to ground communication in input observations [54]; (4) `VQ-VIB`, which uses a variational autoencoder to ground discrete communication in input observations and a mutual information objective to ensure low entropy communication [82].

5.6.3 Input-Oriented Information Results

We provide an ablation of the loss parameter β in table 5.1 in the blind traffic junction scenario. When $\beta = 0$, we use our compositional message paradigm without our derived loss terms. We find that higher complexity and independence losses increase sample complexity. When $\beta = 1$, the model was unable to converge. However, when there is no regularization loss, the model

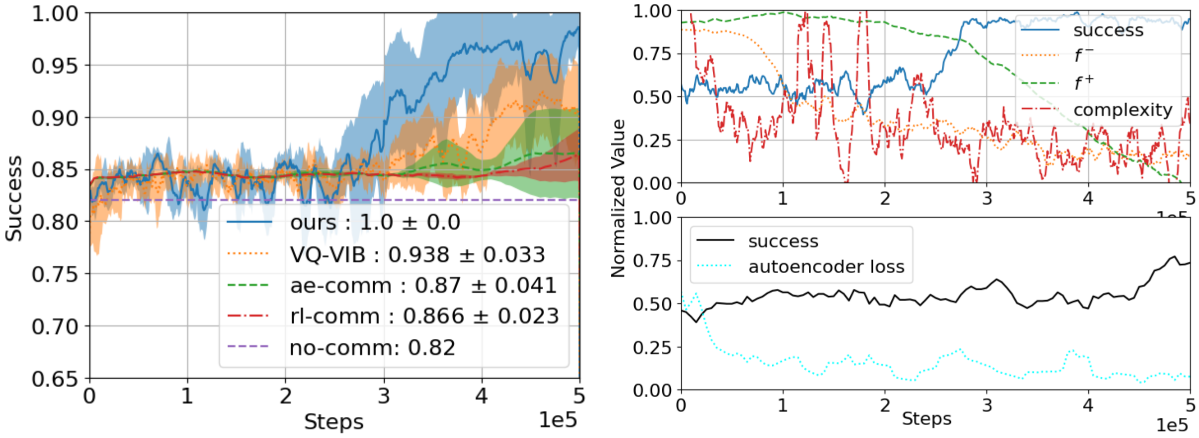


Figure 5.3: **Blind Traffic Junction** Left: Our method uses compositional complexity and contrastive utility to outperform other baselines in terms of performance and sample complexity. The legend provides the mean ± variance of the best performance. Right: Top: success, contrastive, and complexity losses for our method. Right, Bottom: success, autoencoder loss for `ae-comm` with supervised pretraining.

performs worse (with no guarantees about referential representation). We attribute this to the fact that our independence criteria learns a stronger causal relationship. There are fewer spurious features that may cause an agent to take an incorrect action.

In order to understand the effect of the independent concept representation, we analyze the emergent language's capacity for redundancy. A message token m_l is redundant if there exists another token m_k that represents the same information. With our methodology, the emergent 'language' converges to the exact number of observations and intents required to solve the task. With a soft discrete threshold, the independent information loss naturally converges to a discrete number of tokens in the vocabulary. Our β ablation in table 5.1 yields a bijection between each token in the vocabulary and the possible emergent concepts, i.e., the enumerated observations and intents. Thus for $\beta = 0.1$, there is no redundancy.

Sparse Communication In corollary 5.4.3, we assume that there is no mutual information between tokens. In practice, the loss may only be near-zero. Our empirical results yield independence loss around $1e - 4$. In table 5.1, the size of the messages is automatically compressed to the smallest size to represent the information. Despite a trivially small amount of mutual information between tokens, our compositional method is able to reduce the message size in bits by 2.3x using our derived regularization, for a total of an 8x reduction in message size over non-compositional methods such as `ae-comm`. Since the base unit for the token is a 32-bit float, we note that each token in the message may be further compressed. We observe that each token uses three significant digits, which may further compress tokens to 10 bits each for a total message length of 20 bits.

5.6.4 Communication Utility Results

Due to coordination in MARL, grounding communication in referential features is not enough. Finding the communication utility requires grounding messages in ordinal information. Overall, figure 5.3 shows that our compositional, contrastive method outperforms all methods focused on solely input-oriented communication grounding. In the blind traffic junction, our method yields a higher average task success rate and is able to achieve it with a lower sample complexity. Training with the contrastive update tends to spike to high success but not converge, often many episodes before convergence, which leaves area for training improvement. That is, the contrastive update begins to find aligned latent spaces early in training, but it cannot adapt the methodology quickly enough to converge. The exploratory randomness of most of the early online data prevents exploitation of the high utility f^+ examples. This leaves further room for improvement for an adaptive contrastive loss term.

Regularization loss convergence After convergence to high task performance, the autoencoder loss increases in order to represent the coordination information. This follows directly from the information bottleneck, where there exists a tradeoff between utility and complexity. However, communication, especially referential communication, should have an overlap between utility and complexity. Thus, we should seek to make the complexity loss more convex. Our compositional communication complexity loss does not converge before task performance convergence. While the complexity loss tends to spike in the exploratory phase, the normalized value is very small. Interestingly, the method eventually converges as the complexity loss converges below a normalized 0.3. Additionally, the contrastive loss tends to decrease monotonically and converges after the task performance converges, showing a very smooth decrease. The contrastive f^- loss decreases during training, which may account for success spikes prior to convergence. The method is able to converge after only a moderate decrease in the f^+ loss. This implies empirical evidence that the contrastive loss is an optimal critic for messaging. See figure 5.3.

5.6.5 Heterogeneous Alignment Through Communication

In order to test the heterogeneous alignment ability of our methodology to learn higher-order concepts from high-dimensional data, we analyze the performance on the Pascal VOC game. We compare our methodology against `ae-comm` to show that concepts should consist of independent information directly from task signal rather than compression to reconstruct inputs. That is, we show an empirical result on pixel data to verify the premise of the information bottleneck. Our methodology significantly outperforms the observation-grounded `ae-comm` baseline, as demonstrated by figure 5.4. The `ae-comm` methodology, despite using autoencoders to learn observation-grounded communication, performs only slightly better than `no-comm`. On the other hand, our methodology is able to outperform both baselines significantly. It is important to note that based on figure 5.4, our methodology is able to guess more than two of the four labels correctly across the two agents involved, while the baseline methodologies struggle to guess exactly two of thew four labels consistently. This can be attributed to our framework being

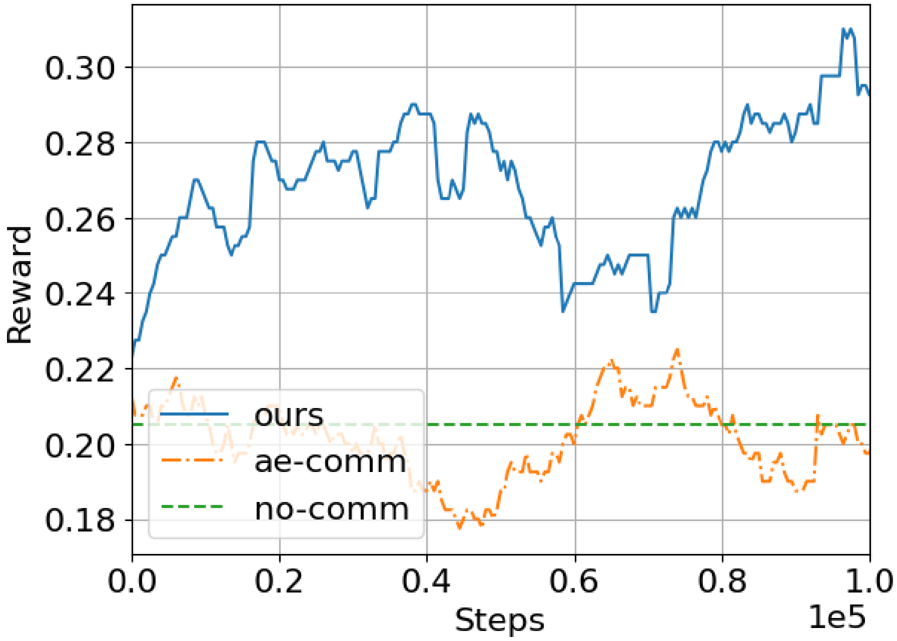


Figure 5.4: **Pascal VOC Game** Representing compositional concepts from raw pixel data in images to communicate multiple concepts within a single image. Our method significantly outperforms `ae-comm` and `no-comm` due to our framework being able to learn composable, independent concepts.

able to learn compositional concepts that are much more easily discriminated due to mutual independence.

5.6.6 Social Shadowing

Critics of emergent communication may point to the increased sample complexity due to the dual communication and action policy learning. In the social shadowing scenario, heterogeneous agents can learn to generate a communication policy without learning the action policy of the watched expert agents. To enable social shadowing, the agent will alternate between a batch of traditional MARL (no expert) and (1st-person) shadowing an expert agent performing the task in its trajectory. The agent only uses the contrastive objective to update its communication policy during shadowing. In figure 5.5, the agent that performs social shadowing is able to learn the action policy with almost half the sample complexity required by the online reinforcement learning agent. Our results show that the structured latent space of the emergent communication learns socially benevolent coordination. This tests our hypothesis that by learning communication to understand the actions of other agents, one can enable lower sample complexity coordination. Thus, it mitigates the issues of solely observing actions.

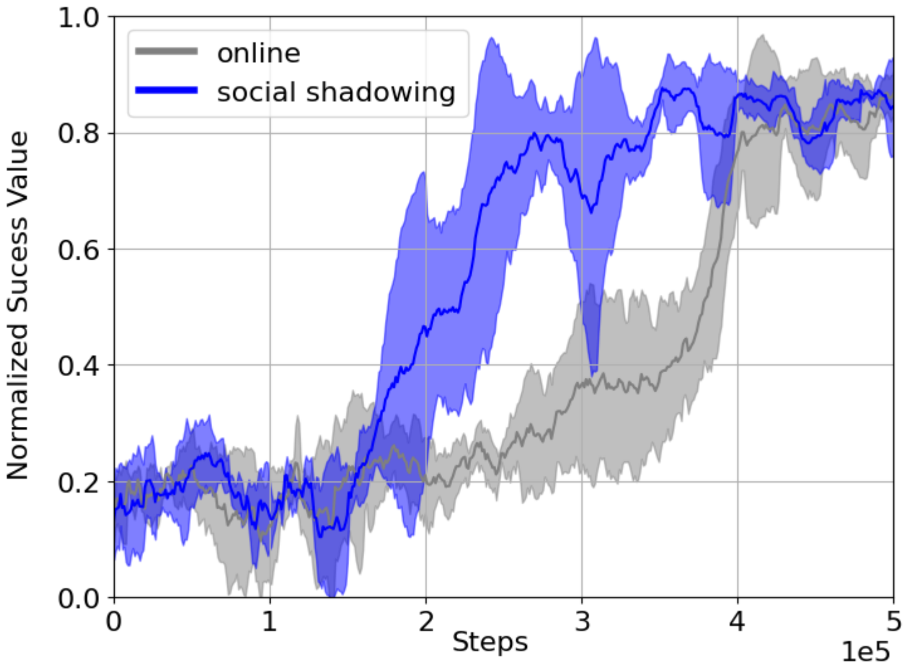


Figure 5.5: **Blind Traffic Junction** Social shadowing enables significantly lower sample complexity when compared to traditional online MARL.

5.7 Discussion

By using our framework to better understand the intent of others, agents can learn to communicate to align policies and coordinate. Any referential-based setup can be performed with a supervised loss, as indicated by the instant satisfaction of referential objectives. Even in the Pascal VOC game, which appears to be a purely referential objective, our results show that intelligent compression is not the only objective of referential communication. The emergent communication paradigm must enable an easy-to-discriminate space for the game. In multi-agent settings, the harder challenge is to enable coordination through communication. Using contrastive communication as an optimal critic aims to satisfy this, and has shown solid improvements. Since contrastive learning benefits from good examples, this method is even more powerful when there is access to examples from expert agents. In this setting, the communication may be bootstrapped, since our optimal critic has examples with strong signals from the 'social shadowing' episodes.

Additionally, we show that the minimization of our independence objective enables tokens that contain minimal overlapping information with other tokens. Preventing trivial communication paradigms enables higher performance. Each of these objectives is complementary, so they are not trivially minimized during training, which is a substantial advantage over comparative baselines. Unlike prior work, this enables the benefits of training with reinforcement learning in multi-agent settings.

In addition to lower sample complexity, the mutual information regularization yields additional benefits, such as small messages, which enables the compression aspect of sparse communication. From a qualitative point of view, the independent information also yields discrete emergent concepts, which can be further made human-interpretable by a post-hoc analysis [92]. This is a step towards white-box machine learning in multi-agent settings. The interpretability of this learned white-box method could be useful in human-agent teaming as indicated by prior work [38]. The work here will enable further results in decision-making from high-dimensional data with emergent concepts. The social scenarios described are a step towards enabling a zero-shot communication policy. This work will serve as future inspiration for using emergent communication to enable ad-hoc teaming with both agents and humans.

5.8 Proofs

Proposition 5.4.1 *For the interaction information between all tokens, the following upper bound holds:* $I(m_1; \ldots; m_L|h) \leq \mathbb{E}_{h \sim p(h)}\left[D_{KL}\left(q(\hat{m}|h)||\pi_m^i(m_1|h) \otimes \cdots \otimes \pi_m^i(m_L|h)\right)\right]$.

Proof. Starting with the independent information objective, we want to minimize the interaction information,

$$\begin{aligned} &I(m_1; \ldots; m_L|h) = \\ &\int \cdots \int f_m(m_1, \ldots, m_L, h) dh\, dm_1 \ldots dm_L \end{aligned}$$

which defines the conditional mutual information between each token and,

$$f_m(*) = p(h)p(m_1; \ldots; m_L|h) \log \frac{p(m_1; \ldots; m_L|h)}{\prod_l^L p(m_l|h)} \tag{5.4}$$

Let $\pi_m^i(m_l|h)$ be a variational approximation of $p(m_l|h)$, which is defined by our message encoder network. Given that each token should provide unique information, we assume independence between m_l. Thus, it follows that our compositional message is a vector, $m = [m_1, \ldots, m_L]$, and is jointly Gaussian. Moreover, we can define $q(\hat{m}|h)$ as a variational approximation to $p(m|h) = p(m_1; \ldots, m_L|h)$. We can model q with a network layer and define its loss as $||\hat{m} - m||_2$. Thus, transforming equation 5.4 into variational form, we have,

$$g_m(m_1, \ldots, m_L, h) = p(h)q(\hat{m}|h) \log \frac{q(\hat{m}|h)}{\prod_l^L \pi_m^i(m_l|h)}$$

Since Kullback-Leibler divergence D_{KL} is non-negative,

$$D_{KL}\left(q(\hat{m}|h)||\pi_m^i(m_1|h) \otimes \cdots \otimes \pi_m^i(m_L|h)\right) \geq 0,$$

it follows that

$$\int q(\hat{m}|h) \log q(\hat{m}|h) d\hat{m} \geq \int q(\hat{m}|h) \log \prod_l^L \pi_m^i(m_l|h) d\hat{m}$$

Thus, we can bound our interaction information,

$$\begin{aligned} I(m_1; \ldots; m_L|h) &\leq \int \cdots \int g_m(*) dh dm_1 \ldots dm_L \\ &= \mathbb{E}_{h \sim p(h)}\left[D_{KL}\left(q(\hat{m}|h)||\pi_m^i(m_1|h) \otimes \cdots \otimes \pi_m^i(m_L|h)\right)\right] \end{aligned}$$

□

Proposition 5.4.2 *For the mutual information between the composed message and encoded information, the following upper bound holds:* $I(H;M) \leq \sum_l^L \mathbb{E}_{h\sim p(h)}\left[D_{KL}\left(q(m_l|h)||z(m_l)\right)\right]$.

Proof. By definition of mutual information between the composed messages M and the encoded observations H, we have,

$$I(H;M) = \int\int p(h)p(\hat{m}|h)\log\frac{p(\hat{m}|h)}{p(\hat{m})}d\hat{m}\,dh$$

Substituting $q(\hat{m}|h)$ for $p(\hat{m}|h)$, the same KL Divergence identity, and defining a Gaussian approximation $z(\hat{m})$ of the marginal distribution $p(\hat{m})$, it follows that,

$$I(H;M) \leq \int\int p(h)q(\hat{m}|h)\log\frac{q(\hat{m}|h)}{z(\hat{m})}d\hat{m}\,dh$$

In expectation of equation 5.1, we have,

$$q(\hat{m}|h) = q(\hat{m}|h) = \prod_l^L \pi_m^i(m_l|h).$$

This implies that, for $\hat{m} = [m_1, \ldots, m_L]$, there is probabilistic independence between $m_j, m_k, j \neq k$. Thus, expanding, it follows that,

$$\begin{aligned} I(H;M) &\leq \sum_l^L \int\int p(h)q(m_l|h)\log\frac{q(m_l|h)}{z(m_l)}dm_l\,dh \\ &= \sum_l^L \mathbb{E}_{h\sim p(h)}\left[D_{KL}\left(q(m_l|h)||z(m_l)\right)\right] \end{aligned}$$

where $z(m_l)$ is a standard Gaussian. □

Proposition 5.5.1. *Utility mutual information is lower bounded by the contrastive NCE-binary objective,* $I(M,Y) \geq \log\sigma(f(s,m,s_f^+)) + \log\sigma(1 - f(s,m,s_f^-))$.

Proof. We suppress the reliance on h since this is directly passed through. By definition of mutual information, we have,

$$I(M^j;Y^i) = \int\int p(m)\pi_{R^+}(y|m)\log\frac{\pi_{R^+}(y|m)}{\pi_{R^-}(y)}dm\,dy$$

Our network model learns $\pi_{R^+}(y|m)$ from rolled-out trajectories, R^+, using our policy. The prior of our network state, $\pi_{R^-}(y)$, can be modeled from rolling out a random trajectory, $R-$. Unfortunately, it is intractable to model $\pi_{R^+}(y|m)$ and $\pi_{R^-}(y)$ directly during iterative learning, but we can sample $y^+ \sim \pi_{R^+}(y|m)$ and $y^- \sim \pi_{R^-}(y)$ directly from our network during training.

It has been shown that $\log p(y|m)$ provides a bound on mutual information [67],

$$I(M^j; Y^i) \geq \mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K} \log \pi_{R^+}(y_k|m_k) + \log \pi_{R^-}(y_k)\right] \tag{5.5}$$

with the expectation over $\prod_l p(m_l, y_l)$. However, we need a tractable understanding of the information Y.

Lemma 5.8.1. $\pi_{R^-}(y) = p(s' = s_f^- | y)$.

In the information bottleneck, Y represents the desired outcome. In our setup, y is coordination information that helps create the desired output, such as any action a^-. This implies, $y \implies a^-$. Since the transition is known, it follows that $a^- \implies s_f^-$, a random future state. Thus, we have, $\pi_{R^-}(y) = p(s' = s_f^- | y)$.

Lemma 5.8.2. $\pi_{R^+}(y|m) = p(s' = s_f^+ | y, m)$.

This is similar to the proof for lemma 5.8.1, but requires assumptions on messages m from the emergent language. We note that when m is random, the case defaults to lemma 5.8.1. Thus, we assume we have at least input-oriented information in m given sufficiently satisfying equation 5.2. Given a sufficient emergent language, it follows that $y \implies a^+$, where a^+ is an intention action based on m. Similarly, since the transition is known, $a^+ \implies s_f^+$, a desired goal state along the trajectory. Thus, we have, $\pi_{R^+}(y|m) = p(s' = s_f^+ | y, m)$.

Recall the following (as shown in [17]), which we have adapted to our communication objective,

Proposition 5.8.3 (rewards → probabilities)**.** *The Q-function for the goal-conditioned reward function $r_g(s_t, m_t) = (1-\gamma)p(s' = s_g|y_t)$ is equivalent to the probability of state s_g under the discounted state occupancy measure:*

$$Q^\pi_{s_g}(s, m) = p^\pi(s_f^+ = s_g | y) \tag{5.6}$$

and

Lemma 5.8.4. *The critic function that optimizes equation 5.5 is a Q-function for the goal-conditioned reward function up to a multiplicative constant* $\frac{1}{p(s_f)}$*:* $\exp(f^*(s, m, s_f) = \frac{1}{p(s_f)} Q^\pi_{s_f}(s, m)$.

The critic function $f(s, m, s_f) = y^\intercal \texttt{enc}(s_f)$ represents the similarity between the encoding $y = \texttt{enc}(s, m)$ and the encoding of the future rollout s_f.

Given lemmas 5.8.1 5.8.2 5.8.4 and proposition 5.8.3, it follows that equation 5.5 is the NCE-binary [56] (InfoMAX [27]) objective,

$$\hat{I}(M^j, Y^i) = \log\left(\sigma(f(s, m, s_f^+))\right) + \log\left(1 - \sigma(f(s, m, s_f^-))\right) \tag{5.7}$$

which lower bounds the mutual information, $I(M^j, Y^i) \geq \hat{I}(M^j, Y^i)$. The critic function is unbounded, so we constrain it to $[0, 1]$ with the sigmoid function, $\sigma(*)$. □

Chapter 6

Conclusion

Communication is necessary for optimal performance of multi-agent teams. We have shown that human interpretability and workload is dependent on the representational power of the emergent communication space. We have proposed a method for multi-agent individualized sparse communication. We reframed sparsity as a representation learning problem through the information bottleneck problem. We have shown that through training a communication-action policy grounded with an autoencoder and analysis during execution of non-sparse messaging, one can exhibit lossless zero-shot sparsity. That is, the sparsity objective may be achieved without any cost of performance with no additional reinforcement learning training. Additionally, we produce individualized regularization to limit performance loss with few-shot sparsity. This allows our model to adhere to messaging constraints in over-constrained bandwidth scenarios. We have also shown that a mutual information objective can be used to shrink the total size of messages by using a compositional setup.

By using our framework to better understand the intent of others, agents can learn to communicate to align policies and coordinate. Any referential-based setup can be performed with a supervised loss, as indicated by the instant satisfaction of referential objectives. Our results show that intelligent compression is not the only objective of referential communication. The emergent communication paradigm must enable an easy-to-discriminate space for the game. In multi-agent settings, the harder challenge is to enable coordination through communication. Using contrastive communication as an optimal critic aims to satisfy this, and has shown solid improvements. Since contrastive learning benefits from good examples, this method is even more powerful when there is access to examples from expert agents. In this setting, the communication may be bootstrapped, since our optimal critic has examples with strong signals from the 'social shadowing' episodes.

Additionally, we show that the minimization of our independence objective enables tokens that contain minimal overlapping information with other tokens. Preventing trivial communication paradigms enables higher performance. Each of these objectives is complementary, so they are not trivially minimized during training, which is a substantial advantage over comparative baselines. Unlike prior work, this enables the benefits of training with reinforcement learning in multi-agent settings.

6.1 Future Work

The future of scalable agent learning will require the use of emergent communication. All multi-agent learning will be learned without the explicit use of any one team. This will allow for social learning of skills from other agents and ad-hoc teaming with any arbitrary agent or human. The decentralized training policy of these multi-agent methods will also enable scalability. The social scenarios described are a step towards enabling a zero-shot communication policy. This work will serve as future inspiration for using emergent communication to enable ad-hoc teaming with both agents and humans. While much of this work is still in its infancy, the results shown in this thesis motivate a clear path towards arbitrary decision-making and teaming through the use of purposeful emergent communication.

Bibliography

[1] A. Agarwal, S. Kumar, K. Sycara, and M. Lewis. Learning transferable cooperative behavior in multi-agent teams. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1741–1743, 2020. 3.2.2, 4.1, 5.3.1

[2] A. Agarwal, G. Swaminathan, V. Sharma, and K. Sycara. Community regularization of visually-grounded dialog. In *Proceedings of the 2019 Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2019. 3.2.3

[3] S. Agrawal. Learning to imitate, adapt and communicate. Master's thesis, Carnegie Mellon University, 2021. 3.1, 4.2.1, 4.2.2, 4.3.1, 4.4

[4] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy. Deep variational information bottleneck. *ICLR*, 2017. 5.3.2

[5] A. Anderson, J. Dodge, A. Sadarangani, Z. Juozapaitis, E. Newman, J. Irvine, S. Chattopadhyay, A. Fern, and M. Burnett. Explaining reinforcement learning to mere mortals: An empirical study. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 1328–1334. International Joint Conferences on Artificial Intelligence Organization, 7 2019. 3.1

[6] J. Andreas, A. Dragan, and D. Klein. Translating neuralese. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 232–242, 2017. 3.1, 3.2.3

[7] C. Berner, G. Brockman, B. Chan, V. Cheung, P. Dębiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019. 3.1

[8] M. Carroll, R. Shah, M. K. Ho, T. Griffiths, S. Seshia, P. Abbeel, and A. Dragan. On the utility of learning about humans for human-ai coordination. *Advances in Neural Information Processing Systems*, 32:5174–5185, 2019. 3.1

[9] D. Chan. The ai that has nothing to learn from humans. *The Atlantic*, 7(1):e1000858, 2017. 3.1

[10] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. 5.3.1

[11] A. Das, T. Gervet, J. Romoff, D. Batra, D. Parikh, M. Rabbat, and J. Pineau. Tarmac: Targeted multi-agent communication. In *International Conference on Machine Learning*, pages 1538–1546. PMLR, 2019. 3.2.2, 4.1

[12] A. Das, S. Kottur, K. Gupta, A. Singh, D. Yadav, J. M. F. Moura, D. Parikh, and D. Batra. Visual dialog. *CoRR*, abs/1611.08669, 2016. 3.2.3

[13] Z. Ding, T. Huang, and Z. Lu. Learning individually inferred communication for multi-agent cooperation. *Advances in Neural Information Processing Systems*, 33:22069–22079, 2020. 4.2.2, 4.5

[14] T. Eccles, Y. Bachrach, G. Lever, A. Lazaridou, and T. Graepel. Biases for emergent communication in multi-agent reinforcement learning. *Advances in neural information processing systems*, 32, 2019. 4.1, 5.1, 5.2.2

[15] B. Ellis, S. Moalla, M. Samvelyan, M. Sun, A. Mahajan, J. N. Foerster, and S. Whiteson. Smacv2: An improved benchmark for cooperative multi-agent reinforcement learning. *arXiv preprint arXiv:2212.07489*, 2022. 1

[16] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 5.6.1

[17] B. Eysenbach, T. Zhang, R. Salakhutdinov, and S. Levine. Contrastive learning as goal-conditioned reinforcement learning. *arXiv preprint arXiv:2206.07568*, 2022. 1, 5.8

[18] X. Fan and J. Yen. Modeling cognitive loads for evolving shared mental models in human–agent collaboration. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 41(2):354–367, 2010. 3.5.1

[19] J. N. Foerster, Y. M. Assael, N. de Freitas, and S. Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 2145–2153, 2016. 1, 3.2.1, 4.1, 4.2.1, 4.3, 5.1

[20] B. Freed, R. James, G. Sartoretti, and H. Choset. Sparse discrete communication learning for multi-agent cooperation through backpropagation. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7993–7998. IEEE, 2020. 1, 3.2.2

[21] B. Freed, R. James, G. Sartoretti, and H. Choset. Sparse discrete communication learning for multi-agent cooperation through backpropagation. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7993–7998, 2020. 4.1, 4.2.1, 5.1

[22] B. Freed, G. Sartoretti, and H. Choset. Simultaneous policy and discrete communication learning for multi-agent cooperation. *IEEE Robotics and Automation Letters*, 5(2):2498–2505, 2020. 4.1

[23] A. Goyal, Y. Bengio, M. Botvinick, and S. Levine. The variational bandwidth bottleneck: Stochastic evaluation on an information budget. *arXiv preprint arXiv:2004.11935*, 2020. 3.2.2

[24] D. M. Green and J. A. Swets. *Signal Detection Theory and Psychophysics*. Wiley, New York, 1966. 3.1

[25] S. Gronauer and K. Diepold. Multi-agent deep reinforcement learning: a survey. *Artificial*

Intelligence Review, pages 1–49, 2021. 3.1

[26] N. A. Grupen, D. D. Lee, and B. Selman. Multi-agent curricula and emergent implicit signaling. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, pages 553–561, 2022. 5.2.1

[27] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*, 2018. 5.8

[28] G. Hu, Y. Zhu, D. Zhao, M. Zhao, and J. Hao. Event-triggered communication network with limited-bandwidth constraint for multi-agent reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–13, 2021. 4.2.2

[29] D. Hughes, A. Agarwal, Y. Guo, and K. Sycara. Inferring non-stationary human preferences for human-agent teams. In *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pages 1178–1185. IEEE, 2020. 3.2.4

[30] R. Iyer, Y. Li, H. Li, M. Lewis, R. Sundar, and K. Sycara. Transparency and explanation in deep reinforcement learning neural networks. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 144–150, 2018. 3.1

[31] N. Jaques, A. Lazaridou, E. Hughes, C. Gulcehre, P. Ortega, D. Strouse, J. Z. Leibo, and N. De Freitas. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In *International conference on machine learning*, pages 3040–3049. PMLR, 2019. 3.2.4, 5.1, 5.2.1

[32] J. Jiang and Z. Lu. Learning attentional communication for multi-agent cooperation. *Advances in Neural Information Processing Systems*, 31:7254–7264, 2018. 4.2.1

[33] Y. Jiang, K. Zhang, Q. Li, J. Chen, and X. Zhu. Multi-agent path finding via tree lstm. *arXiv preprint arXiv:2210.12933*, 2022. 1

[34] S. Karten, S. Kailas, H. Li, and K. Sycara. On the role of emergent communication for social learning in multi-agent reinforcement learning. *arXiv preprint arXiv:2302.14276*, 2023. 1.1

[35] S. Karten, S. Kailas, and K. Sycara. Emergent compositional concept communication through mutual information in multi-agent teams. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, pages 2391–2393, 2023. 1.1

[36] S. Karten, A. Sivaramakrishnan, E. Granados, T. McMahon, and K. E. Bekris. Data-efficient learning of high-quality controls for kinodynamic planning used in vehicular navigation. *Workshop on Machine Learning for Motion Planning at IEEE International Conference on Robotics and Automation (ICRA)*, 2021. 1

[37] S. Karten, M. Tucker, S. Kailas, and K. Sycara. Towards true lossless sparse communication in multi-agent systems. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7191–7197. IEEE, 2023. 1.1, 5.1, 5.2.2, 5.3.1

[38] S. Karten, M. Tucker, H. Li, S. Kailas, M. Lewis, and K. Sycara. Interpretable learned emergent communication for human-agent teams. *IEEE Transactions on Cognitive and Developmental Systems*, pages 1–1, 2023. 1.1, 4.1, 4.2.2, 5.1, 5.7

[39] D. Kim, S. Moon, D. Hostallero, W. J. Kang, T. Lee, K. Son, and Y. Yi. Learning to schedule communication in multi-agent reinforcement learning. In *ICLR 2019: International Conference on Representation Learning*. International Conference on Representation Learning, 2019. 3.2.2

[40] T. Kliegr, Štěpán Bahník, and J. Fürnkranz. A review of possible effects of cognitive biases on interpretation of rule-based machine learning models. *Artificial Intelligence*, 295:103458, 2021. 3.1

[41] S. Krishnan, N. Jaques, S. Omidshafiei, D. Zhang, I. Gur, V. J. Reddi, and A. Faust. Multi-agent reinforcement learning for microprocessor design space exploration. *arXiv preprint arXiv:2211.16385*, 2022. 1

[42] S. Kullback. *Information theory and statistics*. Courier Corporation, 1997. 2.2, 5.3.2

[43] B. M. Lake, T. Linzen, and M. Baroni. Human few-shot learning of compositional instructions. *arXiv preprint arXiv:1901.04587*, 2019. 3.1, 5.1

[44] A. Lazaridou and M. Baroni. Emergent multi-agent communication in the deep learning era. *arXiv preprint arXiv:2006.02419*, 2020. 4.2.1, 5.1

[45] A. Lazaridou, A. Peysakhovich, and M. Baroni. Multi-agent cooperation and the emergence of (natural) language. *arXiv preprint arXiv:1612.07182*, 2016. 1, 3.2.3, 4.1, 5.1

[46] Y. LeCun, Y. Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995. 5.3.1

[47] S. Levine, C. Finn, T. Darrell, and P. Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016. 1

[48] D. Lewis. *Convention*. Harvard University Press, Cambridge, MA, 1969. 2.3, 4.3.1, 5.1

[49] H. Li, T. Ni, S. Agrawal, F. Jia, S. Raja, Y. Gui, D. Hughes, M. Lewis, and K. Sycara. Individualized mutual adaptation in human-agent teams. *IEEE Transactions on Human-Machine Systems*, 51(6):706–714, 2021. 3.2.4

[50] H. Li, I. Oguntola, D. Hughes, M. Lewis, and K. Sycara. Theory of mind modeling in search and rescue teams. In *2022 31st IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pages 483–489. IEEE, 2022. 2.1.1, 5.1

[51] H. Li, K. Zheng, M. Lewis, D. Hughes, and K. Sycara. Human theory of mind inference in search and rescue tasks. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 65, pages 648–652. SAGE Publications Sage CA: Los Angeles, CA, 2021. 2.1.1

[52] S. Li, W. Sun, and T. Miller. Communication in human-agent teams for tasks with joint action. In *International Workshop on Coordination, Organizations, Institutions, and Norms in Agent Systems*, pages 224–241. Springer, 2015. 3.2.4, 3.5.1

[53] S. Li, Y. Zhou, R. Allen, and M. J. Kochenderfer. Learning emergent discrete message communication for cooperative reinforcement learning. *arXiv preprint arXiv:2102.12550*, 2021. 3.1, 4.2.1, 5.2.1

[54] T. Lin, J. Huh, C. Stauffer, S. N. Lim, and P. Isola. Learning to ground multi-agent com-

munication with autoencoders. *Advances in Neural Information Processing Systems*, 34, 2021. 4.1, 5.1, 5.2.2, 5.6.1, 5.6.2

[55] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6382–6393, 2017. 3.2.3, 4.1, 5.1

[56] Z. Ma and M. Collins. Noise contrastive estimation and negative sampling for conditional models: Consistency and statistical efficiency. In *EMNLP*, 2018. 5.8

[57] H. Mao, Z. Zhang, Z. Xiao, Z. Gong, and Y. Ni. Learning agent communication under limited bandwidth by message pruning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5142–5149, 2020. 3.2.2, 4.1

[58] A. R. Marathe, K. E. Schaefer, A. W. Evans, and J. S. Metcalfe. Bidirectional communication for effective human-agent teaming. In *International Conference on Virtual, Augmented and Mixed Reality*, pages 338–350. Springer, 2018. 3.1, 3.2.4, 5.1

[59] S. L. Marlow, C. N. Lacerenza, J. Paoletti, C. S. Burke, and E. Salas. Does team communication represent a one-size-fits-all approach?: A meta-analysis of team communication and performance. *Organizational behavior and human decision processes*, 144:145–170, 2018. 3.1, 3.6, 5.1

[60] R. Mirsky, W. Macke, A. Wang, H. Yedidsion, and P. Stone. A penny for your thoughts: The value of communication in ad hoc teamwork. *Good Systems-Published Research*, 2020. 5.1

[61] I. Mordatch and P. Abbeel. Emergence of grounded compositional language in multi-agent populations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 3.2.3

[62] K. K. Ndousse, D. Eck, S. Levine, and N. Jaques. Emergent social learning via multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 7991–8004. PMLR, 2021. 5.1, 5.2.1

[63] S. Nikolaidis and J. Shah. Human-robot cross-training: Computational formulation, modeling and evaluation of a human team training strategy. In *2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 33–40, 2013. 3.1

[64] G. Papoudakis, F. Christianos, A. Rahman, and S. V. Albrecht. Dealing with non-stationarity in multi-agent deep reinforcement learning. *arXiv preprint arXiv:1906.04737*, 2019. 5.1

[65] P. Peng, Q. Yuan, Y. Wen, Y. Yang, Z. Tang, H. Long, and J. Wang. Multiagent bidirectionally-coordinated nets for learning to play starcraft combat games. *CoRR*, abs/1703.10069, 2017. 3.1

[66] J. C. Peterson, J. T. Abbott, and T. L. Griffiths. Adapting deep network features to capture psychological representations: An abridged report. In *IJCAI*, pages 4934–4938, 2017. 3.2.3

[67] B. Poole, S. Ozair, A. Van Den Oord, A. Alemi, and G. Tucker. On variational bounds of mutual information. In *International Conference on Machine Learning*, pages 5171–5180. PMLR, 2019. 5.8

[68] T. Rashid, M. Samvelyan, C. Schroeder, G. Farquhar, J. Foerster, and S. Whiteson. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 4295–4304. PMLR, 2018. 3.2.2

[69] M. Samvelyan, T. Rashid, C. S. De Witt, G. Farquhar, N. Nardelli, T. G. Rudner, C.-M. Hung, P. H. Torr, J. Foerster, and S. Whiteson. The starcraft multi-agent challenge. *arXiv preprint arXiv:1902.04043*, 2019. 1, 5.1

[70] E. Seraj, Z. Wang, R. Paleja, D. Martin, M. Sklar, A. Patel, and M. Gombolay. Learning efficient diverse communication for cooperative heterogeneous teaming. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, pages 1173–1182, 2022. 3.2.2

[71] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis. Mastering the game of go without human knowledge. *Nature*, 550:354–, Oct. 2017. 3.1

[72] A. Singh, T. Jain, and S. Sukhbaatar. Learning when to communicate at scale in multiagent cooperative and competitive tasks. In *International Conference on Learning Representations*, 2018. 3.1, 3.3.2, 3.3.3, 4.1, 4.2.1, 4.2.2, 4.4, 4.5, 5.1, 5.6.1, 5.6.2

[73] H. C. Siu, J. Peña, E. Chen, Y. Zhou, V. Lopez, K. Palko, K. Chang, and R. Allen. Evaluation of human-ai teams for learned and rule-based agents in hanabi. *Advances in Neural Information Processing Systems*, 34, 2021. 3.1

[74] A. Sivaramakrishnan, E. Granados, S. Karten, T. McMahon, and K. E. Bekris. Improving kinodynamic planners for vehicular navigation with learned goal-reaching controllers. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9038–9043. IEEE, 2021. 1

[75] S. Sokota, C. A. S. De Witt, M. Igl, L. M. Zintgraf, P. Torr, M. Strohmeier, Z. Kolter, S. Whiteson, and J. Foerster. Communicating via markov decision processes. In *International Conference on Machine Learning*, pages 20314–20328. PMLR, 2022. 5.2.1

[76] A. Soltani, P. Khorsand, C. Guo, and J. Liu. Neural substrates of cognitive biases during probabilistic inference. *Nature Communications*, 7(1):e1000858, 2016. 3.1

[77] S. Sukhbaatar, R. Fergus, et al. Learning multiagent communication with backpropagation. *Advances in neural information processing systems*, 29:2244–2252, 2016. 3.1, 3.2.1, 4.1, 4.2.1, 4.5, 5.1

[78] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, 2nd edition, 2018. 2.1

[79] N. Tishby and N. Zaslavsky. Deep learning and the information bottleneck principle. In *2015 ieee information theory workshop (itw)*, pages 1–5. IEEE, 2015. 1, 4.1, 4.3.1, 4.4.1, 5.2.2, 5.3.2

[80] D. Tse, R. F. Langston, M. Kakeyama, I. Bethus, P. A. Spooner, E. R. Wood, M. P. Witter, and R. G. Morris. Schemas and memory consolidation. *Science*, 316(5821):76–82, 2007. 3.2.3

[81] M. Tucker, H. Li, S. Agrawal, D. Hughes, K. Sycara, M. Lewis, and J. A. Shah. Emergent discrete communication in semantic spaces. *Advances in Neural Information Processing Systems*, 34, 2021. 3.1, 3.2.2, 3.2.3, 3.3.1, 4.2.1, 5.2.2, 5.2.3

[82] M. Tucker, J. Shah, R. Levy, and N. Zaslavsky. Towards human-agent communication via the information bottleneck principle. *arXiv preprint arXiv:2207.00088*, 2022. 4.6.2, 5.1, 5.2.3, 5.6.2

[83] A. Van Den Oord, O. Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 5.2.3

[84] J. J. Van Merrienboer and J. Sweller. Cognitive load theory and complex learning: Recent developments and future directions. *Educational psychology review*, 17(2):147–177, 2005. 3.1

[85] W. Van Winsum. The effects of cognitive and visual workload on peripheral detection in the detection response task. *Human factors*, 60(6):855–869, 2018. 3.6

[86] E. M. van Zoelen, A. Cremers, F. P. Dignum, J. van Diggelen, and M. M. Peeters. Learning to communicate proactively in human-agent teaming. In *International Conference on Practical Applications of Agents and Multi-Agent Systems*, pages 238–249. Springer, 2020. 3.2.4

[87] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 5.3.1

[88] V. K. Vijay, H. Sheikh, S. Majumdar, and M. Phielipp. Minimizing communication while maximizing performance in multi-agent reinforcement learning. *arXiv preprint arXiv:2106.08482*, 2021. 3.2.2, 4.2.2

[89] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, J. Oh, D. Horgan, M. Kroiss, I. Danihelka, A. Huang, L. Sifre, T. Cai, J. P. Agapiou, M. Jaderberg, A. S. Vezhnevets, R. Leblond, T. Pohlen, V. Dalibard, D. Budden, Y. Sulsky, J. Molloy, T. L. Paine, C. Gulcehre, Z. Wang, T. Pfaff, Y. Wu, R. Ring, D. Yogatama, D. Wünsch, K. McKinney, O. Smith, T. Schaul, T. P. Lillicrap, K. Kavukcuoglu, D. Hassabis, C. Apps, and D. Silver. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, pages 1–5, 2019. 3.1

[90] R. Wang, X. He, R. Yu, W. Qiu, B. An, and Z. Rabinovich. Learning efficient multi-agent communication: An information bottleneck approach. In *International Conference on Machine Learning*, pages 9908–9918. PMLR, 2020. 1, 3.1, 3.2.2, 4.1, 5.1, 5.2.2

[91] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992. 2.1.2, 3.3.1, 4.5, 5.3.1, 5.6.1

[92] C.-K. Yeh, B. Kim, and P. Ravikumar. Human-centered concept explanations for neural networks. In *Neuro-Symbolic Artificial Intelligence: The State of the Art*, pages 337–352. IOS Press, 2021. 5.7

[93] C. Yu, A. Velu, E. Vinitsky, Y. Wang, A. Bayen, and Y. Wu. The surprising effectiveness of ppo in cooperative, multi-agent games. *arXiv preprint arXiv:2103.01955*, 2021. 5.3.1,

5.6.1

[94] K. Zhang, Z. Yang, and T. Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of Reinforcement Learning and Control*, pages 321–384, 2021. 3.1