Towards True Lossless Sparse Communication in Multi-Agent Systems

Anonymous Author(s)
Affiliation
Address
email

Abstract

Communication enables agents to cooperate to achieve their goals. Learning *when* to communicate, i.e., *sparse* (in time) communication, and *whom* to message is particularly important when bandwidth is limited. Recent work in learning sparse individualized communication, however, 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 this paper, 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.

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 [6, 20, 8, 17, 13]. Sparse multi-agent communication (wherein agents communicate during only some time-steps) has been shown to be an effective solution to internet packet routing [18], multi-robot navigation [7], complex multiplayer online games such as StarCraft [20, 19, 10, 6], and human-agent teaming [11]. 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 [20, 19, 3, 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.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

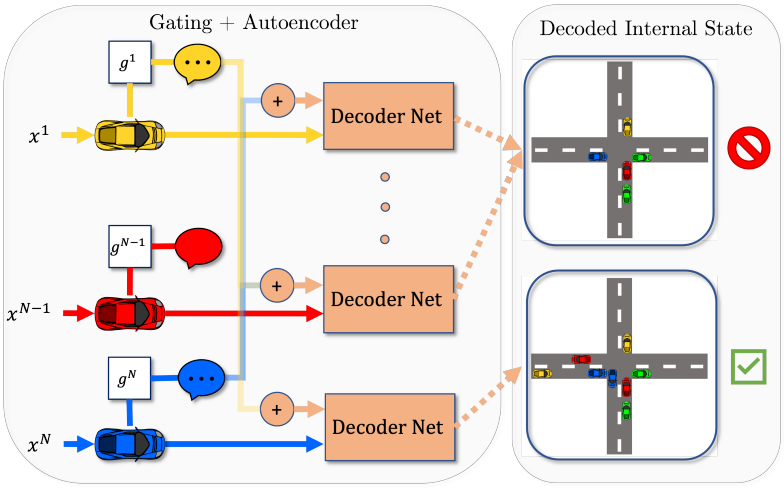

Figure 1: Overview of our multi-agent architecture with gated sparse, informative communication. At every timestep, each agent receives an occluded observation x. Each agent creates a communication message, which is passed to the learned gating function g as well as the `Decoder`. The gating function 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.

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 [5, 16]. 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 [21] may adequately address sparsity constraints [25], 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 work, 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 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.

2 Related Work

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 [10, 20, 19]. 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 [12, 15, 6, 2, 7], 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 [22]. 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.

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 which 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 [19, 9]. In reward-based sparse communication [11, 24], 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 [2]. Rather than building the objective into the reward, I2C [4] 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.

3 Preliminaries

We formulate our setup as a centralized training, decentralized execution (CTDE) [6], 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 $\mathcal{T} : \mathcal{S} \times \mathcal{A}_1, ..., \mathcal{A}_N \rightarrow \mathcal{S}$. Ω defines the set of observations in our partially observable setting. The partial observability requires communication to complete the tasks successfully. $\mathcal{O}_i : \mathcal{M}_1, ..., \mathcal{M}_N \times \mathcal{S} \rightarrow \Omega$ 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.

3.1 The Sparsity Objective

The multi-agent emergent communication problem is phrased as a combination of a Lewis game [14] and the information bottleneck [21]. 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,

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

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

Definition 3.1 (Lossless Sparse Communication)**.** A communication policy π_m is lossless and sparse iff it satisfies the objective in equation 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,

$$\begin{aligned} &\max_{\pi:\mathcal{S}\rightarrow\mathcal{A}\times\mathcal{M}} \mathbb{E}\left[\sum_{t\in\mathcal{T}}\sum_{i\in N}\gamma\mathcal{R}(s_t,a_t)-\lambda s(m_t)\right] \\ &\text{s.t. } (a_t,m_t)\sim\pi, s_t\sim\mathcal{T}(s_{t-1}), m_{AVG}<b \end{aligned} \tag{2}$$

However, depending on the Lagrange multiplier, the objective in equation 1 is not always the same as equation 2. Due to the dual-objective, equation 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 [2].

Definition 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 2.

Thus, in our methodology, we cannot directly optimize equation 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 4.1), which allows for our messages to have independent referential information. Then, given a communication policy which adequately solves the task, one can determine the ordinal utility of each token. By removing unnecessary tokens, we can satisfy the objective in equation 1. Thus, in our methodology, we emphasize learning emergent communication with properties that enable sparse communication with lower optimal budgets b^* (lossless sparsity).

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 [19, 2], 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.

4.1 Sparsity through Information Maximization

The information bottleneck principle [21] 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

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**
    $x^i \sim \mathcal{S}$
    $h^i \leftarrow \texttt{GRU}(x^i)$
    if `useDiscreteMessaging` **then**
      $m^i \leftarrow \texttt{DiscreteProtoNet}(h^i)$
    else
      $m^i \leftarrow h^i$
    end if
    $\texttt{SendMessages}(m^i \odot g(h^i))$
    $\bar{m}^i \leftarrow \texttt{AggregateMessages}()$
    $\tilde{h}^i \leftarrow \texttt{GRU}(\{h^i, \bar{m}^i\})$
    $a^i, v^i, \tilde{x}^i \leftarrow \pi(\tilde{h}^i), V(\tilde{h}^i), \texttt{DecoderNet}(\tilde{h}^i)$
    $L \leftarrow \pi\texttt{Loss}(a^i, v^i) + \mathcal{L}_1(x, \tilde{x}^i) + \mathcal{L}_2(m^i_{AVG})$
  end for
end while

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 $s_t = \{x_t^1, \ldots, x_t^N\}$ 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.

$$\mathcal{L}_1(\theta) = \lambda_1 U(s_t, s_t^{i,\texttt{decoded}}) \tag{3}$$

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 4.1 (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 5.

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,

$$\mathcal{L}_2(\theta) = \lambda_2 \left\| m^i_{AVG} - (b + (1 - b^*)) \right\|_2^2 \tag{4}$$

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. 3. We finetune our model using the regularization penalty in Eq. 4 to observe overconstrained budgets, thus exhibiting few-shot sparsity.

5 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 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. 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 A.

5.1 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 2, 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 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 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, we effectively remove null messages and decrease mutual

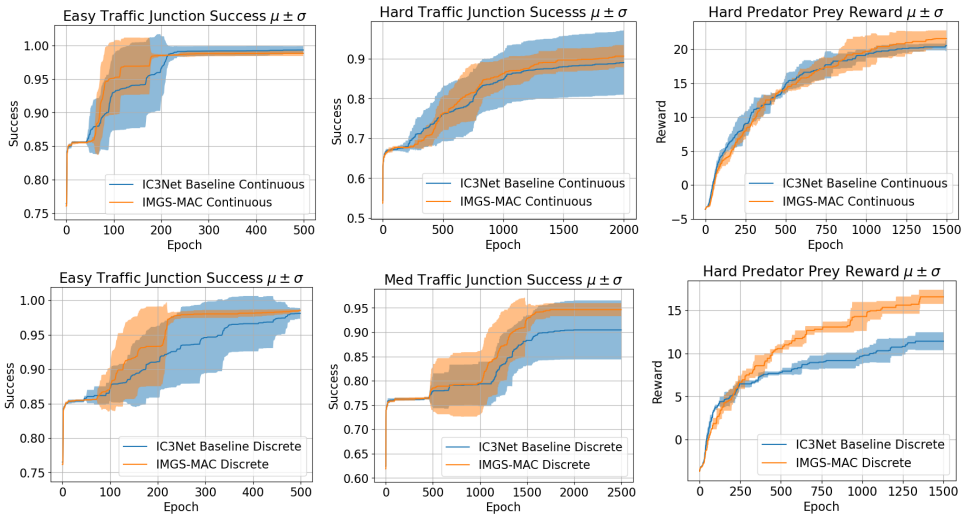

Figure 2: Left and middle figures compare the training IC3Net (blue) vs. our IMGS-MAC (orange) with non-sparse communication ($b = 1$) in Traffic Junction. Our method converges to higher success earlier and with less variance. Right figures compare in Predator-Prey. Our method converges to higher success earlier and with less variance. Top figures use continuous communication vectors while bottom figures use discrete.

Table 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

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.

5.2 Sparsity Analysis

5.2.1 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

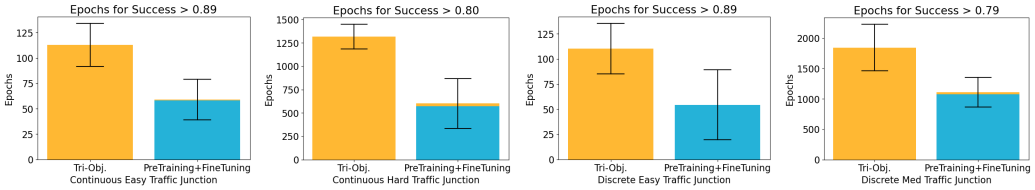

Figure 3: Average success and 95% confidence interval for Tri-objective (left bar, orange) vs. Pretraining with non-sparse $b = 1$ (blue), then Finetuning (orange) with $b = 0.7$. The Pretraining+Finetuning paradigm takes half the amount of training as the Tri-objective.

Figure 4: 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).

Table 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

train the gating function (with the rest of the network) at any $b < b^*$. In Figure 3, 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. 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, we observe that our model only needs a few dozen epochs to converge to a communication budget and is able to safely reduce 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 5.

5.2.2 Zero-shot Sparsity

We use sparsity through information maximization in section 4.1 to reduce the number and usage of null prototypes. In Table 1, one can see that through our analysis, we are able to remove 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 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

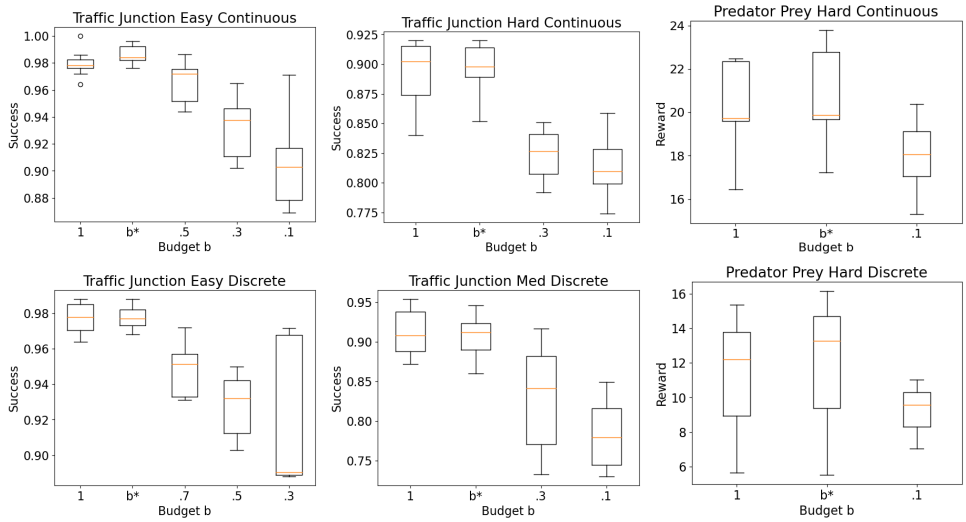

Figure 5: Success versus budget for IMGS-MAC at baseline non-sparse $b = 1$, lossless $b = b^*$, and suboptimal $b < b^*$. Our model provides lossless performance for $b = b^*$ for b^* in Table 2 as compared with the baseline non-sparse $b = 1$. Our performance tapers for smaller budgets until it approaches the no communication performance. Top: continuous communication vectors; Bottom: discrete; Left, middle: Traffic Junction; Right: Predator-Prey.

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 [23]. 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 5, which uses the lossless budget b^* as reported in Table 2. We 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.

6 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.

References

[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. 1

[2] S. Agrawal. Learning to imitate, adapt and communicate. Master's thesis, Carnegie Mellon University, 2021. 2.1, 2.2, 3.1, 4

[3] 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. 1

[4] 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. 2.2, A

[5] 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. 1

[6] 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, 2.1, 3

[7] 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. 1, 2.1

[8] 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. 1

[9] 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. 2.2

[10] J. Jiang and Z. Lu. Learning attentional communication for multi-agent cooperation. *Advances in Neural Information Processing Systems*, 31:7254–7264, 2018. 1, 2.1

[11] S. Karten, M. Tucker, H. Li, S. Kailas, M. Lewis, and K. Sycara. Interpretable learned emergent communication for human-agent teams. *preprint*, 2022. 1, 2.2

[12] A. Lazaridou and M. Baroni. Emergent multi-agent communication in the deep learning era. *arXiv preprint arXiv:2006.02419*, 2020. 2.1

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

[14] D. Lewis. *Convention*. Harvard University Press, Cambridge, MA, 1969. 3.1

[15] 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. 2.1

[16] T. Lin, J. Huh, C. Stauffer, S. N. Lim, and P. Isola. Learning to ground multi-agent communication with autoencoders. *Advances in Neural Information Processing Systems*, 34, 2021. 1

[17] 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. 1

[18] 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. 1

[19] 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. 1, 2.1, 2.2, 4, A

[20] S. Sukhbaatar, R. Fergus, et al. Learning multiagent communication with backpropagation. *Advances in neural information processing systems*, 29:2244–2252, 2016. 1, 2.1, A

[21] 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, 3.1, 4.1

[22] 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. 2.1

[23] M. Tucker, J. Shah, R. Levy, and N. Zaslavsky. Towards human-agent communication via the information bottleneck principle. *arXiv preprint arXiv:2207.00088*, 2022. 5.2.2

[24] 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. 2.2

[25] 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

[26] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992. A