Learning Efficient Multi-agent Communication: An Information Bottleneck Approach

Rundong Wang [1] **Xu He** [1] **Runsheng Yu** [1] **Wei qiu** [1] **Bo An** [1] **Zinovi Rabinovich** [1]

Abstract

We consider the problem of the limited-bandwidth communication for multi-agent reinforcement learning, where agents cooperate with the assistance of a communication protocol and a scheduler. The protocol and scheduler jointly determine *which* agent is communicating *what* message and to *whom*. Under the limited bandwidth constraint, a communication protocol is required to generate informative messages. Meanwhile, an unnecessary communication connection should not be established because it occupies limited resources in vain. In this paper, we develop an *Informative Multi-Agent Communication* (IMAC) method to learn efficient communication protocols as well as scheduling. First, from the perspective of communication theory, we prove that the limited bandwidth constraint requires low-entropy messages throughout the transmission. Then inspired by the information bottleneck principle, we learn a valuable and compact communication protocol and a weight-based scheduler. To demonstrate the efficiency of our method, we conduct extensive experiments in various cooperative and competitive multi-agent tasks with different numbers of agents and different bandwidths. We show that IMAC converges faster and leads to efficient communication among agents under the limited bandwidth as compared to many baseline methods.

1. Introduction

Multi-agent reinforcement learning (MARL) has long been a go-to tool in complex robotic and strategic domains (RoboCup, 2019; OpenAI, 2019). In these scenarios, communicated information enables action and belief correlation that benefits a group's cooperation. Therefore, many recent works in the field of multi-agent communication focus on learning what messages (Foerster et al., 2016; Sukhbaatar et al., 2016; Peng et al., 2017) to send and whom to address them (Jiang & Lu, 2018; Kilinc & Montana, 2018; Das et al., 2019; Singh et al., 2018).

A key difficulty, faced by a group of learning agents in such domains, is the need to efficiently exploit the available communication resources, such as limited bandwidth. The limited bandwidth exists in two processes of transmission: from agents to the scheduler and from the scheduler to agents as shown in Fig. 1. This problem has recently attracted attention and one strategy has been proposed for limited bandwidth settings: downsizing the communication group via a scheduler (Zhang & Lesser, 2013; Kim et al., 2019; Mao et al., 2019). The scheduler allows a part of agents to communicate so that the bandwidth is not overwhelmed with all agents' messages. However, these methods limit the number of agents who can communicate instead of the communication content. Agents may share redundant messages which are unsustainable under bandwidth limitations. For example, a single large message can occupy the whole bandwidth. Also, these methods need specific configurations such as a predefined scale of agents' communication group (Zhang & Lesser, 2013; Kim et al., 2019) or a predefined threshold for muting agents (Mao et al., 2019). Such manual configuration would be of a definite detriment in complex multi-agent domains.

In this paper, we address the limited bandwidth problem by compressing the communication messages. First, from the perspective of communication theory, we view the messages as random vectors and prove that a limited bandwidth can be translated into a constraint on the communicated message entropy. Thus, agents should generate low-entropy messages to satisfy the limited bandwidth constraint. In more details, derived from source coding theorem (Shannon, 1948) and Nyquist criterion (Freeman, 2004), we state that in a noiseless channel, when a K-ary, bandwidth B, quantization interval Δ communication system transmits n messages of dimension d per second, the entropy of the messages $H(\boldsymbol{m})$ is limited by the bandwidth according to $H(\boldsymbol{m}) \leq \frac{2B\log_2 K}{n} + d\log_2 \Delta$.

[1]School of Computer Science and Engineering, Nanyang Technological University, Singapore. Correspondence to: Rundong Wang <rundong001@e.ntu.edu.sg>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020.

Moreover, to allow agents to send and receive low-entropy messages with useful and necessary information, we consider the problem of learning communication protocols and learning scheduling. Inspired by the variational information bottleneck method (Tishby et al., 2000; Alemi et al., 2016), we propose a regularization method for learning informative communication protocols, named *Informative Multi-Agent Communication* (IMAC) [1]. Specifically, IMAC applies the variational information bottleneck to the communication protocol by viewing the messages as latent variables and approximating its posterior distribution. By regularizing the mutual information between the protocol's inputs (the input features extracted from agents) and the protocol's outputs (the messages), we learn informative communication protocols, which convey low-entropy and useful messages. Also, by viewing the scheduler as a virtual agent, we learn a weight-based scheduler with the same principle which aggregates compact messages by reweighting all agents' messages.

We conduct extensive experiments in different environments: cooperative navigation, predator-prey and StarCraftII. Results show that IMAC can convey low-entropy messages, enable effective communication among agents under the limited bandwidth constraint, and lead to faster convergence as compared with various baselines.

2. Related Work

Our work is related to prior works in multi-agent reinforcement learning with communication, which mainly focus on two basic problems: who/whom and what to communicate. They are also expressed as the problem of learning scheduling and communication protocols. One line of scheduling methods is to utilize specific networks to learn a weight-based scheduler by reweighting agents' messages, such as bi-direction RNNs in BiCNet (Peng et al., 2017), a self-attention layer in TarMAC (Das et al., 2019). Another line is to introduce various gating mechanisms to determine the groups of communication agents (Jiang & Lu, 2018; Singh et al., 2018; Kim et al., 2019; Kilinc & Montana, 2018; Mao et al., 2019). Communication protocols are often learned in an end-to-end manner with a specific scheduler: from perceptual input (e.g., pixels) to communication symbols (discrete or continuous) to actions (e.g., navigating in an environment) (Foerster et al., 2016; Kim et al., 2019). While some works for learning the communication protocols focus on discrete human-interpretable communication symbols (Lazaridou et al., 2016; Mordatch & Abbeel, 2018), our method learns a continuous communication protocol in an implicit manner (Foerster et al., 2016; Sukhbaatar et al., 2016; Jiang & Lu, 2018; Singh et al., 2018).

[1] Our code is provided in https://github.com/EC2EZ4RD/IMAC

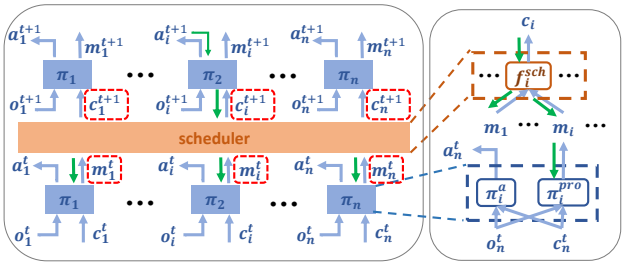

Figure 1. The Architecture of IMAC. **Left:** Overview of the communication scheme. The red dashed box means the communication process with a limited bandwidth constraint. The green line means the gradient flows. **Right:** The upper one is the scheduler for agent i. The below one is the policy π_i^a and the communication protocol network π_i^{pro} for agent i

Methods for addressing the limited bandwidth problem are explored, such as downsizing the communication group via a scheduler. However, all scheduling methods suffer from content redundancy, which is unsustainable under bandwidth limitations. Even if only a single pair of agents is allowed to communicate, a large message may fail to be conveyed due to the limited bandwidth. In addition, scheduling methods with gating mechanisms are inflexible because they introduce manual configuration, such as the predefined size of a communication group (Zhang & Lesser, 2013; Kim et al., 2019), or a handcrafted threshold for muting agents (Jiang & Lu, 2018; Mao et al., 2019). Moreover, most methods for learning communication protocols fail to compress the protocols and extract valuable information for cooperation (Jiang & Lu, 2018). In this paper, we study the limited bandwidth in the aspect of communication protocols. Also, our methods can be extended into the scheduling if we utilize a weight-based scheduler.

The combination between the information bottleneck method and reinforcement learning has brought a few applications in the last few years, especially in imitation learning (Peng et al., 2018), inverse reinforcement learning (Peng et al., 2018) and exploration (Goyal et al., 2019; Jaques et al., 2019; Goyal et al., 2020). Among them, Goyal et al. mention the multi-agent communication in their appendix, showing a method to minimize the communication by penalizing the effect of one agent's messages on another one's policy. However, it does not consider the limited bandwidth constraint.

3. Problem Setting

We consider a communicative multi-agent reinforcement learning task, which is extended from Dec-POMDP and described as a tuple $\langle n, \boldsymbol{S}, \boldsymbol{A}, r, P, \boldsymbol{O}, \boldsymbol{\Omega}, \boldsymbol{M}, \gamma \rangle$, where n

represents the number of agents. $\boldsymbol{S}$ represents the space of global states. $\boldsymbol{A} = \{A_i\}_{i=1,\cdots,n}$ denotes the space of actions of all agents. $\boldsymbol{O} = \{O_i\}_{i=1,\cdots,n}$ denotes the space of observations of all agents. $\boldsymbol{M}$ represents the space of messages. $P : \boldsymbol{S} \times \boldsymbol{A} \to \boldsymbol{S}$ denotes the state transition probability function. All agents share the same reward as a function of the states and agents' actions $r : \boldsymbol{S} \times \boldsymbol{A} \to \mathbb{R}$. Each agent i receives a private observation $o_i \in O_i$ according to the observation function $\boldsymbol{\Omega}(s, i) : \boldsymbol{S} \to O_i$. $\gamma \in [0, 1]$ denotes the discount factor. As shown in Fig. 1, each agent receives observation o_i and a scheduling message c_i, then outputs an action a_i and a message m_i. A scheduler f_i^{sch} is introduced to receive messages $[m_1, \cdots, m_n] \in \boldsymbol{M}$ from all agents and dispatch scheduling messages $c_i = f_i^{sch}(m_1, \cdots, m_n) \in M_i$ for each agent i.

We adopt a centralized training and decentralized execution paradigm (Lowe et al., 2017), and further relax it by allowing agents to communicate. That is, during training, agents are granted access to the states and actions of other agents for the centralized critic, while decentralized execution only requires individual states and scheduled messages from a well-trained scheduler.

Our end-to-end method is to learn a communication protocol $\pi_i^{pro}(m_i|o_i, c_i)$, an policy $\pi_i^a(a_i|o_i, c_i)$, and a scheduler $f_i^{sch}(c_i|m_1, \cdots, m_n)$, which jointly maximize the expected discounted return for each agent i:

$$
\begin{aligned}
J_i &= \mathbb{E}_{\pi_i^a, \pi_i^{pro}, f_i^{sch}}[\Sigma_{t=0}^{\infty} \gamma^t r_i^t(s, a)] \\
&= \mathbb{E}_{\pi_i^a, \pi_i^{pro}, f_i^{sch}}[Q_i(s, a_1, \cdots, a_n)] \\
&\approx \mathbb{E}_{\pi_i^a, \pi_i^{pro}, f_i^{sch}}[Q_i(o_1, \cdots, o_n, c_1, \cdots, c_n, a_1, \cdots, a_n)] \\
&= \mathbb{E}_{\pi_i^a, \pi_i^{pro}, f_i^{sch}}[Q_i(h_1, \cdots, h_n, a_1, \cdots, a_n)]
\end{aligned} \tag{1}
$$

where r_i^t is the reward received by the $i-$th agent at time t, Q_i is the centralized action-value function for each agent i, and $\mathbb{E}_{\pi_i^a, \pi_i^{pro}, f_i^{sch}}$ denotes an expectation over the trajectories $\langle s, a_i, m_i, c_i \rangle$ generated by $p^{\pi_i^a}, \pi_i^a(a_i|o_i, c_i), \pi_i^{pro}(m_i|o_i, c_i), f_i^{sch}(c_i|m_1, \cdots, m_n)$. Here we follow the simplification in (Lowe et al., 2017) to replace the global states with joint observations and use an abbreviation h_i to represent the joint value of $[o_i, c_i]$ in the rest of this paper.

The limited bandwidth B is a range of frequencies within a given band. It exists in the two processes of transmission: messages from agents to the scheduler and scheduling messages from the scheduler to agents as shown in Fig. 1. In the next section, we will discuss how the limited bandwidth B affects the communication.

4. Connection between Limited Bandwidth and Multi-agent Communication

In this section, from the perspective of communication theory, we show how the limited bandwidth B requires low-entropy messages throughout the transmission. We then discuss how to measure the message's entropy.

4.1. Communication Process

We show the communication process (Figure 2) from agents to the scheduler, which consists of five stages: analog-to-digital, coding, transmission, decoding and digital-to-analog (Freeman, 2004). When agent transmits a continuous message m_i of agent i, an analog-to-digital converter (ADC) maps it into a countable set. An ADC can be modeled as two processes: sampling and quantization. Sampling converts a time-varying signal into a sequence of *discrete-time* real-value signal. This operation is corresponding to the discrete timestep in RL. Quantization replaces each real number with an approximation from a finite set of *discrete values*. In the coding phase, the quantized messages m_i^{Δ} is mapped to a bitstream using source coding methods such as Huffman coding. In the transmission phase, the transmitter modulates the bitstream into wave, and transmit the wave through a channel, then the receiver demodulates the wave into another bitstream due to some distortion in the channel. Then, decoding is the inverse operation of coding, mapping the bitstream to the recovered messages $\hat{m}_i^{\Delta}$. Finally, the scheduler receives a reconstructed analog message from a digital-to-analog converter (DAC). The same process happens when sending the scheduled messages c_i from the scheduler to the agent i. The bandwidth actually restricts the transmission phase.

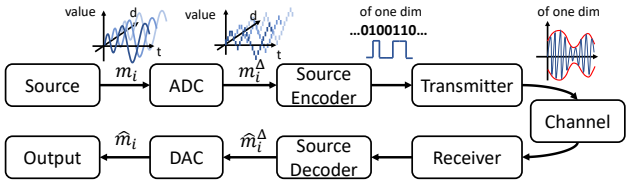

Figure 2. Overview of the Communication Process. Axes of messages are time, dimension of the message vector, and value of each element in the vector.

4.2. Limited Bandwidth Restricts Message's Entropy

We model the messages m_i as continuous random vectors M_i, i.e., continuous vectors sampled from a certain distribution. The reason is that a message is sent by one agent in each timestep, while over a long duration, the messages follow some distributions. We abuse $\boldsymbol{m}$ to represent the random vector by omitting the subscript, and explain the

subscript in special cases.

For simplicity, we consider sending an element X of the continuous random vector $\boldsymbol{m}$, which is a continuous random variable, and then extend our conclusion to $\boldsymbol{m}$. First, we quantize the continuous variable into discrete symbols. The quantization brings a gap between the entropy of the discrete variables and differential entropy of the continuous variables.

Remark 1 (Relationship between entropy and differential entropy)**.** *Consider a continuous random variable X with a probability density function $f_X(x)$. This function is then quantized by dividing its range into K levels of interval Δ, where $K = ceil(|X|/\Delta)$, and $|X|$ is max amplitude of variable. The quantized variable is X^Δ. Then the difference between differential entropy $H(X)$ and entropy $H\left(X^\Delta\right)$ is $H(X) - H\left(X^\Delta\right) = \log_2(\Delta)$.*

Note that for a fixed small identical interval Δ, there is only a constant difference between differential entropy and entropy. Then we encode these quantized symbols.

Remark 2 (Source Coding Theorem (Shannon, 1948))**.** *In the source coding phase, a set of n quantized symbols is to be encoded into bitstreams. These symbols can be treated as n independent samples of a discrete random variable X^Δ with entropy $H(X^\Delta)$. Let L be the average number of bits to encode the n symbols. The minimum L satisfies $H(X^\Delta) \leq L < H(X^\Delta) + \frac{1}{n}$.*

Remark 2 regularizes the coding phase in the communication process. Then in the transmission, over a noiseless channel, the maximum bandwidth is defined as the maximum data rate:

Remark 3 (The Maximum Data Rate (Freeman, 2004))**.** *The maximum data rate R_{max} (bits per second) over a noiseless channel satisfies: $R_{max} = 2B\log_2 K$, where B is the bandwidth (Hz) and K is the number of signal levels.*

Remark 3 is derived from the Nyquist criterion (Freeman, 2004) and specifies how the bandwidth of a communication system affects the transmission data rate for reliable transmission in the noiseless condition. Based on these three remarks, we show how the limited bandwidth constraint affects the multi-agent communication.

Proposition 1. *In a noiseless channel, the bandwidth of channel B limits the entropy of the messages' elements.*

Proof. Given a message's element X as an i.i.d continuous random variable with differential entropy $H(X)$, its quantized time series $X_1^\Delta, \cdots, X_t^\Delta, \cdots$ (here the subscript means different times) with entropy $H(X^\Delta) = H(X) - \log_2 \Delta$, the communication system's bandwidth B, as well as the signal levels K, the communication system transmits n symbols per second. We define R_{code} as an unbiased estimation of L in Remark 2. So the transmission rate $R_{trans}(\frac{\text{bit}}{\text{second}}) = n \cdot R_{code}(\frac{\text{bit}}{\text{symbol}}) \geq n \cdot H(X^\Delta) \geq n \cdot (H(X) - \log \Delta)$.[2] According to Remark 3, $R_{trans} \leq R_{max} = 2B\log_2 K$. Consequently, we have $H(X) \leq \frac{2B\log_2 K}{n} + \log_2 \Delta$. □

Note that although a frequent symbol among these sending symbols uses less bits than $H(X^\Delta)$ and vice versa, when we send a bunch of symbols, R_{code} is lager than $H(X^\Delta)$ *on average*.

Proposition 2. *In a noiseless channel, the bandwidth of channel B limits the entropy of the messages.*

Proof. When sending the random vector, i.e., the message $M_i = [X_1, X_2, \cdots, X_d]$, where the subscript means different entries of the vector and d is the dimension, each variable X_i occupies a bandwidth B_i, which satisfies $\sum_{i=1}^d B_i = B$. Assume all entries are quantilized with the same interval, according to (Cover & Thomas, 2012), the upper bound of the messages $H(M_i) = H(X_1, \cdots, X_d) \leq \sum_{i=1}^d H(X_i) \leq \frac{2dB\log_2 K}{n} + d\log_2 \Delta$. □

Eventually, a limited bandwidth B enforces an upper bound H_c to the message's entropy $H(M_i) \leq H_c \propto B$.

4.3. Measurement of a Message's Entropy

The messages M_i as an i.i.d random vector can follow any distribution, so it is hard to determine the message's entropy. So, we keep a historical record of the message and find a quantity to measure the message's entropy.

Proposition 3. *When we have a historical record of the messages to estimate the messages' mean $\boldsymbol{\mu}$ and co-variance $\boldsymbol{\Sigma}$, the entropy of the messages is upper bounded by $\frac{1}{2}\log((2\pi e)^d|\boldsymbol{\Sigma}|)$, where d is the dimension of M_i.*

Proof. The message M_i follows a certain distribution, and we are only certain about its mean and variance. According to the principle of maximum entropy (Jaynes, 1957), the Gaussian distribution has maximum entropy relative to all probability distributions covering the entire real line $(-\infty, \infty)$ but having a finite mean and finite variance (see the proof in (Cover & Thomas, 2012)). So $H(M_i) \leq \frac{1}{2}\log((2\pi e)^d\boldsymbol{\Sigma})$, where the right term is the entropy of multivariate Gaussian $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. □

We conclude that $\frac{1}{2}\log((2\pi e)^d|\boldsymbol{\Sigma}|)$ offers an upper bound to approximate $H(M_i)$, and this upper bound should be less than or equal to the limited bandwidth constraint to ensure that the message with any possible distribution satisfies the limited bandwidth constraint.

[2] $\frac{\text{bit}}{\text{second}}$ and $\frac{\text{bit}}{\text{symbol}}$ are units of measure.

5. Informative Multi-agent Communication

As shown in the previous section, the limited bandwidth requires agents to send low-entropy messages. In this section, we first introduce our method for learning a valuable and low-entropy communication protocol via the information bottleneck principle. Then, we discuss how to use the same principle in scheduling.

5.1. Variational Information Bottleneck for Learning Protocols

We propose an informative multi-agent communication via information bottleneck principle to learn protocols. Concretely, we propose an information-theoretic regularization on the mutual information $I(H_i; M_i)$ between the messages and the input features

$$I(H_i; M_i) \leq I_c \tag{2}$$

where M_i is a random vector with a probability density function $p_{M_i}(m_i)$, which represents the possible assignments of the messages m_i, and H_i is a random vector with a probability density function $p_{H_i}(h_i)$ which the possible values of $[o_i, c_i]$. We omit the subscripts in the density functions in the rest of the paper. Eventually, with the help of variational information bottleneck (Alemi et al., 2016), this regularization enforces agents to send low-entropy messages.

Consider a scenario with n agents' policies $\{\pi_i^a\}_{i=1,\cdots,n}$ and protocols $\{\pi_i^{pro}\}_{i=1,\cdots,n}$ which are parameterized by $\{\theta_i\}_{i=1,\cdots,n} = \{\theta_i^a, \theta_i^{pro}\}_{i=1,\cdots,n}$, and with schedulers $\{f_i^{sch}\}_{i=1,\cdots,n}$ which are parameterized by $\{\phi_i\}_{i=1,\cdots,n}$. Consequently, for learning the communication protocol with fixed schedulers, the agent i is supposed to maximize:

$$J(\theta_i) = \mathbb{E}_{\pi_i^a, \pi_i^{pro}, f_i^{sch}}[Q_i(h_1, \cdots, h_n, a_1, \cdots, a_n)]$$
$$\text{s.t. } I(H_i; M_i) \leq I_c$$

Practically, we propose to maximize the following objective using the information bottleneck Lagrangian:

$$J'(\theta_i) = \mathbb{E}_{\pi_i^a, \pi_i^{pro}, f_i^{sch}}[Q_i(h_1, \cdots, h_n, a_1, \cdots, a_n)] - \beta I(H_i; M_i) \tag{3}$$

where the β is the Lagrange multiplier. The mutual information is defined according to:

$$I(H_i; M_i) = \iint p(h_i, m_i) \log \frac{p(h_i, m_i)}{p(h_i)p(m_i)} \mathrm{d}h_i \mathrm{d}m_i$$
$$= \iint p(h_i)\pi^{pro}(m_i|h_i) \log \frac{\pi^{pro}(m_i|h_i)}{p(m_i)} \mathrm{d}h_i \mathrm{d}m_i$$

where $p(h_i, m_i)$ is the joint probability of h_i and m_i.

However, computing the marginal distribution $p(m_i) = \int \pi^{pro}(m_i|h_i)p(h_i)dh_i$ can be challenging since we do not know the prior distribution of $p(h_i)$. With the help of variational information bottleneck (Alemi et al., 2016), we use a Gaussian approximation $z(m_i)$ of the marginal distribution $p(m_i)$ and view π^{pro} as multivariate variational encoders. Since $D_{KL}[p(m_i)||z(m_i)] \geq 0$, where the D_{KL} is the Kullback-Leibler divergence, we expand the KL term and get $\int p(m_i) \log p(m_i) dm_i \geq \int p(m_i) \log z(m_i) dm_i$, an upper bound on the mutual information $I(H_i; M_i)$ can be obtained via the KL divergence:

$$I(H_i; M_i) \leq \int p(h_i)\pi_i^{pro}(m_i|h_i) \log \frac{\pi_i^{pro}(m_i|h_i)}{z(m_i)} dh_i dm_i$$
$$= \mathbb{E}_{h_i \sim p(h_i)}[D_{KL}[\pi^{pro}(m_i|h_i)\|z(m_i)]] \tag{4}$$

This provides a lower bound $\tilde{J}(\theta)$ on the regularized objective that we maximize:

$$J'(\theta_i) \geq \tilde{J}(\theta_i) = \mathbb{E}_{\pi_i^a, \pi_i^{pro}, f_i^{sch}}[Q_i(h_1, \cdots, h_n, a_1, \cdots, a_n)] - \beta \mathbb{E}_{h_i \sim p(h_i)}[D_{KL}[\pi_i^{pro}(m_i|h_i)\|z(m_i)]]$$

Consequently, the objective's derivative is:

$$\nabla_{\theta_i} \tilde{J}(\pi_i) = \mathbb{E}_{\pi_i^a, \pi_i^{pro}, f_i^{sch}}\Big[\nabla_{\theta_i} \log(\pi_i(a_t|s_t)) Q_i(h_1, \cdots, h_n, a_1, \cdots, a_n) - \beta \nabla_{\theta_i} D_{KL}[\pi^{pro}(m_i|h_i)\|z(m_i)]\Big] \tag{5}$$

With the variational information bottleneck, we can control the messages' distribution and thus control their entropy with different prior $z(m_i)$ to satisfy different limited bandwidths in the training stage. That is, with the regulation of $D_{KL}[p(m_i|h_i)\|z(m_i)]$, the messages' probability density function $p(m_i) = \sum_{h_i} p(m_i|h_i)p(h_i) \approx \sum_{h_i} z(m_i)p(h_i) = z(m_i)\sum_{h_i} p(h_i) = z(m_i)$. Thus $H(M_i) = -\int p \log p \mathrm{d}m_i \approx -\int z \log z \mathrm{d}m_i$.

5.2. Unification of Learning Protocols and Scheduling

The scheduler for agent i is $f_i^{sch}(c_i|m_1, \cdots, m_n)$. The terms "scheduler are from SchedNet (Kim et al., 2019), which introduces "communication scheduling" and "scheduler" to represent the filtering process instead of timing. Recall the communication protocols for agent i: $\pi_i^{pro}(m_i|h_i)$. Due to the same form of the protocol and the scheduling, the scheduler is supposed to follow the same principle for learning a weight-based mechanism. Variational information bottleneck can be applied in scheduling for agent i with regularization on the mutual information between the scheduling messages c_i and all agents' messages $I(C_i; M_1, \cdots, M_n)$, where C_i is a random vector with a probability density function $p_{C_i}(c_i)$, which represent different values of c_i. We follow the joint training scheme for training the communication protocol and scheduling (Foerster et al., 2016), which allows the gradients flow across agents from the recipient agent to the scheduler to the sender agent.

Formally, the schedulers $\{f_i^{sch}\}_{i=1,\cdots,n}$ are parameterized by $\{\phi_i\}_{i=1,\cdots,n}$ as defined in section 5.1. We would opti-

mize the lower bound in terms of $\{\phi_i\}_{i=1,\cdots,n}$:

$$J'(\phi_i) \geq \tilde{J}(\phi_i) = \mathbb{E}_{\pi_i^a,\pi_i^{pro},f_i^{sch}}[Q_i(h_1,\cdots,h_n,a_1,\cdots,a_n)] \\ -\beta\mathbb{E}_{p(m_1,m_2,\cdots,m_n)}[D_{KL}[f_i^{sch}(c_i|m_1,\cdots,m_n)\|z(c_i)]]$$

Consequently, the objective's derivative is:

$$\nabla_{\theta_i}\tilde{J}(\phi_i) = \mathbb{E}_{\pi_i^a,\pi_i^{pro},f_i^{sch}}\Big[\nabla_{\theta_i}\log(\pi_i(a_t|s_t)) \\ Q_i(h_1,\cdots,h_n,a_1,\cdots,a_n) \\ -\beta\nabla_{\phi_i}D_{KL}[f_i^{sch}(c_i|m_1,\cdots,m_n)\|z(c_i)]\Big] \quad (6)$$

5.3. Implementation of the limited bandwidth Constraint

During the execution stage, the messages may still obey the low-entropy requirement. We implement the limited bandwidth during the execution according to the low-entropy principle. Also, due to the variety of the real-life communication source coding methods, like Huffman coding, and communication protocols, like TCP/UDP, bitstream can carry different amounts of information in different situations. As a result, we utilize the entropy as a general measurement and clip the messages' variance to simulate the limited-bandwidth constraint. Concretely, we use a batch-normalization-like layer which records the messages' mean and variance during training as Prop. 2 requires. And it clips the messages by reducing their variance during execution. The purpose of our normalization layer is to measure the messages' mean and variance, as well as to simulate the external limited bandwidth constraint during execution. It is customized and different from standard batch normalization (Ioffe & Szegedy, 2015). For example, the maximum bandwidth of a 4-ary communication system is 100 bit/s, if we want achieve reliable transmission with transmitting 10^3 messages per second. Then we can determine the equivalent variance $\sigma^2 \approx 3.2$ according to $\frac{1}{2}\log(2\pi e\sigma^2) = \frac{2B\log_2 K}{n}$. In training stage, we record the agent's messages' variance which is 5. In inference stage, the bandwidth requires the messages' entropy not to excess 3.2. Then, we decrease the variance from 5 to 3.2 by using the specific normalization layer.

Algorithm 1: Informative Multi-agent Communication

Initialize the network parameters θ_a, θ_{pro}, θ_Q, and ϕ_{sch}
Initialize the target network parameters θ'_a, and θ'_Q
for *episode* $\leftarrow 1$ ***to*** *num_episodes* **do**
  Reset the environment **for** $t \leftarrow 1$ ***to*** *num_step* **do**
    Get features $h_i = [o_i, c_i]$ for each agent i
    Each agent i gets messages from channel $m_i = \pi^i_{pro}(h_i)$
    Get scheduled message $c_i = f_{sch}(m_1,\cdots,m_n)$
    Each agent i selects action based on features and messages $a_i = \pi^i_a(h_i, c_i)$
    Execute actions $a = (a_1,\cdots,a_n)$, and observe reward r new observation o_i for each agent i
    Store $(\boldsymbol{o_t}, a, r, \boldsymbol{o_{t+1}}, \boldsymbol{m}, \boldsymbol{c})$ in replay buffer D
    if *episode%update threshold* $== 0$ **then**
      Sample a random mini-batch of S samples $(\boldsymbol{o}, a, r, \boldsymbol{o'}, \boldsymbol{m}, \boldsymbol{c})$ from D
      Obtain the features $h'_i = [o'_i, c_i]$ and the messages m_i for each agent i
      Set $y^j = r_i^j + \gamma Q_i^{\boldsymbol{\pi'}}(\boldsymbol{o}, a_1', \cdots, a_n')|_{a_{k'}=\pi_a^i{}'(h'_i,c_i)}$
      Update Critic by minimizing the loss $L(\theta) = \frac{1}{S}\sum_j(Q(\boldsymbol{o}, a_1,\cdots,a_n) - \hat{y})^2$
      Update policy, protocol and scheduler using the sampled policy gradients $\nabla_{\theta_i}\tilde{J}(\pi_i)$ for each agent i
      Update all target networks' parameters for each agent i: $\theta_i' = \tau\theta_i + (1-\tau)\theta_i'$

6. Experiment

Environment Setup. We evaluate IMAC on a variety of tasks and environments: cooperative navigation and predator prey in Multi Particle Environment (Lowe et al., 2017), as well as 3m and 8m in StarCraftII (Samvelyan et al., 2019). The detailed experimental environments are elaborated in the following subsections as well as in supplementary material.

Baselines. We choose the following methods as baselines: (1) TarMAC (Das et al., 2019): A state-of-the-art multi-agent communication method for limited bandwidth, which uses a self-attention weight-based scheduling mechanism for scheduling and learns the communication protocol in an end-to-end manner. (2) GACML (Mao et al., 2019): A multi-agent communication method for limited bandwidth, which uses a gating mechanism for downsizing communication agents and learns the communication protocol in an end-to-end manner. (3) SchedNet (Kim et al., 2019): A multi-agent communication method for limited bandwidth, which uses a selecting mechanism for downsizing communication agents and learns the communication protocol where the message is one real value (float16 type). Also, we modify MADDPG (Lowe et al., 2017) and QMIX (Rashid et al., 2018) with communication as baselines to show that IMAC can facilitate different multi-agent methods and work well with limited bandwidth constraints.

6.1. Cooperative Navigation

In this scenario, n agents cooperatively reach k landmarks while avoiding collisions. Agents observe the relative positions of other agents and landmarks and are rewarded with a shared credit based on the sum of distances between agents to their nearest landmark, while they are penalized when colliding with other agents. Agents learn to infer and occupy the landmarks without colliding with other agents based on their own observation and received information from other

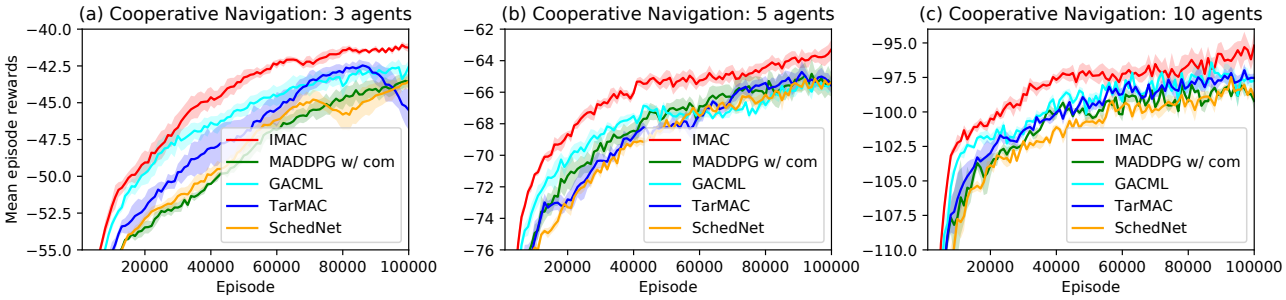

Figure 3. Learning curves comparing IMAC to other methods for cooperative navigation. As the number of agents increases (from left to right), IMAC improves agents' performance and converge faster.

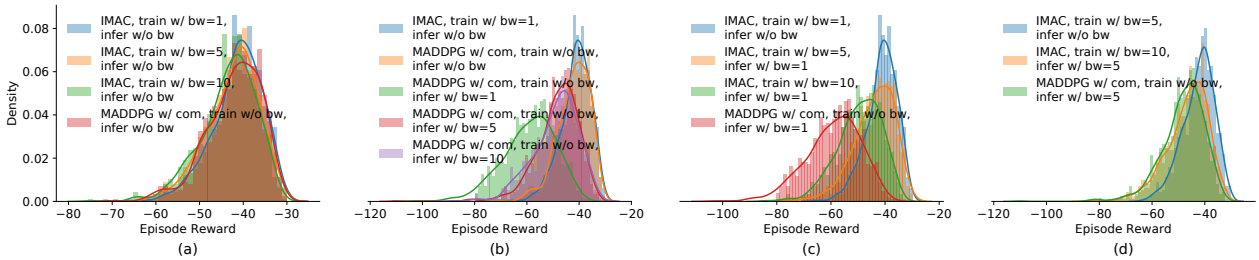

Figure 4. Density plot of episode reward per agent during the execution stage. (a) Reward distribution of IMAC trained with different prior distributions against MADDPG with communication. (b) Reward distribution of MADDPG with communication under different limited bandwidth environment. (c), (d) Reward distribution of IMAC trained with different prior distributions against MADDPG with communication under the same bandwidth constraint. "bw=δ" means in the implementation of the limited bandwidth constraint, the variance $\mathbf{\Sigma}$ of Gaussian distribution is δ.

agents.

Comparison with baselines. We compare IMAC with TarMAC, GACML, and SchedNet because they represent the method of learning the communication protocols via end-to-end training with the specific scheduler and clipping the messages respectively. Also due to different bandwidth definitions, we also compare with the modified MADDPG with communication, which is trained without the limited bandwidth constraint, because it offers the baseline performance of the centralized training and decentralized execution.

Figure 3 (a) shows the learning curve of 100,000 episodes in terms of the mean episode reward over a sliding window of 1000 episodes. We can see that at the end of the training, agents trained with communication have higher mean episode reward. According to (Lowe et al., 2019), "increase in reward when adding a communication channel" is sufficient to effective communication. Additionally, IMAC outperforms other baselines along the process of training, i.e., IMAC can reach upper bound of performance early. By using the information bottleneck method, messages are less redundant, thus agents converge fast (More analysis can be seen in the supplementary materials).

We also investigate the performance when the number of agents increases. We make a slight modification on environment about agents' observation. According to (Jiang & Lu, 2018), we constrain that each agent can observe the nearest three agents and landmarks with relative positions and velocity. Figure 3 (b) and (c) show that the leading performance of IMAC in the 5 and 10-agent scenarios.

Performance under stronger limited bandwidth. We first train IMAC with different priors to satisfy different bandwidths. Then we evaluate IMAC and the modified MADDPG with communication by checking agents' performance under different limited bandwidth constraints during the execution stage. Figure 4 shows the density plot of episode reward per agent during the execution stage. We first respectively train IMAC with different prior distributions $z(M_i)$ of $N(0,1)$, $N(0,5)$, and $N(0,10)$, to satisfy different default limited bandwidth constraints. Consequently, the entropy of agents' messages satisfies the bandwidth constraints. In the execution stage, we constrain these algorithms into different bandwidths. As depicted in Figure 4

Predator \ Prey	IMAC	TarMAC	GACML	SchedNet	MADDPG w/ com
IMAC	**32.32****-4.26**	**28.91**\ -22.27	**28.25** \ -26.11	22.67 \ -36.53	**34.33** \ -22.62
TarMAC	25.13 \ **-2.94**	23.45 \ -20.42	22.12 \ -16.51	**32.52** \ -42.39	27.54 \ -29.36
GACML	21.52 **-12.74**	11.49 \ -24.93	13.93 \ -12.95	25.49 \ -27.42	28.47 \ -27.75
SchedNet	24.74 **-9.63**	7.84 \ -23.56	12.48 \ -23.67	5.98 \ -26.82	21.53 \ -26.43
MADDPG w/ com	28.63 \ -15.60	19.32 \ -21.52	26.91 \ -19.76	22.17 \ -35.37	16.87 \ **-13.09**

Table 1. Cross-comparison between IMAC and baselines on predator-prey.

(a), IMAC with different prior distributions can reach the same outcome as MADDPG with communication. Figure 4 (b) shows that MADDPG with communication fails in the limited bandwidth environment. From Figure 4 (c) and (d), we can see that the same bandwidth constraint is less effective in IMAC than MADDPG with communication. Results here demonstrate that IMAC discards useless information without impairment on performance.

Ablation. We investigate the effect of limited bandwidth and β on multi-agent communication on the performance of agents. Figure 5 (a) shows the learning curve of IMAC with different prior distributions. IMAC with $z(M_i) = N(0, 1)$ achieves the best performance. When the variance is smaller or larger, the performance suffers some degradation. It is reasonable because a smaller variance means a more lossy compression, leading to less information sharing. A larger variance must bring about more redundant information than the variance without regulation, thus leading to slow convergence. β controls the degree of compression between h_i and m_i for each agent i: the larger β, the more lossy compression. Figure 5 (b) shows a similar result to the ablation on limited bandwidth constraint. The reason is the same: a larger β means a more strict compression while a smaller β means a less strict one.

The ablation shows that as a compression algorithm, the information bottleneck method extracts the most informative elements from the source. A proper compression rate is good for multi-agent communication, because it cannot only avoid losing much information caused by higher compression, but also resist much noise caused by lower compression.

6.2. Predator Prey

In this scenario, m slower predators chase n faster preys around an environment with l landmarks impeding the way. As same as cooperative navigation, each agent observes the relative position of other agents and landmarks. Predators share common rewards, which are assigned based on the collision between predators and preys, as well as the minimal distance between two groups. Preys are penalized for running out of the boundary of the screen. In this way, predators would learn to approach and surround preys, while

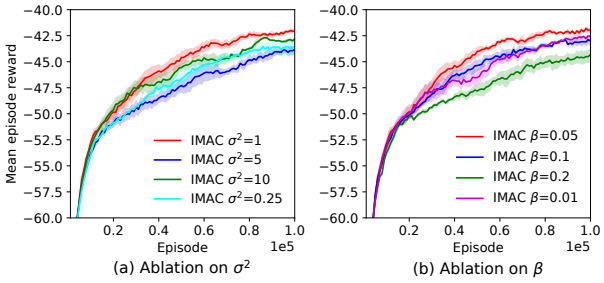

Figure 5. Ablation: learning curves with respect to Σ and β

preys would learn to feint to save their teammates.

We set the number of predators as 4, the number of preys as 2, and the number of landmarks as 2. We use the same architecture in cooperative navigation. Agents only communicate with their teammates. We train our agents by self-play for 100,000 episodes and then evaluate performance by cross-comparing between IMAC and the baselines. We average the episode rewards across 1000 rounds (episodes) as scores.

Comparison with baselines. We use the same baselines as in the cooperative navigation. Table 1 represents the cross-comparing between IMAC and the baselines. Each cell consists of two numbers which denote the mean episode rewards of the predators and preys respectively. The larger the score is, the better the algorithm is. We first focus on the mean episode rewards of predator row by row. Facing the same prey, IMAC has higher scores than the predators of all the baselines and hence are stronger than other predators. Then, the mean episode rewards of the prey column by column show the ability of the prey to escape. We can see that IMAC has higher scores than the preys of most baselines and hence are stronger than other preys. We argue that IMAC leads to better cooperation than the baselines even in competitive environments and the learned policy of IMAC predators and preys can generalize to the opponents with different policies.

Performance under stronger limited bandwidth. Similar to the cooperative navigation, we evaluate algorithms by showing the performance under different limited bandwidth

Predator \Prey	MADDPG_e1	MADDPG_e5	IMAC	IMAC_t5_e1	IMAC_t10_e1	IMAC_t10_e5
MADDPG_e1	18.01 \-**14.22**	24.15 \-29.88	22.38 \-16.91	47.59 \-45.64	34.25 \-27.68	50.81 \-43.62
MADDPG_e5	26.32 \-20.48	15.67 \-**11.59**	29.06 \-22.16	27.07 \-22.89	23.44 \-20.41	32.24 \-26.46
IMAC	**51.24** \-42.56	**37.37** \-45.521	**44.64** \-36.49	**49.12** \-42.65	**36.63** \-30.03	35.42 \-**28.82**
IMAC_t5_e1	38.86 \-32.06	34.54 \-35.03	9.97 \-**3.11**	26.25 \-21.06	11.80 \-7.558	**38.32** \-32.28
IMAC_t10_e1	26.67 \-21.418	34.99 \-35.02	9.71 \-**4.11**	9.82 \-6.92	9.82 \-6.92	37.50 \-31.30
IMAC_t10_e5	45.88 \-38.27	26.39 \-35.42	11.51 \-**9.12**	30.02 \-27.41	29.08 \-25.661	22.25 \-16.51

Table 2. Cross-comparison in different bandwidths on predator-prey. "t5" means that IMAC is trained with the variance $|\Sigma| = 5$. "e1" means that during the execution, we use the batch-norm like layer to clip the message to enforce its variance $|\Sigma| = 5$.

constraints during execution. Table 2 shows the performance under different limited bandwidth constraints during inference in the environment of predator and prey. We can see with limited bandwidth constraint, MADDPG with communication and IMAC suffer a degradation of performance. However, IMAC outperforms MADDPG with communication with respect to resistance to the effect of limited bandwidth.

6.3. StarCraftII

We apply our method and baselines to decentralized StarCraft II micromanagement benchmark to show that IMAC can facilitate different multi-agent methods. We use the setup introduced by SMAC (Samvelyan et al., 2019) and consider combat scenarios.

3m and 8m. Both tasks are symmetric battle scenarios, where marines controlled by the learned agents try to beat enemy units controlled by the built-in game AI. Agents will receive some positive (negative) rewards after having enemy (allied) units killed and/or a positive (negative) bonus for winning (losing) the battle.

Comparison with Baselines. We adapt QMIX with communication and with IMAC, because QMIX uses the centralized training decentralized execution scheme for discrete actions. We also evaluate MADDPG with communication. However, SMAC is a discrete-action scenario, while MADDPG is for continuous control. Even if we modify the MADDPG into discrete action setup, it still fails to get any positive reward. Fig. 6 shows the learning curve of 200 episodes in terms of the mean episode rewards. We can see that at the beginning, QMIX with IMAC has a similar or even poor performance than QMIX with unlimited communication. As the training process going, QMIX with IMAC has a better performance than QMIX with unlimited communication. The result shows that IMAC can facilitate different multi-agent methods which have different centralized training schemes.

Performance under stronger limited bandwidth. We evaluate agents' performance under different limited bandwidth constraints. Results show a similar conclusion as in previous tasks (Details can be seen in the supplementary materials).

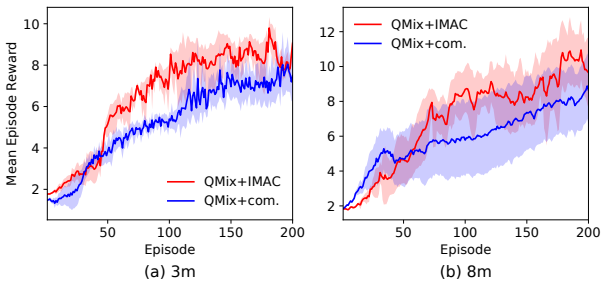

Figure 6. Learning curves comparing IMAC to other methods for 3m and 8m in Starcraft II.

7. Conclusion

In this paper, we have proposed an *informative multi-agent communication* method in the limited bandwidth environment, where agents utilize the information bottleneck principle to learn an informative protocol as well as scheduling. We prove that limited bandwidth constrains the entropy of the messages. We introduce a customized batch-norm layer, which controls the messages'entropy to simulate the limited bandwidth constraint. Inspired by the information bottleneck method, our proposed IMAC algorithm learns informative protocols and a weight-based scheduler, which convey low-entropy and useful messages. Empirical results and an accompanying ablation study show that IMAC significantly improves the agents' performance under limited bandwidth constraint and leads to faster convergence.

Acknowledgements

This research is supported by the National Research Foundation, Singapore under National Satellite of Excellence in Trustworthy Software Systems (Award No: NSOE-TSS2019-01), AI Singapore Programme (AISG Award No: AISG-RP-2019-0013), Singapore MoE AcRFTier-1 RG24/18 (S), and NTU. We gratefully acknowledge the support of NVAITC (NVIDIA AI Tech Center) for our research.

References

Alemi, A. A., Fischer, I., Dillon, J. V., and Murphy, K. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

Cover, T. M. and Thomas, J. A. *Elements of Information Theory*. John Wiley & Sons, 2012.

Das, A., Gervet, T., Romoff, J., Batra, D., Parikh, D., Rabbat, M., and Pineau, J. TarMAC: Targeted multi-agent communication. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 1538–1546, 2019.

Foerster, J., Assael, I. A., de Freitas, N., and Whiteson, S. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 2137–2145, 2016.

Freeman, R. *Telecommunication System Engineering*, pp. 398–399. Wiley Series in Telecommunications and Signal Processing. Wiley, 2004. ISBN 9780471451334.

Goyal, A., Islam, R., Strouse, D., Ahmed, Z., Botvinick, M., Larochelle, H., Levine, S., and Bengio, Y. Infobot: Transfer and exploration via the information bottleneck. *arXiv preprint arXiv:1901.10902*, 2019.

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

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Jaques, N., Lazaridou, A., Hughes, E., Gulcehre, C., Ortega, P., Strouse, D., Leibo, J. Z., and De Freitas, N. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In *International Conference on Machine Learning*, pp. 3040–3049, 2019.

Jaynes, E. T. Information theory and statistical mechanics. *Physical Review*, 106(4):620, 1957.

Jiang, J. and Lu, Z. Learning attentional communication for multi-agent cooperation. In *Advances in Neural Information Processing Systems*, pp. 7265–7275, 2018.

Kilinc, O. and Montana, G. Multi-agent deep reinforcement learning with extremely noisy observations. *arXiv preprint arXiv:1812.00922*, 2018.

Kim, D., Moon, S., Hostallero, D., Kang, W. J., Lee, T., Son, K., and Yi, Y. Learning to schedule communication in multi-agent reinforcement learning. *arXiv preprint arXiv:1902.01554*, 2019.

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

Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, O. P., and Mordatch, I. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*, pp. 6379–6390, 2017.

Lowe, R., Foerster, J., Boureau, Y.-L., Pineau, J., and Dauphin, Y. On the pitfalls of measuring emergent communication. *arXiv preprint arXiv:1903.05168*, 2019.

Mao, H., Gong, Z., Zhang, Z., Xiao, Z., and Ni, Y. Learning multi-agent communication under limited-bandwidth restriction for internet packet routing. *arXiv preprint arXiv:1903.05561*, 2019.

Mordatch, I. and Abbeel, P. Emergence of grounded compositional language in multi-agent populations. In *AAAI Conference on Artificial Intelligence*, 2018.

OpenAI. OpenAI Five. https://openai.com/blog/openai-five/, 2019. Accessed March 4, 2019.

Peng, P., Yuan, Q., Wen, Y., Yang, Y., Tang, Z., Long, H., and Wang, J. Multiagent bidirectionally-coordinated nets for learning to play starcraft combat games. *arXiv preprint arXiv:1703.10069*, 2, 2017.

Peng, X. B., Kanazawa, A., Toyer, S., Abbeel, P., and Levine, S. Variational discriminator bottleneck: Improving imitation learning, inverse RL, and gans by constraining information flow. *arXiv preprint arXiv:1810.00821*, 2018.

Rashid, T., Samvelyan, M., De Witt, C. S., Farquhar, G., Foerster, J., and Whiteson, S. Qmix: monotonic value function factorisation for deep multi-agent reinforcement learning. *arXiv preprint arXiv:1803.11485*, 2018.

RoboCup. Robocup Federation Official Website. https://www.robocup.org/, 2019. Accessed April 10, 2019.

Samvelyan, M., Rashid, T., de Witt, C. S., Farquhar, G., Nardelli, N., Rudner, T. G. J., Hung, C.-M., Torr, P. H. S., Foerster, J., and Whiteson, S. The StarCraft Multi-Agent Challenge. *CoRR*, abs/1902.04043, 2019.

Shannon, C. E. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

Singh, A., Jain, T., and Sukhbaatar, S. Learning when to communicate at scale in multiagent cooperative and competitive tasks. *arXiv preprint arXiv:1812.09755*, 2018.

Sukhbaatar, S., Fergus, R., et al. Learning multiagent communication with backpropagation. In *Advances in Neural Information Processing Systems*, pp. 2244–2252, 2016.

Tishby, N., Pereira, F. C., and Bialek, W. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

Zhang, C. and Lesser, V. Coordinating multi-agent reinforcement learning with limited communication. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems*, pp. 1101–1108, 2013.