2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Sparse Discrete Communication Learning for Multi-Agent Cooperation Through Backpropagation

Benjamin Freed[1], Rohan James[1],Guillaume Sartoretti[2] and Howie Choset[1]

***Abstract*— Recent approaches to multi-agent reinforcement learning (MARL) with inter-agent communication have often overlooked important considerations of real-world communication networks, such as limits on bandwidth. In this paper, we propose an approach to learning sparse discrete communication through backpropagation in the context of MARL, in which agents are incentivized to communicate as little as possible while still achieving high reward. Building on top of our prior work on differentiable discrete communication learning, we develop a regularization-inspired message-length penalty term, that encourages agents to send shorter messages and avoid unnecessary communications. To this end, we introduce a variable-length message code that provides agents with a general means of modulating message length while keeping the overall learning objective differentiable. We present simulation results on a partially-observable robot navigation task, where we first show how our approach allows learning of sparse communication behavior while still solving the task. We finally demonstrate our approach can even learn an effective sparse communication behavior from demonstrations of an expert (potentially communication-free) policy.**

I. Introduction

Multi-agent reinforcement learning (MARL) has recently been used to learn effective control policies for complex multi-agent tasks, such as DOTA2 and StarCraft [1], [2], [3]. Typically, control policies learned using MARL are decentralized, meaning each agent makes decisions independently from one another, based solely on its own local observations. This property allows the computational complexity of multi-agent policies to scale linearly with number of agents, rather than exponentially, as would be the case for a centralized planner. Additionally, action selection can be fully parallelized. The primary drawback to MARL with such fully independent policies is that, because agents make decisions based solely on their local observations, agents cannot take advantage of all the information available to the group [4], [5], [6], [7]. This leads to sub-optimal policies and more limited coordination. One way to mitigate these problems is by allowing agents to selectively exchange information via inter-agent communication. This information exchange enables agents to make more informed decisions, while maintaining the scalability and parallelizability advantages of traditional MARL.

*This work was not supported by any organization

[1]Benjamin Freed, Rohan James, and Howie Choset are with the Robotics Institute at Carnegie Mellon University, Pittsburgh, PA 15213, USA. {bfreed,rohanjam,choset}@andrew.cmu.edu

[2]Guillaume Sartoretti is with the department of Mechanical Engineering at the National University of Singapore, 117575 Singapore. guillaume.sartoretti@nus.edu.sg

Several recent works have focused on the problem of MARL with inter-agent communication [8], [9]. However, these works have largely assumed an idealized communication channels that do not reflect difficulties associated with real-world communication networks. For example, recent works have either assumed agents can exchange real-valued messages [10], [11], containing a potentially unbounded quantity of information, or that agents can send a fixed-size discrete message to all agents within reach of communication, regardless of how much communication is actually necessary at a particular timestep [12], [6], [13]. In reality, communication networks typically support discrete (*e.g.*, digital) communication, and have limits on rates of information transfer. Such a communication network may become overburdened if all agents communicate maximally with each other at all times. For this reason, we seek a new approach in which agents can learn discrete communications while communicating sparsely, that is, sending the least amount of information necessary to achieve satisfactory behavior.

A simple approach to learning sparse communication would be to use a standard RL communication-learning framework, such as [12], [6], [14], and penalize agents with a negative reward for every bit that they send. However, this form of communication learning has been shown to be far less efficient than those based on gradient backpropagation [12], [15], and in theory scales poorly to large message spaces. In contrast, our approach to sparse communication learning through backpropagation builds upon our past approach to scalable, discrete differentiable communication [15]; however, to give agents the ability to control the amount of information they transmit, we introduce a variable-length coding scheme, which assigns each possible discrete messages to a unique binary encoding of variable length. We then derive a message regularization term that encourages message brevity by penalizing (an upper bound for) the expected number of bits in the message encodings.

We test our approach on a simple but illustrative example problem, in which a group of agents must navigate to randomly-chosen goal locations, but can only observe other agents' goals. This partial observability necessitates inter-agent communication. Here we show that with higher levels of message-length penalization, agents learn to transmit fewer bits, while still solving the task, and our approach therefore constitutes an effective way to achieve sparse inter-agent communication.

An additional highlight of our approach is that, because it builds upon communication learning through gradient backpropagation, it enables sparse communication learning

through imitation of expert actions alone [11], [16], [17]. In other words, agents can learn to communicate succinct discrete messages via demonstration of expert actions, without ever having communication behavior demonstrated to them.

In this paper, we first review prior works on multi-agent RL with communication, and specify the mathematical details of the problem. We then detail our approach to achieving sparse discrete inter-agent communication through gradient backpropagation, and present experimental results from a robot navigation task.

II. Prior Works

Approaches to multi-agent RL with inter-agent communication tend to fall into two broad categories: reinforced communication learning (RCL), and differentiable communication learning (DCL). RCL treats agent message selection similarly to action selection, and is trained with typical reinforcement learning techniques, such as Q-learning [12], or policy-gradient [6], [14]. This form of communication learning is extremely general, as it places virtually no assumptions on the affects messages will have on the environmental state transitions or the behavior of recipient agents. RCL also naturally handles discrete and/or noisy channels. RCL, however, has been shown to learn less efficiently than DCL, requiring drastically more episodes to achieve satisfactory policies [12].

DCL approaches, on the other hand, treat communication as a differentiable process, and train communication behavior through gradient backpropagation [15], [12], [13], [11], [10]. That is, since each agent's behavioral output (*i.e.*, its value function or stochastic policy) is recognized by DCL to be a function of the messages the agent receives, which in turn are functions of the parameters of the agents who sent the messages, gradients can be backpropagated from the receiving agent's behavioral output to the sending agents' parameters. Differentiable approaches have been shown to be far more efficient than RCL [12], [15], as agents need not rely solely on empirical observations to determine how their messages affects the behavior of other agents. An additional benefit of DCL is that it avoids the combinatorial explosion in size of the message space that exists for RCL[1]. Therefore, DCL can in theory scale more efficiently to large message spaces. One additional benefit of DCL over RCL is that it can also be used with imitation learning, without requiring communication behavior to be demonstrated [11].

The primary drawbacks to DCL are that it does not naturally handle discrete communication channels, or those with noise. Our prior work in [15] addressed these issues by introducing a stochastic message encoder and decoder, responsible for quantizing agents' real-valued communication output into a discrete message, and reconstructing an estimate of the original real-valued communication output from a received discrete message. With this addition, the encoder-channel-decoder system is a differentiable process, enabling discrete communication behavior to be learned via gradient backpropagation. In particular, the encoder-channel-decoder system is mathematically equivalent to an analog communication channel with additive, independent, uniform noise. Because the noise is independent of the messages, it does not bias the gradient estimates obtained.

[1]This exponential explosion is due to the fact that the size of the message space scales exponentially with the number of bits contained in messages.

III. Background

In this section, we define the problem of multi-agent RL with communication, and provide background on variable-length codes, which allows us to penalize agents' message
.

A. Multi-Agent RL with Communication

Consider an environment containing K agents. At each timestep, each agent selects actions independently of all other agents, according to a policy, parameterized by $\theta = (\theta_1, ...\theta_K)$, where θ_i denotes the parameters for the ith agent. These policies can either take the form of a value function, such as in independent Q-learning [18], or a stochastic policy, as in [6]. Actions selected by agents elicit a stochastic environmental state transition, according to the (unknown) state-transition distribution $S_{t+1} \sim p_{\mathbf{S}'|\mathbf{S},\mathbf{a}}(\mathbf{S}'|\mathbf{S} = S_t, \mathbf{a} = a_t)$, where $a_t = (a_t^{(1)}, ..., a_t^{(K)})$ represents the joint action at time t, and S_t denotes the state at time t. Each agent i then receives an observation $o_{t+1}^{(i)}$, distributed according to $(o_{t+1}^{(1)}, ..., o_{t+1}^{(K)}) = o_{t+1} \sim p_{\mathbf{o}|\mathbf{S}}(\cdot|S_{t+1})$, and reward $r_t^{(i)}$ according to $(r_t^{(1)}, ..., r_t^{(K)}) = r_t \sim p_{\mathbf{r}|\mathbf{S},\mathbf{a},\mathbf{S}'}(\cdot|\mathbf{S} = S_t, \mathbf{a} = a_t, \mathbf{S}' = S_{t+1})$. The goal of each agent i is to maximize its expected sum of discounted rewards, $\sum_{t=0}^{T} \gamma^t r_t^{(i)}$, where T represents the length of the episode. Learning is accomplished by updating the policy parameters according to some reinforcement learning (RL) algorithm.

The standard MARL problem can be augmented with inter-agent communication, which allows agents to send each other information that they would not normally have access to from local sensing alone. At each timestep, each agents sends a (possibly unique) discrete message to all agents with whom it is in contact. The set of recipients of a particular agent's message can be chosen either by the environment, by the agents themselves, or a combination thereof. In our approach, we assume the set of all agents that could possibly receive a message is predetermined (*e.g.*, by the environment) and the sender of the message can choose this message to be a "null message," effectively choosing not to send a message. In this paper, we assume messages sent by agents are received, error-free, by the recipient agent at a subsequent timestep, and form part of its observation.

B. Discrete Communication Learning via Backpropagation

Using the approach outlined in [15], it is possible to reparameterize a discrete communication process as a differentiable process. This is accomplished by introducing a stochastic message encoder, which takes as input a real-valued signal z, and quantizes z in a stochastic fashion into a discrete message. This discrete message can then be sent through the channel and received by another agent, who can

then decode the message according to a stochastic decoder, yielding an approximate reconstruction of the original real-valued signal generated by the sending agent. As shown in [15], the reconstruction differs from the original signal by additive, zero-mean, independent uniform noise, allowing the encoder/channel/decoder system to be reparameterized as a simple analog communication channel. Because under this reparameterization gradients can be backpropagated through the channel, inter-agent communication can be performed using an efficient DCL approach.

C. Variable-Length Codes

A code $C : S \to T$ is a unique mapping from symbols in a source alphabet S (*e.g.*, the Latin alphabet) to a target domain T (*e.g.*, the set of bit sequences $T = \{0, 1\}^*$). Elements of the target domain are called codewords. Variable-length codes, such as Huffman coding and arithmetic coding, use codewords of variable length. Variable-length codes are useful in data compression, because more common symbols in the source domain can be represented with shorter codewords, allowing the data to be represented with a number of bits closer to the theoretical minimum determined by the entropy of the source [19]. In our approach, we incorporate a variable-length code to provide agents with a means by which they can regulate the number of bits they send at a given timestep.

IV. Sparse Differentiable Discrete Communication Learning

For agents to learn to communicate sparsely, they must have the freedom to regulate the amount of information they send at each timestep, which so far has not been a feature of differentiable communication learning approaches. Rather than representing this choice with an additional set of actions available to the agents, we can incorporate this choice implicitly into agents' standard communication behavior by adopting a variable-length coding scheme for discrete messages. As in our previous work, we assume agents' real-valued communication output z is quantized into discrete message m (where timestep and agent indices are omitted for brevity). In this work, we introduce an additional coding step: following message quantization, message m is converted to a bit string, according to the variable-length code C, which we fix *a priori*. This binary message is what is subsequently sent through the communication channel. The variable-length code can be thought of as providing a means for agents to perform data compression on their messages. Whereas in typical data compression, the frequencies of source symbols are fixed, and a code is chosen to maximally compress the the source data, in our case we take the code to be fixed, and allow agents to modulate the frequencies of source symbols (in this case, discrete messages). In the ideal case, agents would choose to use messages with shorter binary representations more frequently, and use longer messages infrequently (only when needed). In our experiments, we adopt a coding scheme that maps successive values of discrete messages to successively higher binary representations, with $m = 0$ mapping to the null message (a message is not sent), $m = 1$ mapping to 0, $m = 2$ mapping to 1, $m = 3$ mapping to 01, and so on.

To encourage agents to send both shorter messages and communicate less frequently (only when necessary), we introduce a penalty term to the typical RL objective that corresponds to a cost for longer messages, yielding the objective

$$J(\theta) = \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r_t - \lambda \sum_{i=1}^{K}\sum_{j=1}^{K} \ell(\text{length}(m_{b,t}^{i\to j}))\right], \quad (1)$$

where $\text{length}(m_{b,t}^{i\to j})$ is the length of the binary message sent from agent i to agent j, ℓ is the penalty function, and λ is the penalty weighting. While ideally one might impose a constant cost per message bit, such a penalty term would not be differentiable. Instead, we adopt a surrogate penalty term computed from agents' real-valued communication outputs $z_t^{i\to j}$ (where i and j denote the indices of the sending and receiving agents, respectively), given by $\ell(z_t^{i\to j}) = \log_2\left(|M| z_t^{i\to j} + 1\right)$, where $|M|$ denotes the size of the set of discrete messages that agents may choose to send. The choice of this ℓ is based on the fact that ℓ can be shown to be an upper bound for expected number of bits in $m_{b,t}^{i\to j}$ given $z_t^{i\to j}$. This is because $\mathbb{E}\big[m_t^{i\to j}\big|z_t^{i\to j}\big] = |M| z_t^{i\to j}$ (where $m_t^{i\to j}$ represents the discrete, but non-binary, message from i to j). For our code, $\text{length}(m_{b,t}^{i\to j}) \le \log_2(m_t^{i\to j} + 1)$. Taking the conditional expectation given $z_t^{i\to j}$, we have that $\mathbb{E}\big[\text{length}(m_{b,t}^{i\to j})\big|z_t^{i\to j}\big] \le \mathbb{E}\big[\log_2(m_t^{i\to j} + 1)\big|z_t^{i\to j}\big]$. Applying Jensen's inequality, we arrive at $\mathbb{E}\big[\text{length}(m_{b,t}^{i\to j})\big|z_t^{i\to j}\big] \le \log_2\left(\mathbb{E}\big[m_t^{i\to j}\big|z_t^{i\to j}\big] + 1\right) = \log_2\left(|M| z_t^{i\to j} + 1\right)$. This penalty term can be thought of as a log-regularizer on $z_t^{i\to j}$, encouraging it to be as small, and messages to therefore be short.

It might appear at first that our coding scheme is overly idealistic, since it assumes no additional overhead bits are necessary for error correction or to make messages uniquely decodable. However, the message penalty we derive is equally suited to any code with codewords that differ in length from ours by multiplicative or additive constants. This is because multiplicative constants can be absorbed into the penalty weight, and additive constants simply impart a constant offset to the objective function and therefore do not change the optimal solution. This property allows our approach to function with a wide range of possible codes.

These modifications result in the final objective function for our problem:

$$J(\theta) = \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r_t - \lambda \sum_{i=1}^{K}\sum_{j=1}^{K} \log_2\left(|M| z_t^{i\to j} + 1\right)\right]. \quad (2)$$

A Monte-Carlo estimate of the gradient of J can be formed according, according to:

$$\nabla_\theta J(\theta) \approx g_{pg} - \lambda \nabla_\theta \left(\sum_{i=1}^{K} \sum_{j=1}^{K} \log_2 \left(|M| z_t^{i \to j} + 1 \right) \right), \tag{3}$$

where g_{pg} denotes a policy-gradient estimator.

The message-length penalty term can also be combined with imitation learning. In this case, the objective function becomes

$$J(\theta) = -L_{IL}(\theta) - \lambda \sum_{i=1}^{K} \sum_{j=1}^{K} \log_2 \left(|M| z_t^{i \to j} + 1 \right), \tag{4}$$

where $L_{IL}(\theta)$ is the imitation loss. One possible choice of imitation loss is to minimize the negative log-likelihood of expert actions, in which case $L_{IL}(\theta) = -\mathbb{E}\left[\log \pi(a_{exp,t}|S_t)\right]$, where π is the joint policy of the imitation learners, and $a_{exp,t}$ the expert action at time t.

V. Simulation Experiments

In this section, we describe the experiments we carried out in simulation to verify our approach on a multi-robot navigation task, and present results from reinforcement and imitation learning.

A. Robot Navigation Task

We demonstrate the ability of our proposed technique to simultaneously learn high-quality behavioral and sparse communication policies, on a simple but illustrative example task. We choose this task to demonstrate our technique because, while it is simple, it possesses some key difficulties of many MARL problems, such as problems of delayed reward (agents actions early in the episode affect reward many timesteps in the future), and difficulties in credit assignment due to shared reward (agents are rewarded based on the quality of not only their actions, but the actions of the other agents in the environment). We consider environments with 3 agents that can continuously move in the 2-dimensional plane. Each agent is tasked with finding its goal location, which can be located anywhere within a square of side length 2, centered at the origin. Agents do not observe their goals directly, and instead observe only the goals of the other two agents, in addition to their own coordinates and velocities. Agents must therefore communicate with each other to find their goals. Goals move randomly according to the following motion model: with a 0.05 chance at each timestep, goals discretely move to a new location sampled from a uniform distribution over the 2×2 square area in which goals can appear. The size of the square and acceleration of the agents is such that it is physically possible for agents to reach a goal placed anywhere in the square.

Agents are modeled as point masses, and select actions at each timestep, corresponding to a force in either the up, down, left, or right directions, or no force. Agents move according to the second-order dynamics model of such a point mass with an applied force, *i.e.*, acceleration of the agent is proportional to its applied force. The episode terminates either when all agents are within 0.1 units of their goals, or 256 timesteps have elapsed.

At each timestep, agents are able to send a discrete message to both of the other agents in the environment. To compute these discrete messages, agents first output a real-valued 10-element communication signal z, where each element $z[i] \in [0, 1]$, which is then quantized according to the stochastic encoding scheme described in [15], yielding an integer. This integer is then converted to a bitstring according to the code descried in IV. Upon receiving a bitstring message, agents first convert this bitstring back to an integer, and then to a real-valued reconstruction of the sending agent's original communication signal according to the stochastic decoder described in [15]. This real-valued reconstruction then becomes part of the recipient agent's observation. Communications are penalized according to the method explained in IV.

To highlight the flexibility of our technique, we train agents with two distinct methods: RL, where we test our approach with penalty weights of $\lambda = 0.01$, 0.005, 0.001, and 0, and imitation learning (IL), where we test our approach with penalty weights of $\lambda = 0.001$, 0.0007, 0.0004, and 0. For RL, we use the actor-critic algorithm [20], in which the policy gradient estimator described in Eq.(3) is given by

$$g_{pg} = \nabla_{\theta_i} \log \pi^{(i)}(a_t|o_{\leq t}) \left(\sum_{k=t}^{T} \gamma^{k-t} r_k^{(i)} - V^{(i)}(S_t) \right), \tag{5}$$

where $r_k^{(i)}$ denotes the ith agent's reward at time k, and $\pi^{(i)}(a_t|O_{\leq t})$ denotes the $i$the agent's policy, which we parameterize as a recurrent neural network, and is conditioned upon all past observations $O_{\leq t}$. $V^{(i)}(S_t)$ is a learned value estimate, parameterized by a feedforward neural network. We allow the value estimate to be conditioned on full state information, consistent with the centralized-training decentralized-execution paradigm.

When all agents have reached their goals, each agent receives a reward of 1200 and the episode terminates. When not every agent has reached its goal, the agent's reward r_t, for $t = 0, ..., T$, is computed as a linear combination of an individual component, as well as a group component to encourage cooperation. The agent's individual component consists of either a negative reward equal to the negative of the squared distance between itself and its goal, computed at every timestep, or a positive reward of magnitude 250 for being within 0.1 units of its goal. The group component is equal to the sum of the agents' individual components.

When using imitation learning, we train agents to predict the actions of an expert in a supervised manner, as described in IV. In our experiments, our expert is composed of a group of agents fully trained with reinforcement learning. During training, the imitation learner has access only to the agents' expert actions, but not their messages.

B. Results

For all three message-length penalty weights we tested, the agents successfully learned to communicate, as evidenced

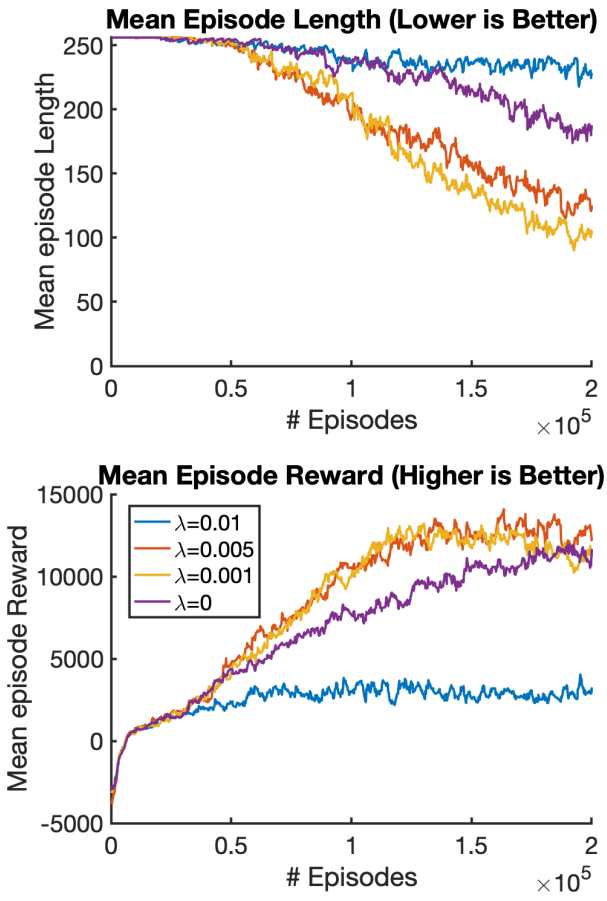

Fig. 1. Average episode reward (top) and length (bottom) as training progresses. For all three values of message penalty weight (λ), agents are able to solve the task; however, note that performance declines with higher values of λ, as agents are not able to communicate as freely.

by the fact that agents were able to find their goals at a substantially higher frequency than if no communication was permitted (in which case, agents virtually never find their goals by the end of an episode). As expected, we found that agent communication reliably decreased as the penalty weight increased, indicating that the penalty weight provides a useful means by which to regulate the amount of communication agents engage in. Moreover, as the penalty weighting increased, the final average reward agents attained was not substantially impacted, except for the largest value of message penalty weight tested, despite over a 2-fold reduction in information transmission (Fig. 2), indicating that agents learned to trim predominately unnecessary information from their messages (Fig. 1, Fig. 2).

Our experiments with imitation learning showed that agents trained to imitate expert actions also learned sparse communication capabilities. Here, agents were capable of achieving nearly the same reward and episode lengths even when sending far fewer bits than their unconstrained values (Table I).

Penalty Weight λ	Mean Episode Length	Mean Group Rewards
0.0001	139.5	13053
0.0004	144	13023
0.0007	146.6	13026
0.001	254.6	3710

TABLE I

MEAN EPISODE LENGTH (LOWER IS BETTER) AND GROUP REWARDS (HIGHER IS BETTER) FOR DIFFERENT VALUES OF λ DURING IMITATION LEARNING.

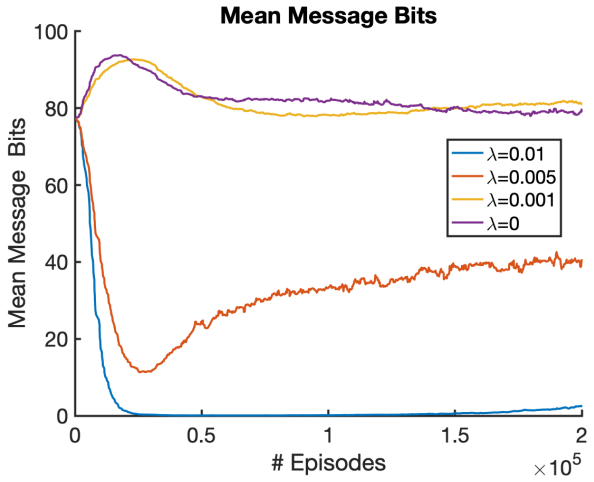

Fig. 2. Mean number of message bits transmitted per timestep during training. As expected, larger values of the message penalty weight (λ) result in fewer bits transmitted.

VI. CONCLUSION

In this paper, we demonstrated a method for simultaneously learning action selection in a multi-agent setting, as well as sparse, discrete inter-agent communication. We built upon past approaches to differentiable discrete communication learning with a variable-length coding procedure, that allows the choice of the number of bits transmitted by agents to be implicit to message selection. We additionally derive a differentiable message-length penalty that corresponds to an upper bound on expected number of message bits sent by agents, and incorporate this into the learning objective function, to encourage sparse communication. This penalty term can be thought of as a log-regularizer on the real-valued communication output that agents generate. Through experiments on a robot navigation problem, we demonstrate that this approach learns multi-agent policies that requires little bandwidth while still solving the task. We also find that agents transmit fewer bits when trained with a higher message-length penalty weight, allowing us to conclude that our message-length penalty provides a useful tool to adjust the extent to which agents communicate. Finally, We demonstrate our approach is able to learn effective behavior through either reinforcement learning or imitation learning. In this case, message bits could be decreased substantially (>2x) from their unconstrained values, while expected reward remained relatively unaffected for all but the largest weight penalty. During imitation learning, communication behavior need not be demonstrated, enabling a very flexible

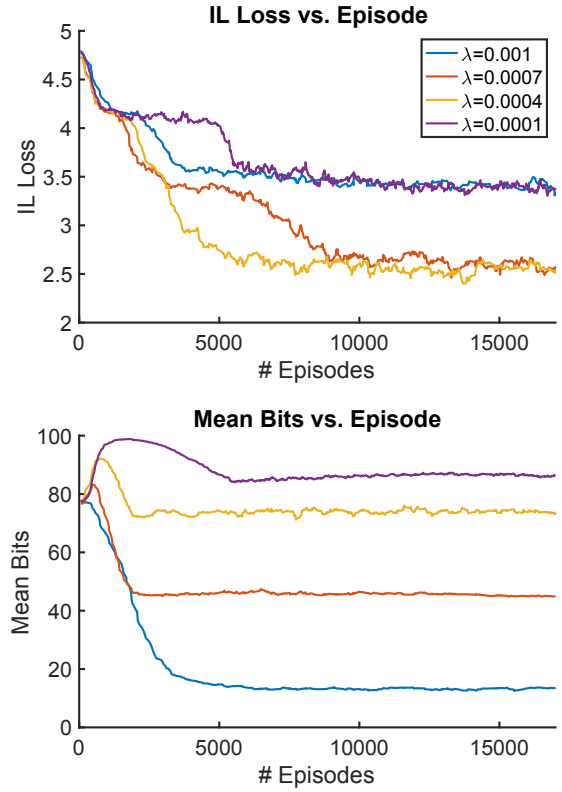

Fig. 3. Imitation loss and mean number of message bits transmitted per timestep during imitation learning-based training. As expected, larger values of the message penalty weight (λ) result in fewer bits transmitted.

choice of expert (for example, a centralized expert that does not undergo inter-agent communication).

In future works, we will investigate learned coding procedures. In this work, we fixed a code *a priori* that mapped larger values of discrete messages to longer binary messages. We believe it may be more effective to allow agents to learn their own mapping, so that message length can be decoupled from message value.

We will also consider communication channels with noise, which differentiable communications learning approaches, other than our prior work, have not considered. Our approach already permits the use of more sophisticated codes than the simplistic one we described, such as codes that incorporate redundancy into messages to enable error-correction (error correcting codes). Past work has explored learned channel coding, with the objective of learning optimal error-correcting codes for particular channel noise models [21], [22]. This work suggests it may be possible, in environments with channel noise, to learn a coding scheme that optimizes the trade-off between number of bits transmitted, message expressively, and error rates. Longer messages can be more expressive or less prone to errors, but place a greater burden on the communication network.

References

[1] OpenAI, "Openai five," https://blog.openai.com/openai-five/, 2018.

[2] M. Jaderberg, W. M. Czarnecki, I. Dunning, L. Marris, G. Lever, A. G. Castañeda, C. Beattie, N. C. Rabinowitz, A. S. Morcos, A. Ruderman, *et al.*, "Human-level performance in 3d multiplayer games with population-based reinforcement learning," *Science*, vol. 364, no. 6443, pp. 859–865, 2019.

[3] B. Baker, I. Kanitscheider, T. Markov, Y. Wu, G. Powell, B. McGrew, and I. Mordatch, "Emergent tool use from multi-agent autocurricula," *arXiv preprint arXiv:1909.07528*, 2019.

[4] G. Sartoretti, J. Kerr, Y. Shi, G. Wagner, T. S. Kumar, S. Koenig, and H. Choset, "Primal: Pathfinding via reinforcement and imitation multi-agent learning," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2378–2385, 2019.

[5] G. Sartoretti, W. Paivine, Y. Shi, Y. Wu, and H. Choset, "Distributed learning of decentralized control policies for articulated mobile robots," *IEEE Transactions on Robotics*, vol. 35, no. 5, pp. 1109–1122, 2019.

[6] R. Lowe, Y. Wu, A. Tamar, J. Harb, O. P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Advances in neural information processing systems*, 2017, pp. 6379–6390.

[7] A. Tampuu, T. Matiisen, D. Kodelja, I. Kuzovkin, K. Korjus, J. Aru, J. Aru, and R. Vicente, "Multiagent cooperation and competition with deep reinforcement learning," *PloS one*, pp. 1–12, 2015.

[8] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3133–3174, 2019.

[9] T. T. Nguyen, N. D. Nguyen, and S. Nahavandi, "Deep reinforcement learning for multi-agent systems: A review of challenges, solutions and applications," *arXiv preprint arXiv:1812.11794*, 2018.

[10] S. Sukhbaatar, R. Fergus, *et al.*, "Learning multiagent communication with backpropagation," in *Advances in neural information processing systems*, 2016, pp. 2244–2252.

[11] J. Paulos, S. W. Chen, D. Shishika, and V. Kumar, "Decentralization of multiagent policies by learning what to communicate," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 7990–7996.

[12] J. Foerster, I. A. Assael, N. De Freitas, and S. Whiteson, "Learning to communicate with deep multi-agent reinforcement learning," in *Advances in neural information processing systems*, 2016, pp. 2137–2145.

[13] I. Mordatch and P. Abbeel, "Emergence of grounded compositional language in multi-agent populations," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[14] B. Freed, G. Sartoretti, and H. Choset, "Simultaneous policy and discrete communication learning for multi-agent cooperation," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2498–2505, April 2020.

[15] B. Freed, G. Sartoretti, J. Hu, and H. Choset, "Communication learning via backpropagation in discrete channels with unknown noise," in *AAAI 2020 - 34th Conference on Artificial Intelligence*, 2020.

[16] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: A survey of learning methods," *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, pp. 1–35, 2017.

[17] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Survey: Robot programming by demonstration," *Handbook of robotics*, vol. 59, 2008.

[18] M. Tan, "Multi-agent reinforcement learning: Independent vs. cooperative agents," in *Proceedings of the tenth international conference on machine learning*, 1993, pp. 330–337.

[19] D. Salomon, *Variable-length codes for data compression*. Springer Science & Business Media, 2007.

[20] V. R. Konda and J. N. Tsitsiklis, "Actor-critic algorithms," in *Advances in neural information processing systems*, 2000, pp. 1008–1014.

[21] R. Fritschek, R. F. Schaefer, and G. Wunder, "Deep learning for channel coding via neural mutual information estimation," in *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2019, pp. 1–5.

[22] K. Choi, K. Tatwawadi, A. Grover, T. Weissman, and S. Ermon, "Neural joint source-channel coding," 2018.