Communication Learning via Backpropagation in Discrete Channels with Unknown Noise

Paper ID: 5764

Abstract

This work focuses on multi-agent reinforcement learning (RL) with inter-agent communication, in which communication is differentiable and optimized through backpropagation. Such differentiable approaches tend to converge more quickly to higher-quality policies compared to techniques that treat communication as actions in a traditional RL framework. However, modern communication networks (e.g., Wi-Fi or Bluetooth) rely on discrete communication channels, for which existing differentiable approaches that consider real-valued messages cannot be directly applied, or require biased gradient estimators. Some works have overcome this problem by treating the message space as an extension of the action space, and use standard RL to optimize message selection, but these methods tend to converge slower and to inferior policies. In this paper, we propose a stochastic message encoding/decoding procedure that makes a discrete communication channel mathematically equivalent to an analog channel with additive noise, through which gradients can be backpropagated. Additionally, we introduce an encryption step for use in noisy channels that forces channel noise to be message-independent, allowing us to compute unbiased derivative estimates even in the presence of unknown channel noise. To the best of our knowledge, this work presents the first differentiable communication learning approach that can compute unbiased derivatives through channels with unknown noise. We demonstrate the effectiveness of our approach in two example multi-robot tasks: a path finding and a collaborative search problem. There, we show that our approach achieves learning speed and performance similar to differentiable communication learning with real-valued messages (i.e., unlimited communication bandwidth), while naturally handling more realistic real-world communication constraints. **Content Areas:** Multi-Agent Communication, Reinforcement Learning.

1 Introduction

Reinforcement learning has recently been successfully applied to complex multi-agent control problems such as DOTA, Starcraft II, and virtual capture the flag (Jaderberg et al. 2019; OpenAI 2018). Many multi-agent reinforcement-learning (MARL) approaches seek to learn decentralized policies, meaning each agent selects actions independently from all other agents, conditioned only on information available to that particular agent (Bernstein et al. 2002; Gupta, Egorov, and Kochenderfer 2017; Sartoretti et al. 2019; Foerster et al. 2017). Such decentralized approaches offer major scalability and parallelizability advantages over centralized planning approaches. Computational complexity of decentralized approaches scales linearly with the team size (as opposed to exponentially, as for typical centralized planners), and action selection for each agent can occur completely in parallel (Busoniu, Babuška, and De Schutter 2010). However, decentralized approaches pay for these benefits with potentially sub-obtimal policies or lower degrees of agent coordination. Particularly in partially-observable environments, a decentralized approach in which agents make decisions based soleley on their own limited observations, cannot make as informed decisions as a centralized planner, which conditions all actions on all available information.

We consider MARL problems in which agents have the ability to communicate with each other. This communication ability offers agents a way to selectively exchange information, potentially allowing them to make more informed decisions, while still achieving the same scalability and parallelizability advantages of a purely decentralized approach. However, the addition of communication abilities also increases the difficulty of the learning problem, as agents now have to make decisions not only about what actions to select, but also what information to send, how to encode it, and how to interpret messages received from other agents. This paper focuses on multi-agent RL with communication that are readily applicable to real-world robotics problems.

In this paper, we present a novel approach to differentiable communication learning that utilizes a randomized message encoding scheme to make discrete (and therefore non-differentiable) communication channels behave mathematically like a differentiable, analog communication channel. We can use this technique to obtain unbiased, low variance gradient estimates through a discrete communication channel. Additionally, we show how our approach can be generalized to communication channels with arbitrary unknown noise, which existing approaches to differentiable communication learning have not been able to deal with.

Copyright © 2019, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

2 Background

2.1 Multi-agent Reinforcement Learning with Communication

Many past approaches to multi-agent reinforcement learning with inter-agent communication fall into one of two general categories: 1) approaches in which communication is treated as a differentiable process, allowing communication behavior to be optimized via backpropagation (differentiable approaches) (Foerster et al. 2016; Sukhbaatar, Szlam, and Fergus 2016; Mordatch and Abbeel 2017; Paulos et al. 2019), and 2) approaches in which messages are treated as an extension to the action space, and communication behavior is optimized via standard reinforcement learning (reinforced communication learning, RCL) (Foerster et al. 2016; Lowe et al. 2017).

RCL approaches tend to be more general than differentiable approaches(Lowe et al. 2017). This is because the communication channel, and all downstream processing of messages selected by agents, is considered to be part of the environment, and is therefore treated as a black box. No assumptions are made about what influence a particular message may have on other agents or future state transitions. RCL approaches therefore naturally handle unknown channel noise (Lowe et al. 2017). Additionally RCL naturally handles discrete communication channels because RCL does not require backpropagation through the communication channel, so message selection can be non-differentiable.

The downside of RCL's generality is that it typically requires significantly more learning updates to converge to a satisfactory policy, compared to differentiable approaches (Foerster et al. 2016). In RCL, agents are not explicitly provided with knowledge of how their message selection impacts the behavioral policy of other agents, and must instead deduce this influence through repeated trial and error. On the other hand, differentiable approaches allow one to explicitly compute the derivative of recipient agents' behavioral policy or action-value function with respect to the sending agent's communication policy parameters. This allows differentiable approaches to converge to better policies after fewer learning updates compared to RCL.

One additional advantage to differentiable approaches is that because the objective function used to optimize communication in general does not directly depend on communication behavior, it is possible to train communication behavior via imitation learning, with no explicit expert demonstrations of communication. This is useful when one can generate expert action demonstrations but not expert communication demonstrations, e.g., when a communication-unaware centralized expert is present (Paulos et al. 2019).

Several differentiable approaches in the past have allowed agents to exchange real-valued messages(Foerster et al. 2017; Sukhbaatar, Szlam, and Fergus 2016). Differentiable approaches that consider real-valued communication signals cannot naturally handle discrete communication channels, as a discrete channel is non-differentiable. Some recent approaches have circumvented this problem by using biased gradient estimators (Foerster et al. 2016; Mordatch and Abbeel 2017; Lowe et al. 2017); however, biased estimators typically require additional tuning parameters or more complex training techniques such as annealing (Foerster et al. 2016; Mordatch and Abbeel 2017; Jang, Gu, and Poole 2016). Moreover, biased gradient estimators lack the convergence guarantees of unbiased ones. Additionally, to the best of our knowledge, existing differentiable approaches cannot function with unknown channel noise, because the channel then represents an unknown stochastic function whose derivatives are consequently unknown. The fact that differentiable approaches are restricted to channels with no/known noise limits their applicability to real-world robotic systems.

3 Theory

In this section we explain our approach to differentiable communication learning with a discrete communication channel. We first focus on the case in which channel noise is not present, and then expand our approach to the noisy channel case. Throughout this paper, we assume centralized training, meaning that all training data (agent observations and ground-truth communication signals) can be thought of as being sent to a centralized server where learning updates are computed for each agent, as is done in many multi-agent RL approaches, such as (Kraemer and Banerjee 2016; Foerster et al. 2017; 2016). We feel that this is a reasonable assumption, because often training takes place in a setting in which additional state information is available than what will be available to agents at execution time. However, we also assume decentralized execution, meaning that at execution time (i.e., after the training has finished/converged), agents do not need to transfer data among themselves or with a centralized server, other than via the communication channels available to them within the environment.

3.1 Communication in a Discrete, Noise-Free Channel

Consider a multi-agent reinforcement learning problem with K agents. Assume that, at a given timestep, a particular agent (Alice) is able to communicate to another agent (Bob) via a noise-free discrete communication channel with capacity C (in bits). For now, we neglect details associated with how agent policies are trained, and assume some sort of gradient-based RL approach is used in which a loss function is computed from data gathered during a set of agent experiences, which is then differentiated with respect to agents' policy parameters to compute a gradient update. This assumption is sufficiently general to encapsulate most prominent RL algorithms, such as Q-learning and variants of policy gradient. Additionally, we do not specify what factors determine which agents can communicate, as this is problem-specific. Here we are primarily concerned with how gradients are backpropagated through the channel.

Because the communication channel is discrete, any message m sent through the channel from Alice to Bob is taken from a finite set of possible messages, *i.e.*, $m \in M$, where $M = \{m_1, ..., m_{|M|}\}$, and $|M| \leq 2^C$. If we wish to perform differentiable communication learning in such an environment, we need to compute the derivative of the input of Bob

with respect to the output of Alice. However, the derivative of the channel output with respect to the input is not well-defined, making typical backpropagation impossible.

Existing approaches use a biased gradient estimator to circumvent this problem, such as the gumbell-softmax estimator. We propose an alternative approach, described below, in which Alice generates a real-valued communication signal z, which is encoded into a discrete message by a stochastic quantization procedure (randomized encoder). The resulting discrete message m is sent through the communication channel and is received by Bob, who then uses this discrete message to compute an approximation of the original real-valued signal generated by Alice, $\hat{z}$, using a stochastic dequantization procedure (randomized decoder) (fig. 1). We show that the encoder/channel/decoder system is mathematically equivalent to an analog communication channel with additive noise, allowing the partial derivative of Bob's reconstruction with respect to Alice's communication signal to be computed.

Randomized Encoder The purpose of the randomized encoder is to convert a real-valued output from Alice into a discrete message that can be sent through the channel to Bob. One simple approach would be to quantize z directly; however, the derivative of this operation then vanishes everywhere except at the boundaries of the quantization intervals, where it is undefined, precluding the use of this simple idea in a differentiable communication setting.

Instead, we propose the following non-deterministic quantization method: for simplicity, assume z is scalar and $z \in [0, 1]$, however the encoding process can be easily generalized to vectors of arbitrary length. To encode z, we first perturb it with noise ϵ, which is sampled independently at each timestep from a uniform distribution centered at 0 with width $\delta = \frac{1}{|M|}$, *i.e.*, $z' = z + \epsilon$, where $\epsilon \sim U\left(\cdot; -\frac{\delta}{2}, +\frac{\delta}{2}\right)$. z' now ranges from $-\frac{\delta}{2}$ to $1 + \frac{\delta}{2}$. z' is then quantized into one of $|M|+1$ quantization intervals of width δ, spaced uniformly along the range of z'. The index of the quantization interval is taken to be the discrete message, with the special case that the last quantization interval is identified with the first, so that there are exactly $|M|$ possible values for m (fig. 2). In other words, the discrete message to be sent through the channel is given by:

$$m = \left\lfloor \frac{(z + \epsilon + \delta/2) \pmod 1}{\delta} \right\rfloor.$$

Note that here $m \in \{0, ..., |M| - 1\}$, however this integer representation can be converted to binary for use in a digital communication channel.

Randomized Decoder The goal of the randomized decoder is to reconstruct z from m, denoted $\hat{z}$, such that $\hat{z}$ can be represented as a continuous, differentiable function of z, and a noise source that is independent of z.

Upon receiving m, Bob computes the reconstruction as $\hat{z} = C(m) - \epsilon$, where $C(m)$ denotes the center of the quantization interval corresponding m, $C(m) = \delta(m + \frac{1}{2})$ (fig. 2). One special case exists in which the decoding process yields $\hat{z} < -\frac{\delta}{2}$, in which $\hat{z} = 1 + C(m) - \epsilon$. In short, the reconstruction is given by:

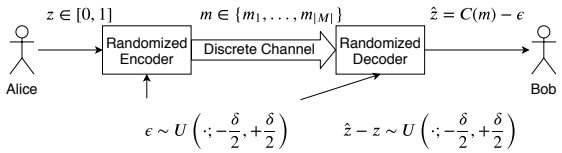

Figure 1: Randomized encoder/channel/decoder: The real-valued output z from agent Alice is converted into a discrete message m by the randomized encoder, which additionally takes random noise variable ϵ. The message m is then sent through the channel to agent Bob, who decodes it into $\hat{z}$ (a reconstruction of z), using the randomized decoder, which also takes ϵ as an input. Under the simple reparameterization $\hat{z} = z + e$, $\hat{z}$ becomes a continuous, differentiable function of z, allowing gradients to be naturally backpropagated through the discrete communication channel.

$$\hat{z} = \begin{cases} \delta(m + \frac{1}{2}), & \text{if } \delta(m + \frac{1}{2}) > -\frac{\delta}{2} \\ 1 + \delta(m + \frac{1}{2}), & \text{otherwise.} \end{cases} \quad (1)$$

We prove in the supplementary material that using the particular encoding/decoding procedure detailed above, $\hat{z}$ can be reparameterized according to $\hat{z} = z + e$, where $e \sim U(\cdot; -\frac{\delta}{2}, +\frac{\delta}{2})$. Under this reparameterization, the partial derivative of $\hat{z}$ with respect to z is precisely 1. The encoder/channel/decoder system therefore becomes mathematically equivalent to an analogous situation in which we send real-valued z directly through an analog communication channel, that introduces additive noise to the signal, but is nonetheless differentiable.

Note that Bob requires knowledge of the value of ϵ that Alice used to encode m, which we do not send through the channel. This is not a problem, because ϵ does not depend upon z and we can therefore assume the agents agree ahead of time upon a sequence of ϵ values to use for each timestep. Practically, this would most likely amount to all agents using the same pseudorandom number generator with the same seed. To generalize the complete encoding/decoding procedure to cases in which z is a vector rather than a scalar, one can apply the above procedure element-wise with a distinct value of ϵ to each component of z.

3.2 Communication in a Discrete Channel with Unknown Noise

Here we present a method for generalizing our approach to channels with unknown noise. We define a noisy channel to be on in which the output of the channel, $\hat{m}$, is not necessarily the same as the input m, but instead is distributed according to some unknown distribution that we assume can depend on the input to the channel and the state, *i.e.*, $\hat{m} \sim P(\cdot|m, S)$. We assume this distribution is completely unknown to us. Consequently, the message reconstruction error will no longer be distributed according to a known distribution, and can depend on z and S in unknown ways.

To make the difficulty introduced by unknown channel noise more clear, we can express our message reconstruction in the following form:

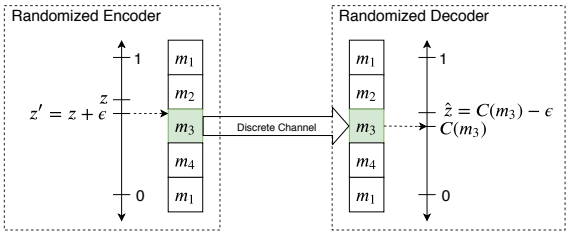

Figure 2: Randomized encoder and decoder: The real-valued output z from agent Alice is first perturbed by random noise ϵ, which is sampled from a zero-centered uniform distribution with width equal to the quantization interval to form z'. z' is then quantized into one of the $|M|$ quantization intervals, the index of which is taken to be the discrete message m. The message m is finally sent through the channel to agent Bob, who uses it to compute a reconstruction of z by subtracting ϵ from the center of the quantization interval corresponding to m.

$$\hat{z} = h(z, \alpha(z, S, \beta)),$$

where h is a known, deterministic function, α is an unknown function representing channel noise, and β is some independent noise source that does not depend on z or S. Differentiating $\hat{z}$ w.r.t. z, we see that the derivative contains an unknown term, namely the derivative of α w.r.t. z:

$$\frac{\partial \hat{z}}{\partial z} = \frac{\partial h(z,\alpha)}{\partial z} + \frac{\partial h(z,\alpha)}{\partial \alpha(z,S)} \frac{\partial \alpha(z,S)}{\partial z}.$$

The only situation in which we could backpropagate through the channel without knowing the exact form of α is if we know α does not depend on z, which is not generally the case for the procedure we've presented so far. However, recall that what is actually sent through the communication channel is m. Because we have a choice in how z is encoded into m, we have some influence over the effect of channel noise on z. In fact, we can use this capability to render reconstruction error completely independent of z, so that $\frac{\partial \hat{z}}{\partial z}$ is known. Consider the following procedure: z is encoded into m in a manner similar to that detailed in 3.1. After this, a one-to-one mapping is randomly generated from each message to every other message in M, described by the permutation sequence Q. m is then mapped according to Q to $\bar{m}$, which is sent through the channel. Here we will refer to this step as the "encryption" step, as it is effectively equivalent to a simple form of encryption, which can be thought of as concealing the value of m from the environment, thus limiting the ways in which the decrypted message distribution can vary with m. The recipient agent receives $\hat{\bar{m}}$, which it then decrypts by mapping it according to Q^{-1} to an estimate of m, which we denote $\hat{m}$. This is then converted to $\hat{z}$ via the a randomized decoder similar to that described in 3.1 (fig. 3).

We make some slight modifications to our randomized encoder for use in noisy channels. We now take communication output z to be a vector on the unit circle. This representation allows z to maintain exactly 1 bounded degree of freedom; however, it allows us to eliminate the discontinuity associated with the modulo used in the noise-free technique described in 3.1, because a vector can vary smoothly even

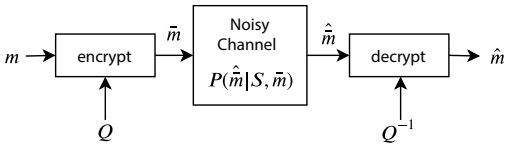

Figure 3: Encryption technique for channels with unknown noise. After discrete message m is generated, it is encrypted according to permutation sequence Q into $\bar{m}$. $\bar{m}$ is then sent through the communication channel, which outputs $\hat{\bar{m}}$, which is not necessarily the same as $\bar{m}$. $\hat{\bar{m}}$ is then decrypted (mapped by Q^{-1} to form $\hat{m}$, which can be thought of as an estimate of the original message m. With these encryption/decryption steps, the apparent channel noise introduced in the mapping form m to $\hat{m}$ takes on a known form that allows to compute unbiased gradients through the channel.

when its angle jumps from $-\pi$ to $+\pi$. This modification will be crucial for representing the communication channel as a differentiable function in the presence of channel noise. We again perturb z by random noise $\epsilon \sim U(\cdot; -\frac{\delta}{2}, +\frac{\delta}{2})$. However, to maintain the unit circle constraint, ϵ is taken to be an angular perturbation, which we apply to z by multiplication with the 2-dimensional rotation matrix corresponding to ϵ, denoted $R(\epsilon$, to form $z' = R(\epsilon)z$. z' is then quantized into one of $|M|$ quantization intervals, each one corresponding to a particular angular range on the unit circle, to form discrete message m. Because the range of angles of z' is now $[-\pi, +\pi)$, the quantization intervals have width $\delta = \frac{2\pi}{|M|}$. (fig. 4). In other words, m is given by:

$$m = \left\lfloor \frac{\arctan2(z'_2, z'_1) + \delta/2)}{\delta} \right\rfloor,$$

where z'_1 and z'_2 are the first and second elements of z' respectively, and $z' = R(\epsilon)z$.

Following the quantization, m is then encrypted into $\bar{m}$ by Q, which is then sent through the communication channel. Upon receiving output of the communication channel $\hat{\bar{m}}$, Bob then decrypts $\hat{\bar{m}}$ into $\hat{m}$ (maps according to Q^{-1}), and estimates z according to:

$$\hat{z} = R(-\epsilon)C(\hat{m}),$$

where $C(m)$ is the vector corresponding to the center of the angular range represented by m.

In the presented communication procedure, there are two sources of error that are introduced into Bob's reconstruction of Alice's real-valued communication output. The first is due to the fact that we have a limit on information transfer rate through the channel, placing a fundamental limit on how precisely we can reconstruct z, even in a noise-free channel. This form of noise is present whenever the communication channel does not corrupt the discrete message, as described in sec. 3.1, *i.e.*, when $\hat{\bar{m}} = \bar{m}$. In this non-corrupting case, we claim (and show in the supplementary material), that the angular error distribution of $\hat{z}$ is uniform, independent of z, and zero-centered, in particular $\hat{z} = R(e)z$, where $e|\hat{m} = \hat{\bar{m}} \sim U(-\frac{\delta}{2}, +\frac{\delta}{2})$. The second source of error is due to channel noise, and comes into effect whenever the discrete message is corrupted, *i.e.*, $\hat{\bar{m}} \neq \bar{m}$. In the supple-

mentary material, we show that in this case, the angular error distribution is still zero-centered, but now has uniform density everywhere except within the interval $[-\frac{\delta}{2}, +\frac{\delta}{2}]$. $\hat{z}$ can therefore be represented in the following way:

$$\hat{z} = R(e)z,$$

where e is distributed according to a mixture of the two distributions described above. In particular, we show in the supplementary material that e is distributed according to:

$$P_{\mathbf{e}|\mathbf{z},\mathbf{S}}(e|z,S) = \begin{cases} \frac{1}{\delta}(1 - P_e(S)), & -\frac{\delta}{2} \leq e < -\frac{\delta}{2} \\ \frac{1}{2\pi-\delta}P_e(S), & -\pi \leq e < -\frac{\delta}{2} \\ \frac{1}{2\pi-\delta}P_e(S), & +\frac{\delta}{2} \leq e < \pi \end{cases} \quad (2)$$

where $P_e(S) = \frac{1}{|M|}\sum_{m_{in} \in M} P(m_{out} = m_{in}|S, m_{in})$ is the state-dependent average channel error rate, with m_{in} and m_{out} representing the channel input/output, respectively. Since e is independent of z with the introduction of our encryption step, it is possible to compute the Jacobian representing the partial derivatives of $\hat{z}$ with respect to z, regardless of the form of channel noise, given by: $\frac{\partial \hat{z}}{\partial z} = R(e)$.

Similar to the noise-free case, to generalize the entire procedure to multi-elements messages (*i.e.*, z is a collection of unit-length vectors rather than a single vector), one can simply apply the above procedure to each element of z independently, using unique values of ϵ and Q for each element.

4 Experiments

We compare the performance of our approach to a differentiable communication learning approach with real-valued messages, and a reinforced communication learning approach, on two simple but illustrative example tasks, described below, with a noise-free communication channel. Additionally, we compare the performance of our encryption-based, channel-noise-tolerant approach to the same RCL approach, on the same two example tasks, but introducing message-dependent channel noise.

4.1 Implementation Details

The actor-critic algorithm is used as the reinforcement learning algorithm for all experiments, with separate actor and critic networks. Both actor and critic networks for all tasks are composed of a convolutional stack followed by two fully-connected layers. The policy networks used in the search task also use a simple single-layer recurrent neural network to help them remember areas they have previously explored and therefore do not need to revisit. Because the critic network is used only during training, we choose to give it full state observability in all experiments. During training, the policy networks are given only information they are will have during execution time. For a given task, the same architecture is used for both the differentiable and RCL approaches we test, except for differences to the message inputs and outputs necessitated by the differences between the two approaches. In both tasks, we consider 2 agents that are each able to send the other 40 bits of information at each timestep. In both the differentiable and RCL approaches we test, policy parameters are shared across agents, and each agents' action policies are represented as a categorical distribution over discrete actions. In our RCL approach, messages between agents are strings of 40 bits, and the communication policy of each agent (from which a message is sampled and sent to the other agent at each timestep) is represented as a collection of independent Bernoulli distributions representing the probability of each corresponding message bit being a 1. In our differentiable approach, agents output a real-valued communication signal with 10 elements at each timestep. Each element of this communication signal is then discretized independently into one of 16 discrete message elements, meaning each element can be thought of as a 4-bit word (a nibble), allowing for a total of $16^{10} = 2^{40}$ different possible messages, equivalent to 40 bits.

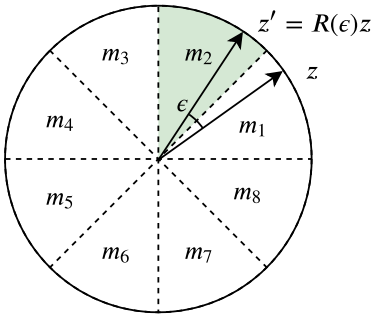

Figure 4: Randomized encoder for channels with unknown noise: the real-valued vector output z from agent Alice is constrained to lie on the unit circle. It is then rotated by random noise ϵ, which is sampled from a zero-centered uniform distribution with width equal to the quantization interval to form z'. z' is then quantized into one of $|M|$ quantization intervals, each corresponding to a particular angular range, and the index of which is taken to be the discrete message m. In the case depicted above, $m = m_2$.

4.2 Reinforced communication Learning

In our experiments involving RCL, we allow agents to exchange bitstring messages of length 40 at each timestep (*i.e.*, we represent the communication policy of each agent as a set of independent bitwise probabilities, each representing the probability that the corresponding message bit is a 1).

4.3 Task 1: Hidden-Goal Path-Finding

The first example task is a path-finding problem in which each agent must navigate to a randomly-assigned goal cell in a 10x10 grid, and at each timestep is able to move up, down, left, right, or remain stationary. Agents do not observe their own goals or the other agent's location, but instead observe only their location and the other agent's goals; thus, communication is necessary for agents to reliably find their goals. In this task, agents' policy networks are feed-forward neural networks with no persistent state, meaning they cannot remember past messages or observations, and must therefore communicate at every timestep. In the noise-free channel case, we compare a differentiable learning approach in which agents are able to exchange real-valued messages (in the form of 10-element vectors in which each element is bounded between 0 and 1), a reinforced communication learning approach in which messages are bit-

strings, and our proposed differentiable learning approach (described in Sect. 3.1) in which messages are composed of 10 discrete elements, each taking one of 16 possible discrete values. In this case, messages sent by each agent are guaranteed to be received by the other agent at the next timestep, completely unaltered. In the noisy channel case, we corrupt messages with an asymmetric bit-flip noise as described in Sect. 4.5. We compare the reinforced communication learning approach with the noise-tolerant variant of our proposed differentiable approach (described in Sect. 3.2), where messages are converted to a 40-bit binary representation so the same channel noise can be added.

We utilize a shared reward structure, in which agents both receive the same total reward, computed as a sum of both agents' individual rewards. At each timestep, agents are rewarded proportionately to how much closer to their goals they become during that timestep, with an additional penalty for every timestep the agents have not reached their goal, and a penalty for each timestep agents choose not to move while not at their goal location to encourage exploration, similar to (Sartoretti et al. 2018a; 2018b).

Each agent's policy network observes its own location and the other agent's goal in the form of two one-hot 10x10 matrices, which are fed into the convolutional stack of the agent's policy network. Additionally, each agent observes the message generated by the other agent at the last timestep, that is fed into the policy network as a separate input, which is processed by a fully-connected layer before being combined with the output of the convolutional stack. Each agent's critic network observes its location and goal, and the other agent's location in goal, again as one-hot matrices.

4.4 Task 2: Coordinated Multi-Agent Search

The second example task is a two-agent search problem in which each agent is again assigned a goal location in a 10x10 grid. However, in this task, either agent can observe either goal, but only when they are in one of the grid cells immediately adjacent to the goal, or on the goal itself. This limited observability is meant to simulate a sensor with a finite field of view. In this task, to most effectively solve the problem, agents should exchange information related to their observations, in effect broadening each one's sensor footprint. Again, in the noise-free case, we compare differentiable communication learning with real-valued messages, reinforced communication learning with binary messages, and our proposed differentiable approach with discrete messages, and in the case with channel noise, we compare reinforced communication learning with the noise-tolerant variant of our approach described in 3.2. We additionally compare these approaches to a version of the task with no communication among agents, to test the hypothesis that communication improve performance.

The reward function for this task is the same as the reward function used for the hidden-goal path-finding task. The observations to both the actor network and critic network are also largely the same, with the exception that each agent's actor network now receives as input 4 matrices, representing its location, its goal (if visible), the other agent's location, and the other agent's goal (if visible).

4.5 Channel Noise Model

The channel noise model we use for both tasks is an asymmetric bit-flip error model. Bits are flipped with unequal probability depending on their value. The probability of a particular message bit being flipped is taken to be independent of all other bits. More specifically, in the hidden-goal path-finding task, the probability of the ith message bit being flipped is given by $P(\text{flip}_i|\bar{m}_i = 0) = 0.1$ and $P(\text{flip}_i|\bar{m}_i = 1) = 0.05$. In the coordinated multi-agent search task, the bit-flip probability is given by $P(\text{flip}_i|\bar{m}_i = 0) = 0.02$ and $P(\text{flip}_i|\bar{m}_i = 1) = 0.01$.

The noise was chosen to be asymmetric so that it would be message-dependent, and therefore highlight the capability of our approach to deal with channel noise that depends on the message in arbitrary ways.

4.6 Results

In both example tasks, with and without channel noise, we found that when either method was able to solve the task, our proposed discrete differentiable communication learning approach drastically outperformed RCL approaches. In the noise-free hidden-goal path-finding task, our proposed technique was able to solve the task nearly as rapidly as a real-valued differentiable approach that places no limit whatsoever on information transfer rate through the channel. Unsurprisingly, in the path-finding task with channel noise, the performance of our approach was somewhat degraded, requiring more time to converge than in the noise-free case, and attained worse final performance. However, our approach represents a significant improvement over RCL, the only other approach we are aware of that is applicable to such a noisy channel setting. In fact, RCL with noise completely failed to solve the task, likely due to the additional variance being introduced into its policy gradient estimates.

In the coordinated multi-agent search task with no channel noise, our approach outperforms the communication-free variant we tested, indicating that agents are using their communication abilities to their benefit. There again, we find that our approach outperforms RCL, which largely fails to solve the task. The fact that RCL fails to solve the task is not hugely unexpected, as (Foerster et al. 2016) observed similar behavior in some experiments with their reinforced inter-agent learning technique (a form of RCL). Interestingly, the differentiable approach that used real-valued messages seemed to escape a local minimum that our discrete differentiable approach became trapped in. It is possible that the message discretization we used (16 possible values for each message element) was too coarse, and therefore introduced too much noise. Perhaps fewer message elements with a finer discretization would have allowed our approach to perform better on this task while keeping the required information transfer rates the same. We will investigate this trade-off in future work.

In the coordinated multi-agent search task with channel noise, both RCL and our proposed differentiable approach struggle to make progress, and learning appears to be unstable. It is possible that additional tuning of the learning algorithms could result in better performance on this task, which we will determine in future work.

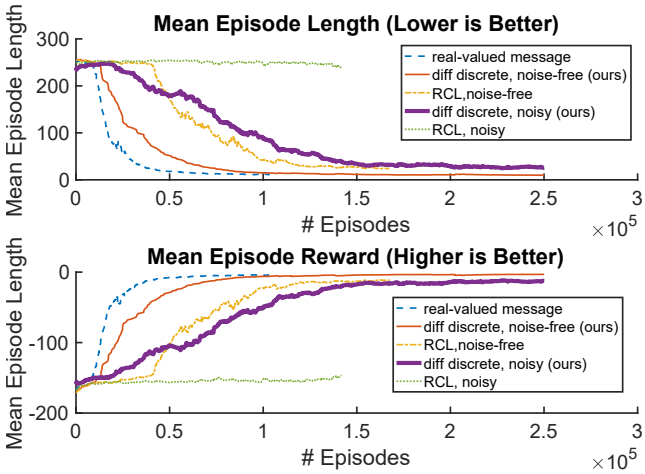

Figure 5: Results for the Hidden-Goal Path-Finding Task.

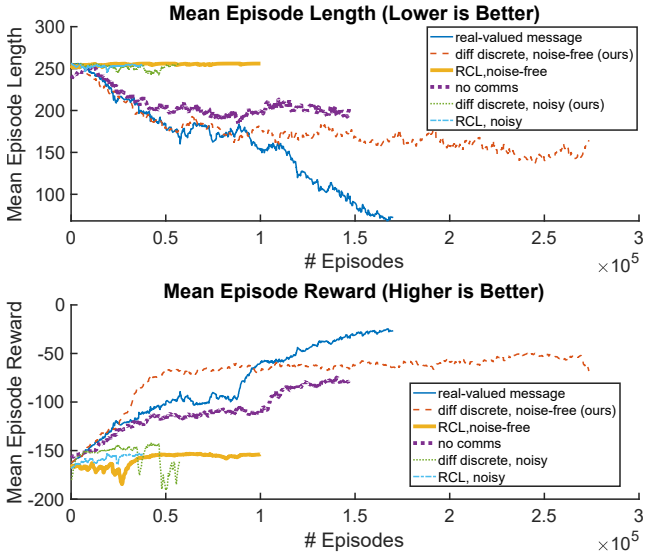

Figure 6: Results for the Coordinated Search Task.

5 Conclusions

In this paper, we presented novel approaches that allow differentiable communication learning to be effectively applied to realistic settings where such approaches were not previously applicable. Specifically, we presented a novel stochastic encoding/decoding procedure, that can be used in conjunction with multi-agent reinforcement learning, to allow unbiased gradient estimates through a discrete channel. This form of communication learning is made possible by the fact that, with our proposed technique, the encoder/discrete channel/decoder system becomes mathematically equivalent to an analog channel with additive noise, through which derivatives can easily be backpropagated. We additionally extended our approach to allow gradient backpropagation through a channel with unknown noise. To the best of our knowledge, our differentiable communication learning approach is the first one that can handle such unknown noise. We enabled this noise-tolerance by introducing an encryption step immediately before the message is sent through the discrete channel, which forces apparent channel noise to be independent of the communication signal being sent, meaning it does not contribute to the gradient of the channel and can therefore be ignored in gradient calculations.

We evaluated the effectiveness of our proposed techniques in two example tasks, with and without channel noise, compared to a reinforced communication learning approach, and a differentiable approach using real-valued messages (where applicable). We found that our approach outperforms the RCL approaches we tested in all cases in which either algorithm can solve the task. Additionally, we found that in one of the two example tasks, our differentiable approach, which uses a very coarse message discretization and a fairly tight limit on channel capacity, performed nearly as well as the real-valued differentiable approach we tested, which assumes no limit on channel capacity. Our experimental results, along with the theoretical justification of our technique, indicate that our approach offers significant advantages over existing approaches to multi-agent reinforcement learning with communication, especially in cases with realistic discrete and noisy communication channels.

Ongoing work focuses on three natural extensions of our presented approaches. First, we are applying our approach to more complex and realistic problems, such as tasks involving many agents in environments with obstacles and other impediments. We are also investigating the use of alternative learning modalities, such as imitation learning, in our approach. Because agents learn to exchange messages that optimize some performance metric that depends on messages only indirectly through the agent policies, expert demonstrations can be used to train policies without explicit communication demonstration. Therefore, it is likely possible that our approach could learn emergent communication behavior solely through imitation of a centralized expert that does not rely on communication, similar to (Paulos et al. 2019).

Second, we are also currently investigating how more established encryption techniques, such as RSA (Rivest, Shamir, and Adleman 1978), could be applied to our approach. For instance, while the type of encryption we use is information-theoretically secure and therefore the environment is guaranteed to have no knowledge of the message we are communicating, modern encryption techniques rely instead on requiring third parties to do an enormous amount of computation to decrypt the message. It may be sufficient to use such non-information-theoretically secure but more efficient encryption techniques to make channel noise *nearly* perfectly independent of the communication signal.

Finally, additional ongoing work explores how error correcting codes could be utilized to decrease the amount of noise introduced into agents' communication inputs. Some error-correction techniques, such as a cyclic redundancy check, allow one to detect when an error has occurred in message transmission. Because a large source of noise in our technique results form cases in which the channel outputs an erroneous message, being able to detect such cases with some probability, and neglect the erroneous messages, could lead to an improvement in performance in situations with channel noise.

References

Bernstein, D. S.; Givan, R.; Immerman, N.; and Zilberstein, S. 2002. The complexity of decentralized control of markov decision processes. *Mathematics of operations research* 27(4):819–840.

Busoniu, L.; Babuška, R.; and De Schutter, B. 2010. Multi-agent reinforcement learning: An overview. *Innovations in Multi-Agent Systems and Applications-1* 310:183–221.

Foerster, J. N.; Assael, Y. M.; de Freitas, N.; and Whiteson, S. 2016. Learning to communicate with deep multi-agent reinforcement learning. *CoRR* abs/1605.06676.

Foerster, J. N.; Farquhar, G.; Afouras, T.; Nardelli, N.; and Whiteson, S. 2017. Counterfactual multi-agent policy gradients. *CoRR* abs/1705.08926.

Gupta, J. K.; Egorov, M.; and Kochenderfer, M. J. 2017. Cooperative multi-agent control using deep reinforcement learning. In *AAMAS Workshops*.

Jaderberg, M.; Czarnecki, W. M.; Dunning, I.; Marris, L.; Lever, G.; Castañeda, A. G.; Beattie, C.; Rabinowitz, N. C.; Morcos, A. S.; Ruderman, A.; et al. 2019. Human-level performance in 3d multiplayer games with population-based reinforcement learning. *Science* 364(6443):859–865.

Jang, E.; Gu, S.; and Poole, B. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Kraemer, L., and Banerjee, B. 2016. Multi-agent reinforcement learning as a rehearsal for decentralized planning. *Neurocomputing* 190:82–94.

Lowe, R.; Wu, Y.; Tamar, A.; Harb, J.; Abbeel, P.; and Mordatch, I. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. *CoRR* abs/1706.02275.

Mordatch, I., and Abbeel, P. 2017. Emergence of grounded compositional language in multi-agent populations. *CoRR* abs/1703.04908.

OpenAI. 2018. Openai five.

Paulos, J.; Chen, S. W.; Shishika, D.; and Kumar, V. S. A. 2019. Decentralization of multiagent policies by learning what to communicate. *2019 International Conference on Robotics and Automation (ICRA)* 7990–7996.

Rivest, R. L.; Shamir, A.; and Adleman, L. 1978. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM* 21(2):120–126.

Sartoretti, G.; Kerr, J.; Shi, Y.; Wagner, G.; Kumar, T. K. S.; Koenig, S.; and Choset, H. 2018a. PRIMAL: pathfinding via reinforcement and imitation multi-agent learning. *CoRR* abs/1809.03531.

Sartoretti, G.; Wu, Y.; Paivine, W.; Kumar, T. K. S.; Koenig, S.; and Choset, H. 2018b. Distributed reinforcement learning for multi-robot decentralized collective construction. In *DARS 2018 - International Symposium on Distributed Autonomous Robotic Systems*, 35–49.

Sartoretti, G.; Paivine, W.; Shi, Y.; Wu, Y.; and Choset, H. 2019. Distributed learning of decentralized control policies for articulated mobile robots. *CoRR* abs/1901.08537.

Sukhbaatar, S.; Szlam, A.; and Fergus, R. 2016. Learning multiagent communication with backpropagation. *CoRR* abs/1605.07736.