A Joint Learning and Communication Framework for Multi-Agent Reinforcement Learning over Noisy Channels

Tze-Yang Tung, Joan Roig Pujol, Szymon Kobus, Deniz Gündüz
Information Processing and Communications Laboratory (IPC-Lab)
Dept. of Electrical and Electronic Engineering, Imperial College London, UK

Abstract

We propose a novel formulation of the “effectiveness problem” in communications, put forth by Shannon and Weaver in their seminal work [2], by considering multiple agents communicating over a noisy channel in order to achieve better coordination and cooperation in a multi-agent reinforcement learning (MARL) framework. Specifically, we consider a multi-agent partially observable Markov decision process (MA-POMDP), in which the agents, in addition to interacting with the environment can also communicate with each other over a noisy communication channel. The noisy communication channel is considered explicitly as part of the dynamics of the environment and the message each agent sends is part of the action that the agent can take. As a result, the agents learn not only to collaborate with each other but also to communicate “effectively” over a noisy channel. This framework generalizes both the traditional communication problem, where the main goal is to convey a message reliably over a noisy channel, and the “learning to communicate” framework that has received recent attention in the MARL literature, where the underlying communication channels are assumed to be error-free. We show via examples that the joint policy learned using the proposed framework is superior to that where the communication is considered separately from the underlying MA-POMDP. This is a very powerful framework, which has many real world applications, from autonomous vehicle planning to drone swarm control, and opens up the rich toolbox of deep reinforcement learning for the design of multi-user communication systems.

This work was supported in part by the European Research Council (ERC) Starting Grant BEACON (grant agreement no. 677854) and by the UK EPSRC (grant no. EP/T023600/1).

An earlier version of this work was presented at the IEEE Global Communications Conference (GLOBECOM) in December 2020 [1].

I. Introduction

Communication is essential for our society. Humans use language to communicate ideas, which has given rise to complex social structures, and scientists have observed either gestural or vocal communication in other animal groups, complexity of which increases with the complexity of the social structure of the group [3]. Communication helps achieving complex goals by enabling cooperation and coordination [4], [5]. Advances in our capability to store and transmit information over time and long distances have greatly expanded our capabilities, and allowed us to turn the world into a connected society. Communication technologies are at the core of this massively complex system, and we continuously strive to improve our communication capabilities with faster, more reliable, more energy-efficient and more agile communication systems.

Our communication technologies are built upon fundamental mathematical principles and engineering expertise. The fundamental quest in the design of these systems have been to deal with various imperfections in the communication channel (e.g., noise and fading) and the interference among transmitters. Decades of research and engineering efforts have produced highly advanced networking protocols, modulation techniques, waveforms and coding techniques that can deal with these challenges quite effectively. However, this design approach ignores the aforementioned core objective of communication in enabling coordination and cooperation. To some extent, we have separated the design of technologies that can enable the creation of a communication network that can reliably carry signals from one point to another, and the ‘language’ that is formed to achieve the underlying purpose of communication, which allows agents to communicate their view of the world and their intentions to others to achieve coordination and cooperation.

This engineering approach was also highlighted by Shannon and Weaver in [2] by organizing the communication problem into three “levels”: They described level A as the *technical problem*, which tries to answer the question “How accurately can the symbols of communication be transmitted?”. Level B is referred to as the *semantic problem*, and asks the question “How precisely do the transmitted symbols convey the desired meaning?”. Finally, Level C, called the *effectiveness problem*, strives to answer the question “How effectively does the received meaning affect conduct in the desired way?”. As we have described above, our communication technologies mainly deal with Level A, ignoring the semantics or the effectiveness problems. This simplifies the problem into the transmission of a discrete message or a continuous waveform

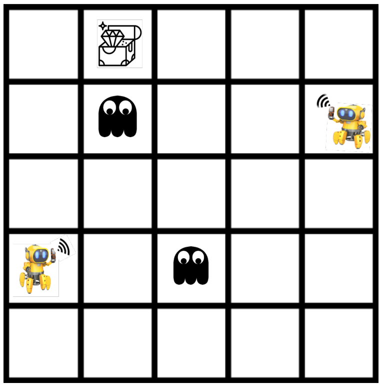


Fig. 1. An illustration of a MARL problem with noisy communication between the agents, e.g., agents communicating over a shared wireless channel. The emerging communication scheme should not only allow the agents to better coordinate and cooperate to maximize their rewards, but also mitigate the adverse effects of the wireless channel, such as noise and interference.

over a communication channel in the most reliable manner. The semantics problem deals with the meaning of the messages, and is rather abstract. There is a growing interest in the semantics problem in the recent literature [6]–[10]; however, these works typically formulate the semantics as an end-to-end joint source-channel coding problem, where the reconstruction objective can be distortion with respect to the original signal [11], [12], or a more general function that can model some form of 'meaning' [6], [13]–[15], which goes beyond reconstructing the original signal[1].

In this paper, we deal with the 'effectiveness problem', which generalizes the problems in both level A and level B. In particular, we formulate a multi-agent problem with noisy communications between the agents, where the goal of communications is to help agents to better cooperate to achieve a common goal. See Fig. 1 for an illustration of a multi-agent grid-world, where agents can communicate through noisy wireless links. It is well-known that in tasks where multiple agents need to collaborate towards achieving a common goal, communication can greatly improve the efficiency of cooperation [5], [16]. Recently, there has been significant interest in the *emergence of communication* among agents within the reinforcement learning (RL) literature [17]–[20]. These works consider multi-agent RL (MARL) problems, in which agents have access to a dedicated communication channel, and the objective is to learn a communication protocol, which can be considered as a 'language' to achieve the underlying goal, which is typically

[1] To be more precise, remote hypothesis testing, classification, or retrieval problems can also be formulated as end-to-end joint source-channel coding problems, albeit with a non-additive distortion measure.

translated into maximizing a specific reward function. This corresponds to Level C as described by Shannon and Weaver in [2] as the agents change their behavior based on the messages received over the channel in order to maximize their reward. However, the focus of the aforementioned works is the emergence of communication protocols within the limited communication resources that can provide the desired impact on the behavior of the agents; and, unlike Shannon and Weaver, these works ignore the physical layer characteristics of the channel.

Our goal in this work is to treat the effectiveness problem taking into account both the channel noise and the end-to-end learning objective. In this problem, the goal of communication is not "reproducing at one point either exactly or approximately a message selected at another point" as stated by Shannon in [2], and which laid the foundations of the communication and information theoretic formulations we have studied over the last seven decades. Instead, the goal is to enable cooperation in order to improve the objective of the underlying multi-agent game. As we will show later in this paper, the codes that emerge from the proposed framework can be very different from those that would be used for reliable communication of messages.

We formulate this novel communication problem as a MARL problem, in which the agents have access to a noisy communication channel. More specifically, we formulate this as a multi-agent partially observable Markov decision process (POMDP), and come up with RL algorithms that can learn policies that govern both the actions of the agents in the environment and the signals they transmit over the channel. A communication protocol in this scenario should enable cooperation and coordination among agents in the presence of channel noise. Therefore, the emerging modulation and coding schemes must not only be capable of error correction/ compensation, but also enable agents to share their knowledge of the environment and/or their intentions. We believe that this novel formulation opens up many new directions for the design of communication protocols and codes that will be applicable in many multi-agent scenarios from teams of robots to platoons of autonomous cars [21], to drone swarm planning [22].

We summarize the main contributions of this work as follows:

1) We propose a novel formulation of the "effectiveness problem" in communications, where agents communicate over a noisy communication channel in order to achieve better co-ordination and cooperation in a MARL framework. This can be interpreted as a *joint communication and learning approach* in the RL context [15]. The current paper is an initial study of this general framework, focusing on scenarios that involve only point-to-point communications for simplicity. More involved multi-user communication and

coordination problems will be the subject of future studies.

2) The proposed formulation generalizes the recently studied “learning to communicate” framework in the MARL literature [17]–[20], where the underlying communication channels are assumed to be error-free. This framework has been used to argue about the emergence of natural languages [23], [24]; however, in practice, there is inherent noise in any communication medium, particularly in human/animal communications. Indeed, languages have evolved to deal with such noise. For example, Shannon estimated that the English language has approximately 75% redundancy. Such redundancy provides error correction capabilities. Hence, we argue that the proposed framework better models realistic communication problems, and the emerging codes and communication schemes can help better understand the underlying structure of natural languages.
3) The proposed framework also generalizes communication problems at level A, which have been the target of most communication protocols and codes that have been developed in the literature. Channel coding, source coding, as well as joint source-channel coding problems, and their multi-user extensions can be obtained as special cases of the proposed framework. The proposed DRL framework provides alternative approaches to the design of codes and communication schemes for these problems that can go beyond the existing ones. We highlight that there are very limited practical code designs in the literature for most multi-user communication problems, and the proposed framework and the exploitation of deep representations and gradient-based optimization in DRL can provide a scalable and systematic methodology to make progress in these challenging problems.
4) We study a particular case of the proposed general framework as an example, which reduces to a point-to-point communication problem. In particular, we show that any single-agent Markov decision process (MDP) can be converted into a multi-agent partially observable MDP (MA-POMDP) with a noisy communication link between the two agents. We consider both the binary symmetric channel (BSC) and the additive white Gaussian noise (AWGN) channel for the noisy communication link and solve the MA-POMDP problem by treating the other agent as part of the environment, from the perspective of one agent. We employ deep Q-learning (DQN) [25] and deep deterministic policy gradient (DDPG) [26] to train the agents. Substantial performance improvement is observed in the resultant policy over those learned by considering the cooperation and communication problems separately.
5) We then present the joint modulation and channel coding problem as an important special

case of the proposed framework. In recent years, there has been a growing interest in using machine learning techniques to design practical channel coding and modulation schemes [11], [27]–[31]. However, with the exception of [31], most of these approaches assume that the channel model is known and differentiable, and they use a supervised training by directly backpropagating through the channel using the known and differentiable channel model. Instead, in this paper, we learn to communicate over an unknown channel solely based on the reward function by formulating it as a RL problem. The proposed DRL framework goes beyond the method employed in [31], which treats the channel as a random variable, and numerically approximates the gradient of the loss function. It is shown through numerical examples that the proposed DRL techniques employing DDPG and actor-critic [32] algorithms significantly improve the block error probability (BLER) of the resultant code.

II. Related Works

The study of communication for multi-agent systems is not new [33]. However, due to the success of deep neural networks (DNNs) for reinforcement learning (RL), this problem has received renewed interest in the context of DNNs [23] and deep RL (DRL) [17], [34], [35], where partially observable multi-agent problems are considered. In each case, the agents, in addition to taking actions that impact the environment, can also also communicate with each other via a limited-capacity communication channel. Particularly, in [17], two approaches are considered: reinforced inter-agent learning (RIAL), where two centralized Q-learning networks learn to act and communicate, respectively, and differentiable inter-agent learning (DIAL), where communication feedback is provided via backpropagation of gradients through the channel. However, communication between agents is restricted only during execution.

Similarly, in [36], [37], the authors propose a *centralized learning, decentralized execution* approach, where a central critic is used to learn the state-action values of all agents and use those values to train individual policies of each agent. Although they also consider the transmitted messages as part of the agents' actions, the communication channel is assumed to be noiseless.

CommNet [34] attempts to leverage communications in cooperative MARL by using multiple continuous-valued transmissions at each time step to make decisions for all agents. Each agent broadcasts its message to every other agent, and the averaged message received by each agent forms part of the input. However, this solution lacks scalability as it depends on a centralized network by treating the problem as a single RL problem. Similarly, BiCNet [38] utilizes recurrent neural networks to connect individual agent's policy with a centralized controller aggregating the hidden states of each agent, acting as communication messages.

The reliance of the aforementioned works on a broadcast channel to communicate with all agents simultaneously may be infeasible or highly inefficient in practice. To overcome this limitation, in [18], the authors propose an attentional communication model that learns when communication is needed and how to integrate shared information for cooperative decision making. In [20], directional communication between agents is achieved with a signature-based soft attention mechanism, where each message is associated to the target recipient. They also propose multi-stage communication, where multiple rounds of communication take place before an action is taken.

It is important to note that all of the prior works discussed above rely on error-free communication channels. MARL over noisy communication channels is considered in [39], where two agents placed on a grid world aim to coordinate to step on the goal square simultaneously. However, the problem presented in [39] does not actually require any communication to accomplish the desired task. In fact, it can be shown that even if the agents are trained independently without any communication, the total discounted reward would still be higher than the average reward achieved by the solution proposed in [39].

To the best of our knowledge, this is the first work that presents a MARL with communications framework that can be used to solve various MARL with noisy communication problems. We highlight the importance of joint learning and communication over noisy channels by demonstrating the superior performance of the learned policy over those that separate communication from learning. We will in subsequent sections present a framework for solving MARL with communication problems and demonstrate its use with two example problems.

III. Problem Formulation

The problem we consider herein is a multi-agent partially observable Markov decision process (MA-POMDP) with noisy communications. Consider first a Markov game $(\mathcal{S}, \{\mathcal{O}_i\}_{i=1}^N, \{\mathcal{A}_i\}_{i=1}^N, P, r)$, where $\mathcal{S}$ represents all possible configurations of the environment and agents, $\mathcal{O}_i$ and $\mathcal{A}_i$ are the observation and action sets of agent i, respectively, P is the transition kernel that governs the environment, r is the reward function, and there are in total N agents. The agents' observations $\{\mathcal{O}_i\}_{i=1}^N$ are partial observations of the true state of the environment $\mathcal{S}$. At each step

t of this Markov game, agent i observes the partial state $o_i^{(t)} \in \mathcal{O}_i$, and takes action $a_i^{(t)} in \mathcal{A}_i$. Then, the state of the MA-POMDP transitions from $s^{(t)}$ to $s^{(t+1)}$ according to the joint actions of the agents following the transition probability $P(s^{(t+1)}|s^{(t)}, \mathbf{a}^{(t)})$, where $\mathbf{a}^{(t)} = (a_1^{(t))}, \dots, a_N^{(t))})$. The observations in the next time instant follow the conditional distribution $\Pr(o^{(t+1)}|s^{(t)}, \mathbf{a}^{(t)})$. While, in general, each agent can have a separate reward function, we consider herein the fully cooperative setting, where the agents receive the same team reward $r^{(t)} = r(s^{(t)}, \mathbf{a}^{(t)})$ at time t.

In order to coordinate and maximize the total reward, the agents are endowed with a noisy communication channel, which is orthogonal to the environment; that is, the environment transitions depend only on the environment actions, and the only impact of the communication channel is that the actions of the agents can now depend on the past received messages as well as the past observations and rewards. We assume that the communication channel is governed by the conditional probability distribution P_c, and we allow the agents to use the channel M times at each time t. Here, M can be considered as the *channel bandwidth*. Let the signals transmitted and received by agent i at time step t be denoted by $m_i^{(t)} \in \mathcal{C}_t^M$ and $\hat{m}_i^{(t)} \in \mathcal{C}_r^M$, respectively, where $\mathcal{C}_t$ and $\mathcal{C}_r$ denote the input and output alphabets of the channel, which can be discrete or continuous. We assume for simplicity that the input and output alphabets of the channel are the same for all the agents. Channel inputs and outputs at time t are related through the conditional distribution $P_c\big(\hat{\mathbf{m}}^{(t)}|\mathbf{m}^{(t)}\big) = \Pr\big(\hat{\mathbf{m}} = \{\hat{m}_i^{(t)}\}_{i=1}^N|\mathbf{m} = \{m_i^{(t)}\}_{i=1}^N\big)$, where $\hat{\mathbf{m}} = (\hat{m}_1, \dots, \hat{m}_M)$ is the vector of received signals and $\mathbf{m} = (m_1, \dots, m_M)$ is the vector of transmitted signals. That is, the received signal of agent i over the communication channel is a random function of the signals transmitted by all other agents, characterized by the conditional distribution of the multi-user communication channel.

We can now define a new Markov game with noisy communications, where the actions of agent i now consist of two components, the environment actions $a_i^{(t)}$ as before, and the signal to be transmitted over the channel $m_i^{(t)}$. Each agent, in addition to taking actions that affect the state of the environment, can also send signals to other agents over M uses of the noisy communication channel. The observation of each agent is now given by $(o_i^{(t)}, \hat{m}_i^{(t)})$; that is, a combination of the partial observation of the environment as before and the channel output signal.

At each time step t, agent i observes $(o_i^{(t)}, \hat{m}_i^{(t)})$ and selects an action $(a_i^{(t)}, m_i^{(t)})$ according to its policy $\pi_i : \mathcal{O}_i \times \mathcal{C}_r^M \to \mathcal{A}_i \times \mathcal{C}_t^M$. The overall policy over all agents can be defined as $\Pi : \mathcal{S} \to \mathcal{A}$. The objective of the Markov game with noisy communications is to maximize the

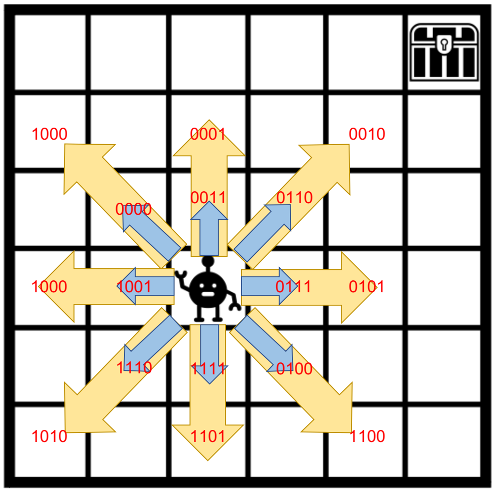


Fig. 2. Illustration of the guided robot problem in grid world. The set $\mathcal{A}_2$ of 16 possible actions the scout agent can take using hand crafted codewords.

discounted sum of rewards

$$V_\Pi(s) = \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} r^{(t)} \middle| s^{(1)} = s\right] \tag{1}$$

for any initial state $s \in \mathcal{S}$ and γ is the discount factor to ensure convergence. We also define the state-action value function, also referred to as Q-function as

$$Q_\Pi(s^{(t)}, a^{(t)}) = \mathbb{E}_\Pi\left[\sum_{i=t}^{\infty} \gamma^{(i-t)} r^{(t)} \middle| s^{(t)}, a^{(t)}\right] \tag{2}$$

In the subsequent sections we will show that this formulation of the MA-POMDP with noisy communications lends itself to multiple problem domains where communication is vital to achieve non-trivial total reward values, and we devise methods that jointly learn to collaborate and communicate despite the noise in the channel. Although the introduced MA-POMDP framework with communications is fairly general and can model any multi-agent scenario with complex multi-user communications, our focus in this paper will be on point-to-point communications. This will allow us to expose the benefits of the joint communication and learning design, without having to deal with the challenges of multi-user communications. Extensions of the proposed framework to scenarios that would involve multi-user communication channels will be studied in future work.

IV. Guided Robot with Point-to-Point Communications

In this section, we consider a single-agent MDP and turn it into a MA-POMDP problem by dividing the single agent into two separate agents, a *guide* and a *scout*, which are connected through a noisy communication channel. In this formulation, we assume that the guide observes the state of the original MDP perfectly, but cannot take actions on the environment directly. Contrarily, the scout can take actions on the environment, but cannot observe the environment state. Therefore, the guide communicates to the scout through a noisy communication channel and the scout has to take actions based on the signals it receives from the guide through the communication channel. The scout can be considered as a robot remotely controlled by the guide agent which has sensors to observe the environment.

We consider this particular setting since it clearly exposes the importance of communication as the scout depends solely on the signals received from the guide. Without the communication channel, the scout is limited to purely random actions independent of the current state. Moreover, this scenario also allows us to quantify the impact of the channel noise on the overall performance since we recover the original single-agent MDP when the communication channel is perfect; that is, if any desired message can be conveyed over the channel in a reliable manner. Therefore, if the optimal reward for the original MDP can be determined, this would serve as an upper bound on the reward of the MA-POMDP with noisy communications.

As an example to study the proposed framework and to develop and test numerical algorithms aiming to solve the obtained MA-POMDP problem, we consider a grid world of size $L \times L$, denoted by $\mathcal{L} = [L] \times [L]$, where $[L] = \{0, 1, \ldots, L-1\}$. We denote the scout position at time step t by $p_s^{(t)} = (x_s^{(t)}, y_s^{(t)}) \in \mathcal{L}$. At each time instant, the scout can take one action from the set of 16 possible actions $\mathcal{A} = \{[1,0], [-1,0], [0,1], [0,-1], [1,1], [-1,1], [-1,-1], [1,-1]$ $, [2,0], [-2,0], [0,2], [0,-2], [2,2], [-2,2], [-2,-2], [2,-2]\}$. See Fig. 6 for an illustration of the scout and the 16 actions it can take. If the action taken by the scout ends up in a cell outside of the grid world, the agent remains in its original location. The transition probability kernel of this MDP is specified as follows: after each action, the agent moves to the intended target location w.p. $1-\delta$, and to a random neighboring cell w.p. δ. That is, the next state is given by $s^{(t+1)} = s^{(t)} + a^{(t)}$ w.p. $1-\delta$, and $s^{(t+1)} = s^{(t)} + a^{(t)} + z^{(t)}$, where $z^{(t)}$ is uniformly distributed over the set $\{[1,0], [1,1], [0,1], [-1,1], [-1,0], [0,-1], [-1,-1], [1,-1]\}$, w.p. δ.

The objective of the scout is to find the treasure, located at $p_g = (x_g, y_g) \in \mathcal{L}$ as quickly as possible. We assume that the initial position of the scout and the location of the treasure are random, and are not the same. The scout takes instructions from the guide, who observes the grid world, and utilizes a noisy communication channel M times to transmit signal $m^{(t)}$ to the scout, who observes $\hat{m}^{(t)}$ from the output of the channel. To put it in the context of the MA-POMDP defined in Section III, agent 1 is the guide, with observable state $o_1^{(t)} = s^{(t)}$, where $s^{(t)} = (p_s^{(t)}, p_g)$, and action set $\mathcal{A}_1 = \mathcal{C}_t$. Agent 2 is the scout, with observation $o_2^{(t)} = \hat{m}^{(t)}$ and action set $\mathcal{A}_2 = \mathcal{A}$ (or, more precisely, $o_1^{(t)} = (s^{(t)}, \varnothing), o_2^{(t)} = (\varnothing, \hat{m}_2^{(t)})$). We define the reward function as follows to encourage the agents to collaborate to find the treasure as quickly as possible:

$$r^{(t)} = \begin{cases} 10, & \text{if } p_s^{(t)} = p_g \\ -1, & \text{else} \end{cases} \tag{3}$$

We should highlight that despite the simplicity of the problem, the original MDP is not a trivial one when both the initial state of the agent and the target location are random, as it has a rather large state space, and learning the optimal policy requires a long training process in order to observe all possible agent and target location pairs sufficiently many times. In order to simplify the learning of the optimal policy, and focus on learning the communication scheme, we will pay special attention to the scenario where $\delta = 0$. This corresponds to the scenario in which the underlying MDP is deterministic, and it is not difficult to see that the optimal solution to this MDP is rather trivial; the agent should simply take the shortest path to the treasure.

We consider two types of channel distributions: a binary symmetric channel (BSC) and an additive white Gaussian noise channel (AWGN). In the BSC case, we have $\mathcal{C}_t = \{-1, +1\}$. For the AWGN channel, we can have $\mathcal{C}_t = \{-1, +1\}$ if the input is constrained to binary phase shift keying (BPSK) modulation, whereas we have $\mathcal{C}_t = \mathbb{R}$ if no limitation is imposed on the input constellation. We will impose an average power constraint in the latter case. In both cases, the output alphabet is $\mathcal{C}_r = \mathbb{R}$. For the BSC, the output of the channel is given by $\hat{m}_i^{(t)} = m_i^{(t)} \oplus n^{(t)}$, where $n^{(t)} \sim \mathrm{Bernoulli(p_e)}$. For the binary input AWGN channel, the output at the ith use of the channel is given by $\hat{m}_i^{(t)} = m_i^{(t)} + n^{(t)}$, where $n^{(t)} \sim \mathcal{N}(0, \sigma_n^2)$ is the zero-mean Gaussian noise term with variance σ_n^2. In both cases, we assume that the channel is memoryless across different uses.

We first consider the BSC case, which was also considered in [1]. The action set of agent 1 is $\mathcal{A}_1 = \{-1, +1\}^M$, while the observation set of agent 2 is $\mathcal{O}_2 = \{-1, +1\}^M$. We will employ deep

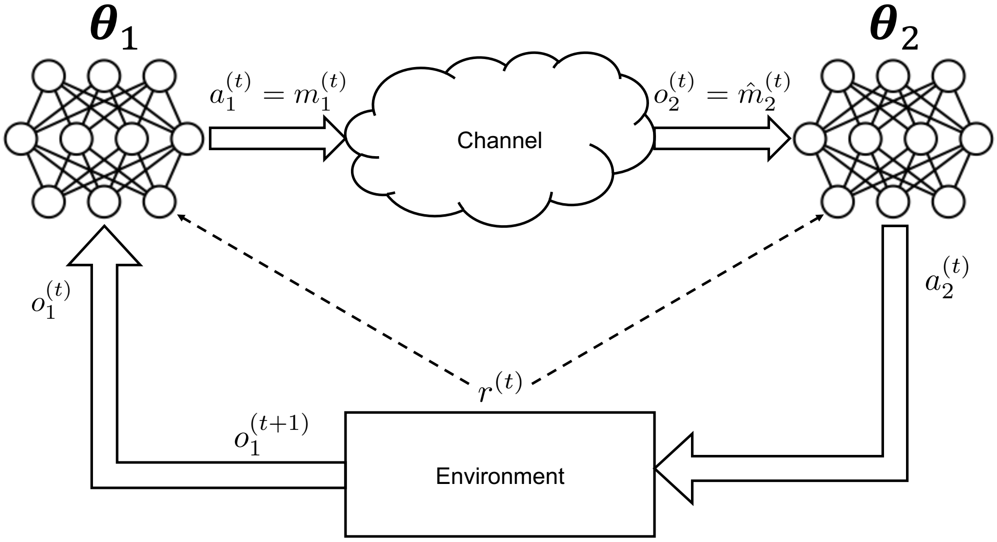


Fig. 3. Information flow between the guide and the scout.

Q-learning network, introduced in [25], which uses deep neural networks (DNNs) to approximate the Q-function in Eqn. (2). More specifically, we use two distinct DNNs, parameterized by $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$, respectively, representing DNNs for approximating the Q-functions of agent 1 (guide) and agent 2 (scout).

The guide observes $o_1^{(t)} = (p_s^{(t)}, p_g)$ and chooses a channel input signal $m_1^{(t)} = a_1^{(t)} = \arg\max_a Q_{\boldsymbol{\theta}_1}(o_1^{(t)}, a) \in \mathcal{A}_1$, based on the current Q-function approximation. The signal is then transmitted across M uses of the BSC. The scout observes $o_2^{(t)} = \hat{m}_2^{(t)}$ at the output of the BSC, and chooses an action based on the current Q-function approximation $a_2^{(t)} = \arg\max_a Q_{\boldsymbol{\theta}_2}(o_2^{(t)}, a) \in \mathcal{A}_2$. The scout then takes the action $a_2^{(t)}$, which updates its position $p_s^{(t+1)}$, collects reward $r^{(t)}$, and the process is repeated. The reward $r^{(t)}$ is fed to both the guide and the scout to update $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$.

As is typical in Q-learning methods, we use *replay buffer*, *target networks* and *ϵ-greedy* to improve the learned policy. The replay buffers $\mathcal{R}_1$ and $\mathcal{R}_2$ store experiences $(o_1^{(t)}, a_1^{(t)}, r^{(t)}, o_1^{(t+1)})$ and $(o_2^{(t)}, a_2^{(t)}, r^{(t)}, o_2^{(t+1)})$ for the guide and scout, respectively, and we sample them uniformly to update the parameters $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$. This prevents the states from being correlated, which would break the assumption in most optimization algorithms that the samples are independent. We use target parameters $\boldsymbol{\theta}_1^-$ and $\boldsymbol{\theta}_2^-$, which are copies of $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$, to compute the DQN loss function:

$$L_{\text{DQN}}(\boldsymbol{\theta}_i) = \frac{1}{2}\Big(r^{(t)} + \gamma \max_a \big\{Q_{\boldsymbol{\theta}_i^-}\big(o_i^{(t+1)}, a\big)\big\} - Q_{\boldsymbol{\theta}_i}\big(o_i^{(t)}, a_i^{(t)}\big)\Big)^2, \; i = 1, 2. \tag{4}$$

The parameters $\boldsymbol{\theta}_i$ are then updated via gradient descent according to the gradient $\nabla_{\boldsymbol{\theta}_i} L_{\text{DQN}}(\boldsymbol{\theta}_i)$, and the target network parameters are updated via

$$\boldsymbol{\theta}_i^- \leftarrow \tau\boldsymbol{\theta}_i + (1-\tau)\boldsymbol{\theta}_i^-, \quad i = 1, 2, \tag{5}$$

where $0 \leq \tau \leq 1$. The target networks here stabilize the updates. Due to Q-learning being bootstrapped, if the same $Q_{\boldsymbol{\theta}_i}$ is used to estimate the state-action value of time step t and $t+1$, both values would move at the same time, which may lead to the updates to never converge (like a dog chasing its tail). By introducing the target networks, this effect is reduced due to the much slower updates of the target network, as done in Eqn. (5).

To promote exploration, we use ϵ-greedy, which chooses a random action w.p. ϵ at each time step. That is,

$$a_i^{(t)} = \begin{cases} \arg\max_a Q_{\boldsymbol{\theta}_i}(o_i^{(t)}, a), & \text{w.p. } 1-\epsilon \\ a \sim \text{Uniform}(\mathcal{A}_i), & \text{w.p. } \epsilon, \end{cases} \tag{6}$$

where $a \sim \text{Uniform}(\mathcal{A}_i)$ denotes an action that is sampled uniformly from the action set $\mathcal{A}_i$. The proposed solution for the BSC case is shown in Algorithm 1. We find that the joint learning and communication policies are significantly better than those of separate learning and communication. The numerical results for this example will be discussed in detail in Section VI.

For the binary input AWGN channel, we can use the exact same solution as the one used for BSC. Note that the observation set of the scout is $\mathcal{O}_2 = \mathbb{R}^M$. However, the more interesting case is when $\mathcal{A}_1 \in \mathbb{R}^M$. It has been observed in the JSCC literature [11], [40], that relaxing the constellation constraints, similarly to analog communications, and training the JSCC scheme in an end-to-end fashion can provide significant performance improvements thanks to the greater degree of freedom available to the transmitter. In this case, since the guide can output continuous actions, we can employ the deep deterministic policy gradient (DDPG) algorithm proposed in [26]. DDPG uses a parameterized policy function $\mu_{\boldsymbol{\psi}}(o_1^{(t)})$, which specifies the current policy by deterministically mapping the state $o_1^{(t)}$ to a continuous action. The critic $Q_{\boldsymbol{\theta}_1}(o_1^{(t)}, \mu_{\boldsymbol{\psi}}(o_1^{(t)}))$, then estimates the value of the action taken by $\mu_{\boldsymbol{\psi}}(o_1^{(t)})$, and is updated as it was done with DQN in Eqn. (4).

The guide policy is updated by applying the chain rule to the expected return from the initial distribution

$$J = \mathbb{E}_{o_1^{(t)} \sim \rho^{\pi_1}, o_2^{(t)} \sim \rho^{\pi_2}, a_1^{(t)} \sim \pi_1, a_2^{(t)} \sim \pi_2} \left[\sum_{t=1}^{\infty} \gamma^{t-1} r^{(t)}(o_1^{(t)}, o_2^{(t)}, a_1^{(t)}, a_2^{(t)}) \right], \tag{7}$$

Algorithm 1: Proposed solution for the guided robot problem with BSC.

```
Initialize Q networks, $\boldsymbol{\theta}_i, i = 1, 2$, using Gaussian initialization $\mathcal{N}(0, 10^{-2})$.
Copy parameters to target networks $\boldsymbol{\theta}_i^- \leftarrow \boldsymbol{\theta}_i$.
$episode = 0$
while $episode < episode\text{-}max$ do
    $episode = episode + 1$
    $t = 0$
    $\epsilon = \epsilon_{\text{end}} + (\epsilon_0 - \epsilon_{\text{end}}) e^{\left(\frac{\text{episode}}{-\lambda}\right)}$
    while Treasure NOT found OR $t < t_{max}$ do
        $t = t + 1$
        Observe $o_1^{(t)} = (p_s^{(t)}, p_g)$
        $m_1^{(t)} = a_1^{(t)}$
        $= \begin{cases} \arg\max_a Q_{\boldsymbol{\theta}_1}(o_1^{(t)}, a), \text{ w.p. } 1 - \epsilon, \\ a \sim \text{Uniform}(\mathcal{A}_1), \text{ w.p. } \epsilon. \end{cases}$
        Observe $o_2^{(t)} = P_{\text{BSC}}(\hat{m}_2^{(t)} | m_1^{(t)})$
        $a_2^{(t)} = \begin{cases} \arg\max_a Q_{\boldsymbol{\theta}_1}(o_2^{(t)}, a), \text{ w.p. } 1 - \epsilon, \\ a \sim \text{Uniform}(\mathcal{A}_2), \text{ w.p. } \epsilon. \end{cases}$
        Take action $a_2^{(t)}$, collect reward $r^{(t)}$
        if $t > 1$ then
            Store experiences:
            $(o_1^{(t-1)}, a_1^{(t-1)}, r^{(t-1)}, o_1^{(t)}) \in \mathcal{R}_1$
            $(o_2^{(t-1)}, a_2^{(t-1)}, r^{(t-1)}, o_2^{(t)}) \in \mathcal{R}_2$
    end
    Get batches $\mathcal{B}_1 \subset \mathcal{R}_1$, $\mathcal{B}_2 \subset \mathcal{R}_2$
    Compute DQN average loss $L_{\text{DQN}}(\boldsymbol{\theta}_i), i = 1, 2$ as in Eqn. (4) using batch $\mathcal{B}_i$
    Update $\boldsymbol{\theta}_i$ using $\nabla_{\boldsymbol{\theta}_i} L_{\text{DQN}}(\boldsymbol{\theta}_i), i = 1, 2$
    Update target networks $\boldsymbol{\theta}_i^-, i = 1, 2$ via Eqn. (5)
end
```

where ρ^{π_i} is the discounted observation visitation distribution for the agent i policy π_1. Since we solve this problem by letting each agent treat the other agent as part of the environment, the value of the action taken by the guide is only dependent on its observation $o_1^{(t)}$ and action $\mu_{\boldsymbol{\psi}}(o_1^{(t)})$. Thus, we use a result in [41] where the gradient of the objective J in Eqn. (7) with respect to the guide policy parameters $\boldsymbol{\psi}$ is shown to be

$$\nabla_{\boldsymbol{\psi}} J = \mathbb{E}_{o_1^{(t)} \sim \rho^{\pi_1}} \left[\nabla_{\boldsymbol{\psi}} Q_{\boldsymbol{\theta}_1}(o, a) \big|_{o = o_1^{(t)}, a = \mu_{\boldsymbol{\psi}}(o_1^{(t)})} \right] \tag{8}$$

$$= \mathbb{E}_{o_1^{(t)} \sim \rho^{\pi_1}} \left[\nabla_a Q_{\boldsymbol{\theta}_1}(o,a)\big|_{o=o_1^{(t)}, a=\mu_{\boldsymbol{\psi}}(o_1^{(t)})} \nabla_{\boldsymbol{\psi}} \mu_{\boldsymbol{\psi}}(o)\big|_{o=o_1^{(t)}} \right] \tag{9}$$

if the conditions in Theorem 1 is satisfied as follows:

Theorem 1: [41] A function approximator $Q_{\boldsymbol{\theta}}(o,a)$ is compatible (i.e. the gradient of the true Q function $Q_{\boldsymbol{\theta}^*}$ is preserved by the function approximator) with a deterministic policy $\mu_{\boldsymbol{\psi}}(o)$, such that $\nabla_{\boldsymbol{\psi}} J(\boldsymbol{\psi}) = \mathbb{E}[\nabla_{\boldsymbol{\psi}} \mu_{\boldsymbol{\psi}}(o) \nabla_a Q_{\boldsymbol{\theta}}(o,a)|_{a=\mu_{\boldsymbol{\psi}}(o)}]$, if

1) $\nabla_a Q_{\boldsymbol{\theta}}(o,a)|_{a=\mu_{\boldsymbol{\psi}}(o)} = \nabla_{\boldsymbol{\psi}} \mu_{\boldsymbol{\psi}}(o)^\top \boldsymbol{\theta}$, and
2) $\boldsymbol{\theta}$ minimizes the mean-squared error, $\mathbb{E}[e(o;\boldsymbol{\theta},\boldsymbol{\psi})^\top e(o;\boldsymbol{\theta},\boldsymbol{\psi})]$, where $e(o;\boldsymbol{\theta},\boldsymbol{\psi}) = \nabla_a \big[Q_{\boldsymbol{\theta}}(o,a)|_{a=\mu_{\boldsymbol{\psi}}(o)} - Q_{\boldsymbol{\theta}^*}(o,a)|_{a=\mu_{\boldsymbol{\psi}}(o)}\big]$, and $\boldsymbol{\theta}^*$ are the parameters that describe the true Q function exactly.

In practice, criterion 2) of Theorem 1 is approximately satisfied via mean-squared error loss and gradient descent, but criterion 1) may not be satisfied. Nevertheless, DDPG works well in practice.

The DDPG loss is two-fold: the critic loss is computed as

$$L^{\text{Critic}}_{\text{DDPG}}(\boldsymbol{\theta}_1) = \Big(r^{(t)} + \gamma\Big\{Q_{\boldsymbol{\theta}_1^-}(o_1^{(t+1)}, \mu_{\boldsymbol{\psi}^-}(o_1^{(t+1)}))\Big\} - Q_{\boldsymbol{\theta}_1}(o_1^{(t)}, \mu_{\boldsymbol{\psi}}(o_1^{(t)}))\Big)^2, \tag{10}$$

whereas the policy loss is computed as

$$L^{\text{Policy}}_{\text{DDPG}}(\psi) = -Q_{\boldsymbol{\theta}_1}(o_1^{(t)}, \mu_{\boldsymbol{\psi}}(o_1^{(t)})). \tag{11}$$

As with the DQN case, we can also use a replay buffer and target network to train the DDPG policy. To promote exploration, we add noise to the actions taken as follows:

$$a_1^{(t)} = \mu_{\boldsymbol{\psi}}(o_1^{(t)}) + w^{(t)}, \tag{12}$$

where $w^{(t)}$ is an Orstein-Uhlenbeck process [42] to generate temporally correlated noise terms. The proposed solution for the AWGN channel case is summarized in Algorithm 2. We find that by relaxing the modulation constraint to $\mathbb{R}^M$, the learned policies of guide and scout are substantially better those achieved in the BPSK case. The numerical results illustrating this conclusion will be discussed in Section VI.

To ensure that the actions taken by the guide meet the power constraint we normalize the channel input to an average power of 1 as follows:

$$a_1^{(t)}[k] \leftarrow \sqrt{M} \frac{a_1^{(t)}[k]}{\sqrt{\left(a_1^{(t)}\right)^\top a_1^{(t)}}}, \; k = 1, \ldots, M. \tag{13}$$

The signal-to-noise ratio (SNR) of the AWGN channel is then defined as

$$\text{SNR} = -\log_{10}(\sigma_n^2) \text{ (dB)}. \tag{14}$$

In Section VI, we will study the effects of both the channel SNR and the channel bandwidth on the performance. Naturally, the capacity of the channel increases with both the SNR and the bandwidth. However, we would like to emphasize that the Shannon capacity is not a relevant metric *per se* for the problem at hand. Indeed, we will observe that the benefits from increasing channel bandwidth and channel SNR saturate beyond some point. Indeed, the performance achieved for the underlying single-agent MDP assuming a perfect communication link from the guide to the scout serves as a bound on the performance with any noisy communication channel.

V. Joint Channel Coding and Modulation

The formulation given in Section III can be readily extended to the aforementioned classic "level A" communication problem of channel coding and modulation. The channel coding is a problem where B bits are communicated over M channel uses, which corresponds to a code rate of $\frac{B}{M}$ bits per channel use. In the context of the Markov game introduced previously, we can consider 2^B states corresponding to each possible message. Agent 2 has 2^B actions, each corresponding to a different reconstruction of the message at agent 1. All the actions transition to the end state. The transmitter observes the state and sends a message by using the channel M times, and the receiver observes a noisy version of the message at the output of the channel and chooses an action. A unit reward is given at each episode if the action taken by agent 2 is equal to the state of the system observed by agent 1, while the reward is zero otherwise. Herein, we consider the scenario with real channel input and output values, and an average power constraint on the transmitted signals at each time t. As such, we can define $\mathcal{O}_1 = \mathcal{A}_2 = \{0,1\}^B$ and $\mathcal{A}_1 = \mathcal{O}_2 = \mathcal{C}_t^M$. We note that maximizing the average reward in this problem is equivalent to designing a channel code with blocklength B and rate $\frac{B}{M}$ with minimum BLER.

There have been many recent studies focusing on the design of channel coding and modulation schemes using machine learning techniques [11], [27]–[31]. Most of these works use supervised learning techniques, assuming a known and differentiable channel model, which allows back-propagating through the channel during training. On the other hand, here we assume that the channel model is not known, and the agents are limited to their observations of the noisy channel output signals, and must learn a communication strategy through trial and error.

Algorithm 2: Proposed solution for guided robot problem for AWGN channel.

```
Initialize Q networks $\boldsymbol{\theta}_i, i = 1, 2$, using Gaussian initialization $\mathcal{N}(0, 10^{-2})$ and policy network $\boldsymbol{\psi}$ if
  $\mathcal{A}_1 \in \mathbb{R}^M$. Copy parameters to target networks $\boldsymbol{\theta}_i^- \leftarrow \boldsymbol{\theta}_i$, $\boldsymbol{\psi}^- \leftarrow \boldsymbol{\psi}$.
$episode = 1$
while $episode < episode\text{-}max$ do
    $t = 1$
    $\epsilon = \epsilon_{\text{end}} + (\epsilon_0 - \epsilon_{\text{end}})e^{\left(\frac{\text{episode}}{-\lambda}\right)}$
    while Treasure NOT found OR $t < t_{max}$ do
        Observe $o_1^{(t)} = (p_s^{(t)}, p_g)$
        if $\mathcal{A}_1 = \{-1, +1\}^M$ then
            $m_1^{(t)} = a_1^{(t)} = \begin{cases} \arg\max_a Q_{\boldsymbol{\theta}_1}(o_1^{(t)}, a), & \text{w.p. } 1 - \epsilon, \\ a \sim \text{Uniform}(\mathcal{A}_1), & \text{w.p. } \epsilon. \end{cases}$
        else if $\mathcal{A}_1 = \mathbb{R}^M$ then
            $m_1^{(t)} = \mu_\psi(o_1^{(t)}) + w^{(t)}$
            Normalize $m_1^{(t)}$ via Eqn. (13)
        Observe $o_2^{(t)} = P_{\text{AWGN}}(\hat{m}_2^{(t)} | m_1^{(t)})$
        $a_2^{(t)} = \begin{cases} \arg\max_a Q_{\boldsymbol{\theta}_1}(o_2^{(t)}, a), & \text{w.p. } 1 - \epsilon, \\ a \sim \text{Uniform}(\mathcal{A}_2), & \text{w.p. } \epsilon. \end{cases}$
        Take action $a_2^{(t)}$, collect reward $r^{(t)}$
        if $t > 1$ then
            Store experiences:
            $(o_1^{(t-1)}, a_1^{(t-1)}, r^{(t-1)}, o_1^{(t)}) \in \mathcal{R}_1$ and $(o_2^{(t-1)}, a_2^{(t-1)}, r^{(t-1)}, o_2^{(t)}) \in \mathcal{R}_2$
        $t = t + 1$
    end
    Compute average scout loss $L_{\text{DQN}}(\boldsymbol{\theta}_2)$ as in Eqn. (4) using batch $\mathcal{B}_2 \subset \mathcal{R}_2$
    Update $\boldsymbol{\theta}_2$ using $\nabla_{\boldsymbol{\theta}_2} L_{\text{DQN}}(\boldsymbol{\theta}_2)$
    if $\mathcal{A}_1 = \{-1, +1\}^M$ then
        Compute DQN average loss $L_{\text{DQN}}(\boldsymbol{\theta}_1)$ as in Eqn. (4) using batch $\mathcal{B}_1 \subset \mathcal{R}_1$
        Update $\boldsymbol{\theta}_1$ using $\nabla_{\boldsymbol{\theta}_1} L_{\text{DQN}}(\boldsymbol{\theta}_1)$
        Update target network $\boldsymbol{\theta}_i^-, i = 1, 2$ via Eqn. (5)
    else if $\mathcal{A}_1 = \mathbb{R}^M$ then
        Compute average DDPG Critic loss $L_{\text{DDPG}}^{\text{Critic}}(\boldsymbol{\theta}_1)$ as in Eqn. (10) using batch $\mathcal{B}_1$
        Compute average DDPG Policy loss $L_{\text{DDPG}}^{\text{Policy}}(\boldsymbol{\psi})$ as in Eqn. (11) using batch $\mathcal{B}_1$
        Update $\boldsymbol{\theta}_1$ and $\boldsymbol{\psi}$ using $\nabla_{\boldsymbol{\theta}_1} L_{\text{DDPG}}^{\text{Critic}}(\boldsymbol{\theta}_1)$ and $\nabla_\psi L_{\text{DDPG}}^{\text{Policy}}(\boldsymbol{\psi})$
        Update target network $\boldsymbol{\theta}_i^-, i = 1, 2, \boldsymbol{\psi}^-$ via Eqn. (5)
    episode = episode + 1
end
```

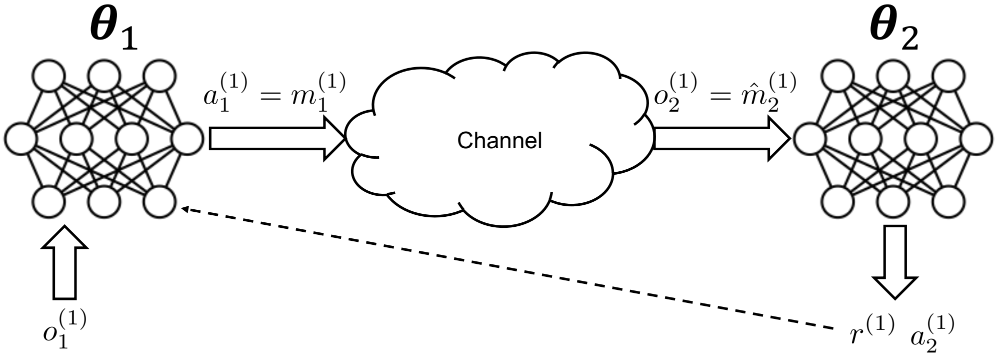


Fig. 4. Information flow between the transmitter and the receiver.

A similar problem is considered in [31] from a supervised learning perspective. The authors show that by approximating the gradient of the transmitter with the stochastic policy gradient of the vanilla REINFORCE algorithm [43], it is possible to train both the transmitter and the receiver without knowledge of the channel model. We wish to show here that this problem is actually a special case of the problem formulation we constructed in Section III and that by approaching this problem from a RL perspective, the problem lends itself to a variety of solutions from the vast RL literature.

Here, we opt to use DDPG to learn a deterministic joint channel coding-modulation scheme and use the DQN algorithm for the receiver, as opposed to the vanilla REINFORCE algorithm used in [31]. We use negative cross-entropy (CE) loss function as the reward function. It is defined as follows:

$$r^{(1)} = -L_{\text{CE}}(\hat{m}_1^{(1)}) = \sum_{k=1}^{2^B} \log(Pr(c_k|\hat{m}_1^{(1)})), \tag{15}$$

where c_k is the kth codeword in $\mathcal{O}_1$. Moreover, the receiver DQN is trained simply with the CE loss, while the transmitter DDPG algorithm receives the reward $r^{(1)}$. Similar to the *guided robot* problem in Section IV, we use replay buffer to improve the training process. We note here that in this problem, each episode is simply a 1-step MDP, as there is no state transition. As such, the replay buffers store only $(o_1^{(1)}, a_1^{(1)}, r^{(1)})$, $(o_2^{(1)}, a_2^{(1)}, r^{(1)})$ and a target network is not required. Consequently, the DDPG losses can be simplified as

$$L_{\text{DDPG}}^{\text{Critic}}(\boldsymbol{\theta}_1) = \Big(Q_{\boldsymbol{\theta}_1}(o_1^{(1)}, \mu_{\boldsymbol{\psi}}(o_1^{(1)}) - r^{(1)}\Big)^2, \tag{16}$$

$$L(\boldsymbol{\psi})_{\text{DDPG}}^{\text{Policy}} = -Q_{\boldsymbol{\theta}_1}(o_1^{(1)}, \mu_{\boldsymbol{\psi}}(o_1^{(1)})) \tag{17}$$

Furthermore, we improve upon the algorithm used in [31] by implementing a critic, which estimates the advantage of a given state-action pair by subtracting a baseline from policy gradient. That is, in the REINFORCE algorithm, the gradient is estimated as

$$\nabla_{\boldsymbol{\theta}_1} J(\boldsymbol{\theta}_1) = \nabla_{\boldsymbol{\theta}_1} \log \pi_1(a_1^{(1)}|o_1^{(1)};\boldsymbol{\theta}_1) r^{(1)} \ . \tag{18}$$

It is shown in [32] that by subtracting a baseline $b(o_1^{(1)})$, the variance of the gradient $\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$ can be greatly reduced. Herein, we use the value of the state, defined by Eqn. (1), except, in this problem, the trajectories all have length 1. Therefore, the value function can be simplified to

$$b(o_1^{(1)}) = v_{\pi_1}(o_1^{(1)}) = \mathbb{E}_{\pi_1}\left[r^{(1)}|o_1^{(1)}\right]. \tag{19}$$

The gradient of the policy with respect to the expected return $J(\boldsymbol{\theta}_1)$ is then

$$\nabla_{\boldsymbol{\theta}_1} J(\boldsymbol{\theta}_1) = \nabla_{\boldsymbol{\theta}_1} \log \pi_1(a_1^{(1)}|o_1^{(1)};\boldsymbol{\theta}_1)(r^{(1)} - v_{\pi_1}(o_1^{(1)})). \tag{20}$$

In practice, to estimate $v_{\Pi}(o_1^{(1)})$, we use a weighted moving average of the reward collected for a given state $o_1^{(1)} \in \mathcal{O}_1$ in $\mathcal{B}_1(o_1^{(1)}) = \{(o,a) \in \mathcal{B}_1 | o = o_1^{(1)}\}$ for the batch of trajectories $\mathcal{B}_1$:

$$v_{\pi_1}(o_1^{(1)}) \leftarrow (1-\alpha) v_{\pi_1}(o_1^{(1)}) + \frac{\alpha}{|\mathcal{B}_1(o_1^{(1)})|} \sum_{(o,a)\in\mathcal{B}_1(o_1^{(1)})} r^{(1)}(o,a), \tag{21}$$

where α is the weight of the average and $v_{\pi_1}(o_1^{(1)})$ is initialized with zeros. We use $\alpha = 0.01$ in our experiments. The algorithm for solving the joint channel coding and modulation problem is shown in Algorithm 3. The numerical results and comparison with alternative designs are presented in the next section.

VI. Numerical Results

We first define the DNN architecture used for all the experiments in this section. For all the networks, the inputs are processed by three fully connected layers followed by rectified linear unit (ReLU) activation function. The weights of the layers are initialized using Gaussian initialization with mean 0 and standard deviation 0.01. We store $100K$ experience samples in the replay buffer ($|\mathcal{R}_i| = 100K$), and sample batches of size 128 for training. We train every experiment for $500K$ episodes. The function used for ϵ-greedy exploration is

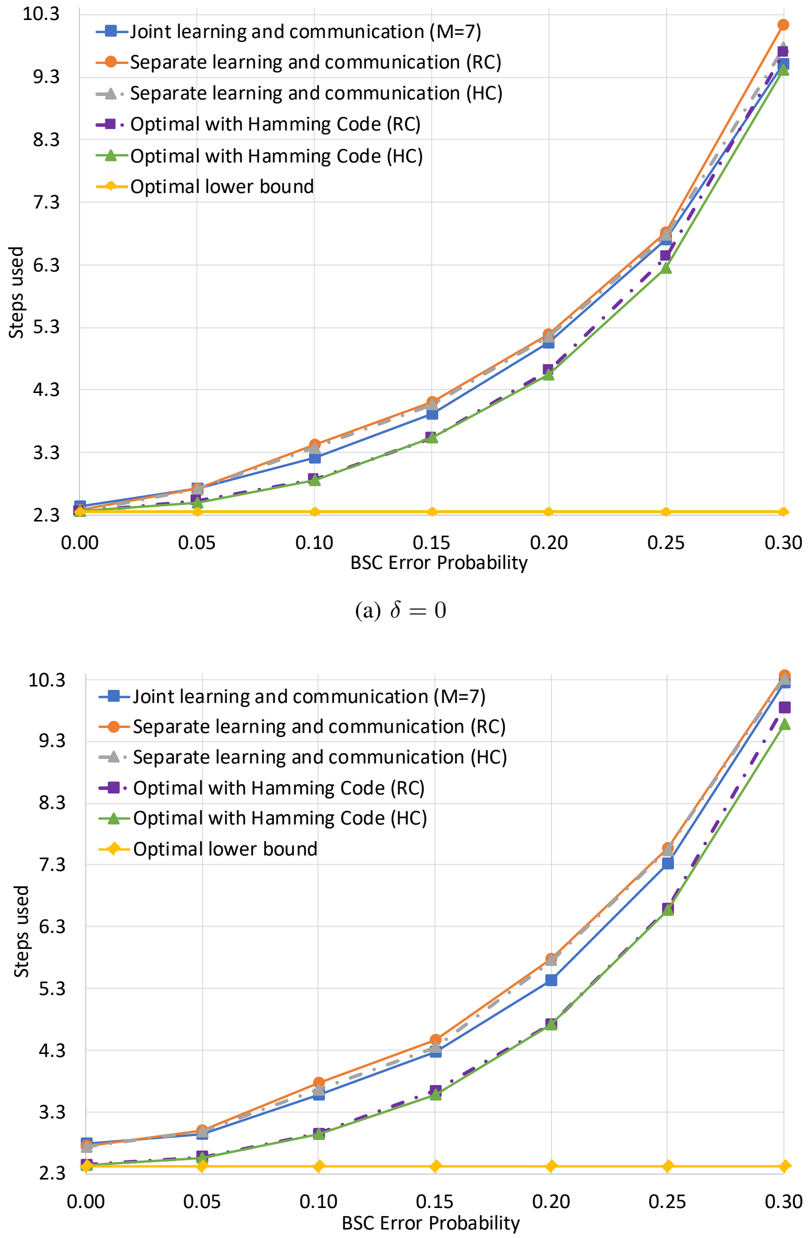
$$\epsilon = \epsilon_{\text{end}} + (\epsilon_0 - \epsilon_{\text{end}}) e^{\left(-\frac{\text{episode}}{\lambda}\right)} \tag{22}$$

Algorithm 3: Proposed solution for joint channel coding-modulation problem.

```
Initialize DNNs $\boldsymbol{\theta}_i, i = 1, 2$, with Gaussian initialization $\mathcal{N}(0, 10^{-2})$, and policy network $\boldsymbol{\psi}$ if using DDPG.
$episode = 1$
while $episode < episode\text{-}max$ do
    $\epsilon = \epsilon_{\text{end}} + (\epsilon_0 - \epsilon_{\text{end}})e^{-\frac{\text{episode}}{\lambda}}$
    Observe $o_1^{(1)} \sim \text{Uniform}(\mathcal{O}_1)$
    $m_1^{(1)} = \mu_{\boldsymbol{\psi}}(o_1^{(1)}) + w^{(1)}$
    Normalize $m_1^{(1)}$ via Eqn. (13)
    Observe $o_2^{(1)} = P_{\text{AWGN}}(\hat{m}_2^{(1)} | m_1^{(1)})$
    $a_2^{(1)} = \arg\max_a Q_{\boldsymbol{\theta}_1}(o_2^{(1)}, a)$
    Collect reward $r^{(1)}$
    Store experiences:
    $(o_1^{(1)}, a_1^{(1)}, r^{(1)}) \in \mathcal{R}_1$
    $(o_2^{(1)}, a_2^{(1)}, r^{(1)}) \in \mathcal{R}_2$
    Get batches $\mathcal{B}_1 \subset \mathcal{R}_1$, $\mathcal{B}_2 \subset \mathcal{R}_2$
    Compute average receiver loss $L_{\text{CE}}(o_2^{(1)}; \boldsymbol{\theta}_2)$ as in Eqn. (15) using batch $\mathcal{B}_2$
    Update $\boldsymbol{\theta}_2$ using $\nabla_{\boldsymbol{\theta}_2} L_{\text{CE}}(o_2^{(1)}; \boldsymbol{\theta}_2)$
    if use DDPG then
        Compute average transmitter losses $L_{\text{DDPG}}^{\text{Critic}}(\boldsymbol{\theta}_1)$ and $L_{\text{DDPG}}^{\text{Policy}}(\boldsymbol{\psi})$ as in Eqns. (16,17) using $\mathcal{B}_1$
        Update $\boldsymbol{\theta}_1$ and $\boldsymbol{\psi}$ $\nabla_{\boldsymbol{\theta}_1} L_{\text{DDPG}}^{\text{Critic}}(\boldsymbol{\theta}_1)$ and $\nabla_{\boldsymbol{\psi}} L_{\text{DDPG}}^{\text{Policy}}(\boldsymbol{\psi})$
    else if use REINFORCE then
        Compute average transmitter gradient $\nabla_{\boldsymbol{\theta}_1} J(\boldsymbol{\theta}_1)$ as in Eqn. (18) using $\mathcal{B}_1$
        Update $\boldsymbol{\theta}_1$ using $\nabla_{\boldsymbol{\theta}_1} J(\boldsymbol{\theta}_1)$
    else if use Actor-Critic then
        Compute average transmitter loss $\nabla_{\boldsymbol{\theta}_1} J(\boldsymbol{\theta}_1)$ as in Eqn. (20) using $\mathcal{B}_1$
        Update $\boldsymbol{\theta}_1$ using $\nabla_{\boldsymbol{\theta}_1} J(\boldsymbol{\theta}_1)$
        Update value estimate $v_{\pi_1}(o_1^{(1)})$ via Eqn. (21)
    episode = episode + 1
end
```

where λ controls the decay rate of ϵ. We use the ADAM optimizer [44] with learning rate 0.001 for all the experiments. The network architectures and the hyperparameters chosen are summarized in Table I.

For the grid world problem, presented in Section IV, the scout and treasure are uniformly randomly placed on any distinct locations upon initialization (i.e., $p_g \neq p_s^{(0)}$). The location of the scout and treasure are one-hot encoded to form a $2L^2$ vector that is the observation of the



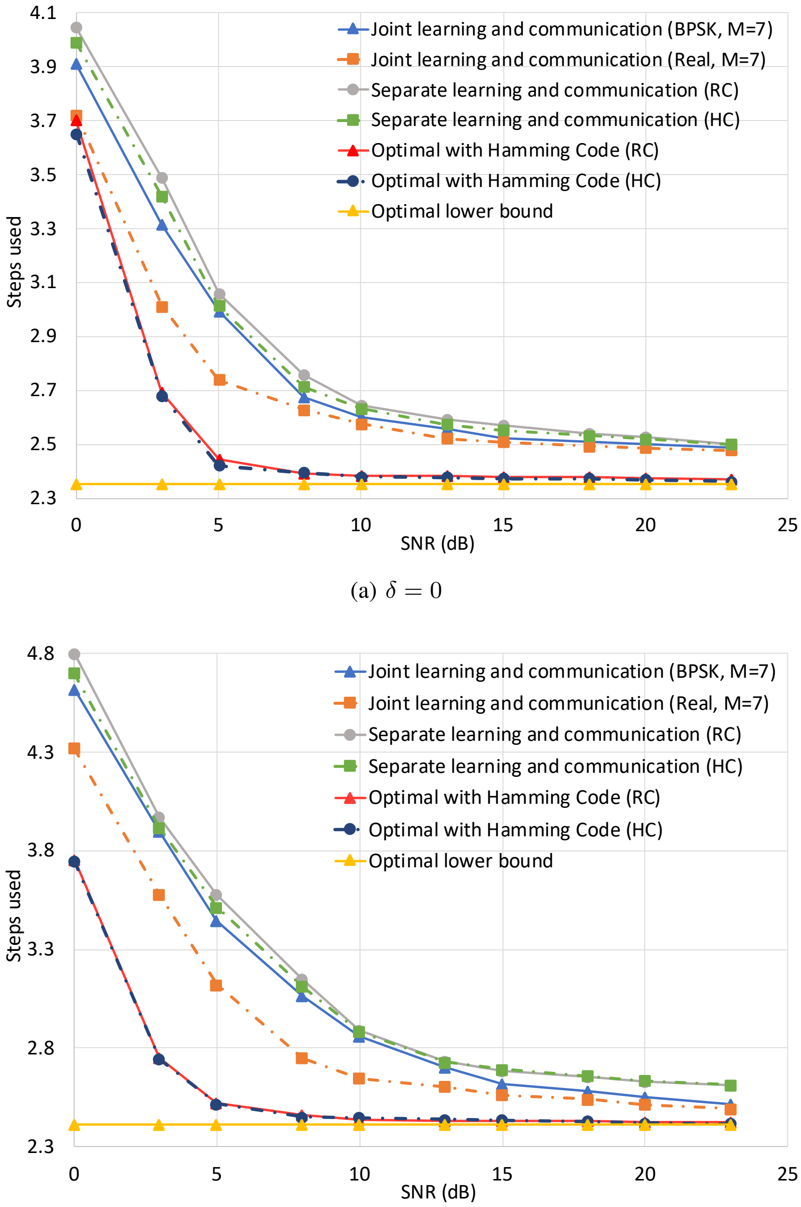
(a) $\delta = 0$

(b) $\delta = 0.05$

Fig. 5. Comparison of agents jointly trained to collaborate and communicate over a BSC to separate learning and communications with (7,4) Hamming code.

guide $o_1^{(t)}$.

In our simulation we fix the channel bandwidth to $M = 7$. We compare our solutions to a scheme that separates the channel coding from the underlying MDP. That is, we first train a RL agent that solves the grid world problem without communication constraints. We then introduce a noisy communication channel and encode the action chosen by the RL agent using a (7,4)



(a) $\delta = 0$

(b) $\delta = 0.05$

Fig. 6. Comparison of the agents jointly trained to collaborate and communicate over an AWGN channel to separate learning and communications with (7,4) Hamming code.

Hamming code before transmission across the channel. The received message is then decoded and the resultant action taken. We note that the (7,4) Hamming code is a perfect code that encodes four data bits into seven channel bits by adding three parity bits; thus, it can correct single bit errors. The association between the 16 possible actions and codewords of 4 bits can be done by random permutation, which we refer to as random codewords (RC), or hand-crafted

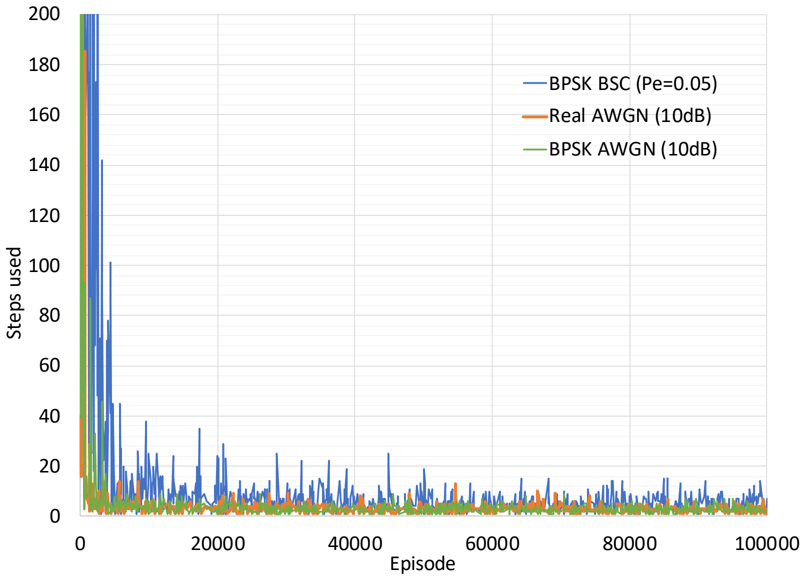


Fig. 7. Convergence of each channel scenario for the grip world problem without noise ($\delta = 0$).

TABLE I

DNN architecture and hyperparameters used.

$Q_{\boldsymbol{\theta}_i}$	μ_{ψ}	Hyperparameters
Linear: 64	Linear: 64	$\gamma = 0.99$
ReLU	ReLU	$\epsilon_0 = 0.9$
Linear: 64	Linear: 64	$\epsilon_{\text{end}} = 0.05$
ReLU	ReLU	$\lambda = 1000$
Linear: $\begin{cases} \lvert\mathcal{A}_i\rvert, & \text{if DQN,} \\ 1, & \text{if DDPG} \end{cases}$	Linear: $\dim(\mathcal{A}_i)$	$\tau = 0.005$

(HC) association by assigning adjacent codewords to similar actions, as shown in Fig. 2. By associating adjacent codewords to similar actions, even if the scout takes a different action than what was intended by the guide, due to channel errors, it would take a similar action as long as the number of bit errors is not too high. Lastly, we compute the optimal solution, where the steps taken are the shortest path to the treasure, and use a Hamming (7,4) channel code, referred to as "Optimal with Hamming Code". This acts as a lower bound for the separation-based results.

Fig. 5 shows the number of steps, averaged over 10K episodes, needed by the scout agent to reach the treasure for the BSC case. The optimal lower bound refers to the minimum number of steps required to reach the treasure assuming a perfect communication channel and acts as the lower bound for all the experiments. It is clear that the agents that jointly learn to communicate and collaborate over a noisy channel outperforms both the RC and HC separation results. It can also be seen that the hand-crafted (HC) codeword assignment gives better performance than random assignment (RC), suggesting that indeed by associating adjacent codewords to similar actions, the scout takes similar actions to the action intended by the guide even if there is an error. Moreover, neither the joint learning and communication results nor the separation-based results achieve the performance of the optimal solution with Hamming code. The gap between the optimal solution with Hamming code and the results obtained by the guide/scout formulation is due to the DQN architectures' limited capability to learn the optimal solution and the challenge of learning under noisy environments.

Similarly, in the AWGN case in Fig. 6, the results from joint learning and communication clearly outperforms those obtained via separate learning and communication. Here, the "Real" results refer to the guide agent where $\mathcal{A}_1 = \mathbb{R}^M$, while the "BPSK" results refer to the guide agent where $\mathcal{A}_1 = \{-1, +1\}^M$. The "Real" results here clearly outperform all other schemes considered. The relaxation of the channel constellation to all real values within a power constraint allows the guide to convey more information than a binary constellation can achieve. We observe that the gain from this relaxation is higher at lower SNR values for both δ values. We note that this is in contrast to the gap between the channel capacities achieved with Gaussian and binary inputs in an AWGN channel, which is negligible at low SNR values and increases with SNR. This shows that channel capacity is not the right metric in this problem, and even the two channels can be similar in terms of capacity, they can result in very different performances in terms of the discounted sum reward when used in the MARL context.

In both Figs. 5 and 6, it can be seen that when the grid world itself is noisy (i.e., $\delta > 0$), the agents are still able to collaborate, albeit at the cost of higher average steps required to reach the treasure. The convergence of the number of steps used to reach the treasure for each channel scenario is shown in Fig. 7.

We also study the affect of the bandwidth M on the performance. In Fig. 8, we present the average number of steps required for channel bandwidths $M = 7$ and $M = 10$. As expected, increasing the channel bandwidth improves the performance; that is, the scout reaches the treasure more quickly on average. The gain is particularly significant at the low SNR regime, as the guide is better able to protect the information conveyed against the channel noise thanks to the increased

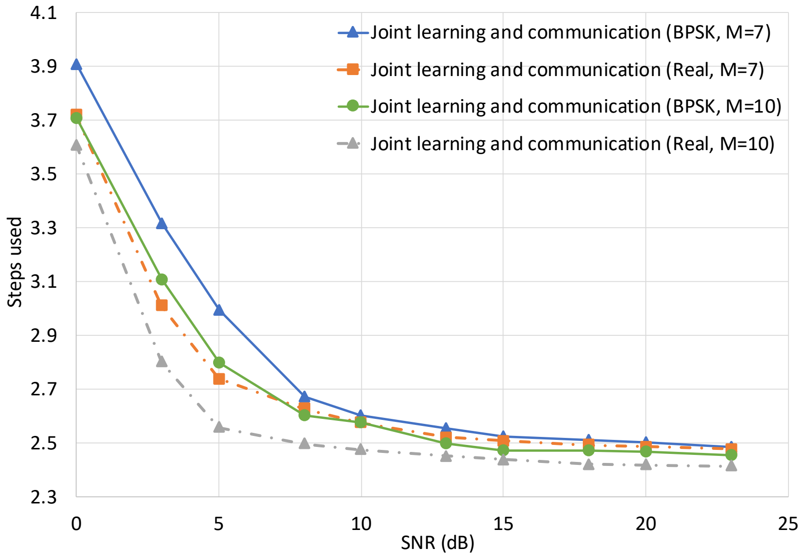


Fig. 8. Comparison of the number of channel uses $M = \{7, 10\}$ on the performance of the agents using an AWGN channel $(\delta = 0)$.

bandwidth.

For the joint channel coding-modulation problem, presented in Section V, we again compare the DDPG and actor-critic results with a (7,4) Hamming code using BPSK modulation. The input codeword is again one-hot encoded to form the input state $o_1^{(1)}$ of the transmitter. We also compare with the algorithm derived in [31], which uses supervised learning for the receiver and the REINFORCE policy gradient to estimate the gradient of the transmitter. Fig. 9 shows the BLER performance obtained by BPSK modulation and Hamming (7,4) code, our DDPG transmitter described in Section V, the one proposed by [31], and the proposed approach using an additional critic, labeled as “Hamming (7,4)”, “DDPG”, “REINFORCE”, and “Actor-Critic”, respectively. It can be seen that the learning approaches (DDPG, REINFORCE and Actor-Critic) perform better than the Hamming (7,4) code. Additionally, stochastic policy algorithms (REINFORCE and Actor-Critic) perform better than DDPG. This is likely due to the limitations of DDPG, as in practice, criterion 1) of Theorem 1 is often not satisfied. Lastly, we show that we can improve upon the algorithm proposed in [31] by adding an additional critic that reduces the variance of the policy gradients; and therefore, learns a better policy. The results obtained by the actor-critic algorithm are superior to those from the REINFORCE algorithm, especially at higher SNR regimes. On average, the learning-based results are better than the Hamming (7,4) performance by 1.24, 2.58 and 3.70 dB for DDPG, REINFORCE and Actor-Critic, respectively. Fig. 10 shows the convergence behavior of different learning algorithms for 5dB channel SNR. We reiterate that the joint channel coding and modulation problem studied from the perspective of supervised learning in [31] is indeed a special case of the joint learning and communication framework we presented in Section III from a MARL perspective, and can be solved using a myriad of algorithms from the RL literature.

Remark 1: We note that both the grid world problem and the channel coding and modulation problem are POMDPs. Therefore, recurrent neural networks (RNNs), such as long-short term memory (LSTM) [45] networks, should provide performance improvements as the cell states can act as belief propagation. However, in our initial simulations, we were not able to observe such improvements. We will continue to study the use of RNNs in this context by experimenting with different architectures, in which gains from recurrent architectures can be visible.

Remark 2: Even though we have only considered the channel modulation and coding problem in this paper due to lack of space, our framework can also be reduced to the source coding and joint source-channel coding problems by changing the reward function. If we consider an error-free channel with binary inputs and outputs, and let the reward depend on the average distortion between the B-length source sequence observed by agent 1 and its reconstruction generated by agent 2 as its action, we recover the lossy source coding problem, where the length-B sequence is compressed into M bits. If we instead consider a noisy channel in between the two agents, we recover the joint source-channel coding problem with an unknown channel model.

VII. Conclusion

In this paper, we have proposed a comprehensive framework that jointly considers the learning and communication problems in collaborative MARL over noisy channels. Specifically, we consider a MA-POMDP where agents can exchange messages with each other over an available noisy channel in order to improve the shared long-term average reward they can accrue. By considering the noisy channel as part of the environment dynamics and the message each agent sends as part of its action, the agents not only learn to collaborate with each other via communications but also learn to communicate "effectively". This corresponds to "level C" of Shannon and Weaver's organization of the communication problems in [2], which seeks to answer the question "How effectively does the received meaning affect conduct in the desired way?". We show that by jointly considering learning and communications in this framework, the learned joint

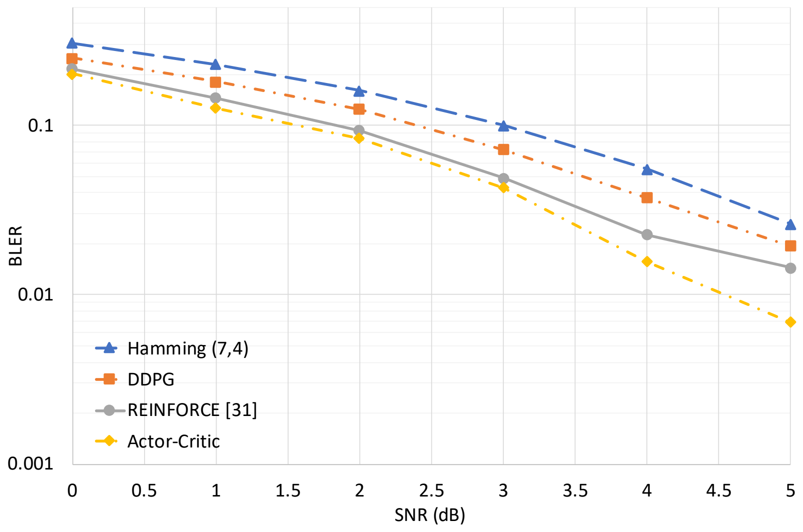


Fig. 9. BLER performance of different modulation and coding schemes over AWGN channel.

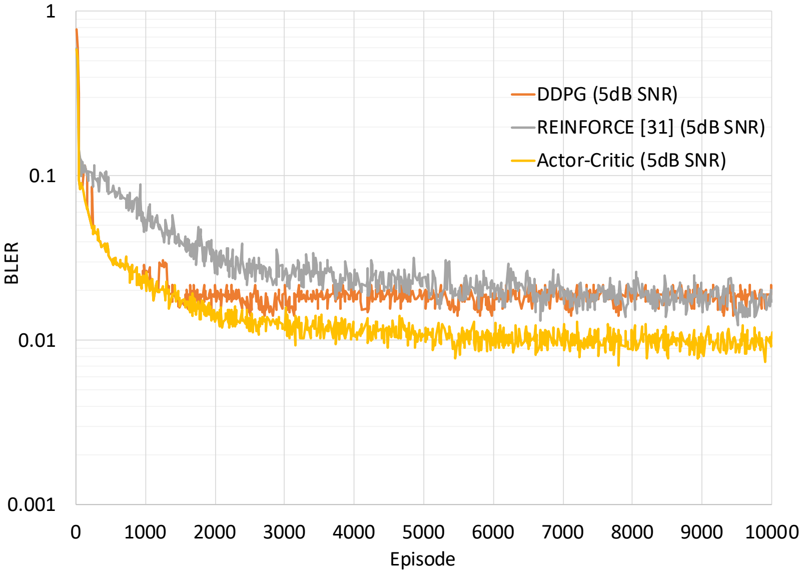


Fig. 10. Convergence of each learning algorithm for the joint channel coding and modulation problem.

policy of all agents is superior to that obtained by treating the communication and the underlying MARL problems separately. We emphasize that the latter is the conventional approach when MARL solutions typically obtained in the machine learning literature by considering error-free communication links are employed in practice when autonomous vehicles or robots communicate over noisy wireless links to achieve the desired coordination and cooperation. This approach inherently assumes that an underlying communication protocol takes care of the errors over the channel, but this introduces significant complexity and delays. Moreover, we demonstrate via numerical examples that the policies learned from our joint approach produce higher average rewards than those where separate learning and communication is employed. We also show that the proposed framework is a generalization of most of the communication problems that have been traditionally studied in the literature, corresponding to "level A" as described by Shannon and Weaver. This formulation opens the way to employing available numerical MARL techniques, such as the actor-critic framework, for the design of channel modulation and coding schemes for communication over unknown channels. We believe this is a very powerful framework, which has many real world applications, and can greatly benefit from the fast developing algorithms in the MARL literature to come up with novel communication codes and protocols, particularly with the goal of enabling collaboration and cooperation among distributed agents.

REFERENCES

[1] J. P. Roig and D. Gndz, "Remote reinforcement learning over a noisy channel," in *Proc. of IEEE GLOBECOM*, 2020.

[2] C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication*. Urbana, IL: University of Illinois Press, 1949.

[3] M. Tomasello, *Origins of Human Communication*. Cambridge, Mass.: The MIT Press, reprint edition ed., Aug. 2010.

[4] D. H. Ackley and M. Littmann, "Altruism in the evolution of communication," in *Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems (Artificial Life IV)*, pp. 40–48, Cambridge, MA: MIT Press, 1994.

[5] K.-C. Jim and C. L. Giles, "How communication can improve the performance of multi-agent systems," in *Proceedings of the Fifth International Conference on Autonomous Agents*, AGENTS '01, (New York, NY, USA), p. 584591, Association for Computing Machinery, 2001.

[6] B. Gler, A. Yener, and A. Swami, "The semantic communication game," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 4, pp. 787–802, 2018.

[7] P. Popovski, O. Simeone, F. Boccardi, D. Gunduz, and O. Sahin, "Semantic-effectiveness filtering and control for post-5G wireless connectivity," *Journal of Indian Inst of Sciences*, 2020.

[8] M. Kountouris and N. Pappas, "Semantics-empowered communication for networked intelligent systems," *arXiv cs.IT:2007.11579*, 2020.

[9] H. Xie, Z. Qin, G. Y. Li, and B.-H. Juang, "Deep learning enabled semantic communication systems," *arXiv eess.SP:2006.10685*, 2020.

[10] E. C. Strinati and S. Barbarossa, "6G networks: Beyond shannon towards semantic and goal-oriented communications," *arXiv cs.NI:2011.14844*, 2020.

[11] E. Bourtsoulatze, D. B. Kurka, and D. Gunduz, "Deep Joint Source-Channel Coding for Wireless Image Transmission," *arXiv:1809.01733 [cs, eess, math, stat]*, Sept. 2018. arXiv: 1809.01733.

[12] Z. Weng, Z. Qin, and G. Y. Li, "Semantic communications for speech signals," *arXiv eess.AS:2012.05369*, 2020.

[13] S. Sreekumar and D. Gndz, “Distributed Hypothesis Testing Over Discrete Memoryless Channels,” *IEEE Transactions on Information Theory*, vol. 66, pp. 2044–2066, Apr. 2020. Conference Name: IEEE Transactions on Information Theory.

[14] M. Jankowski, D. Gndz, and K. Mikolajczyk, “Wireless Image Retrieval at the Edge,” *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 1, pp. 89–100, 2021.

[15] D. Gunduz, D. B. Kurka, M. Jankowski, M. M. Amiri, E. Ozfatura, and S. Sreekumar, “Communicate to Learn at the Edge,” *IEEE Communications Magazine*, 2021.

[16] A. R. Balch, T., “Communication in reactive multiagent robotic systems,” *Autonomous Robots*, pp. 27–52, 1994.

[17] J. N. Foerster, Y. M. Assael, N. de Freitas, and S. Whiteson, “Learning to Communicate with Deep Multi-Agent Reinforcement Learning,” *arXiv:1605.06676 [cs]*, May 2016. arXiv: 1605.06676.

[18] J. Jiang and Z. Lu, “Learning attentional communication for multi-agent cooperation,” in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, p. 72657275, 2018.

[19] N. Jaques, A. Lazaridou, E. Hughes, C. Gulcehre, P. A. Ortega, D. J. Strouse, J. Z. Leibo, and N. de Freitas, “Social Influence as Intrinsic Motivation for Multi-Agent Deep Reinforcement Learning,” *arXiv:1810.08647 [cs, stat]*, June 2019. arXiv: 1810.08647.

[20] A. Das, T. Gervet, J. Romoff, D. Batra, D. Parikh, M. Rabbat, and J. Pineau, “TarMAC: Targeted multi-agent communication,” in *Proceedings of the 36th International Conference on Machine Learning* (K. Chaudhuri and R. Salakhutdinov, eds.), vol. 97 of *Proceedings of Machine Learning Research*, (Long Beach, California, USA), pp. 1538–1546, PMLR, 09–15 Jun 2019.

[21] J. Wang, J. Liu, and N. Kato, “Networking and Communications in Autonomous Driving: A Survey,” *IEEE Communications Surveys Tutorials*, vol. 21, no. 2, pp. 1243–1274, 2019. Conference Name: IEEE Communications Surveys Tutorials.

[22] M. Campion, P. Ranganathan, and S. Faruque, “UAV swarm communication and control architectures: a review,” *Journal of Unmanned Vehicle Systems*, Nov. 2018. Publisher: NRC Research Press.

[23] A. Lazaridou, A. Peysakhovich, and M. Baroni, “Multi-Agent Cooperation and the Emergence of (Natural) Language,” *arXiv:1612.07182 [cs]*, Mar. 2017. arXiv: 1612.07182.

[24] A. Lazaridou, A. Potapenko, and O. Tieleman, “Multi-agent communication meets natural language: Synergies between functional and structural language learning,” 2020.

[25] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, pp. 529–533, Feb. 2015. Number: 7540 Publisher: Nature Publishing Group.

[26] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, “Continuous control with deep reinforcement learning,” *arXiv:1509.02971 [cs, stat]*, July 2019. arXiv: 1509.02971.

[27] E. Nachmani, E. Marciano, L. Lugosch, W. J. Gross, D. Burshtein, and Y. Beery, “Deep learning methods for improved decoding of linear codes,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 119–131, 2018.

[28] S. Dorner, S. Cammerer, J. Hoydis, and S. ten Brink, “On deep learning-based communication over the air,” in *2017 51st Asilomar Conference on Signals, Systems, and Computers*, pp. 1791–1795, 2017.

[29] A. Felix, S. Cammerer, S. Drner, J. Hoydis, and S. Ten Brink, “OFDM-autoencoder for end-to-end learning of communications systems,” in *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1–5, 2018.

[30] D. B. Kurka and D. Gndz, “Deepjscc-f: Deep joint source-channel coding of images with feedback,” *IEEE Journal on Selected Areas in Information Theory*, vol. 1, no. 1, pp. 178–193, 2020.

[31] F. A. Aoudia and J. Hoydis, "Model-Free Training of End-to-End Communication Systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 11, pp. 2503–2516, 2019.

[32] V. Konda and J. Tsitsiklis, "Actor-critic algorithms," in *Advances in Neural Information Processing Systems* (S. Solla, T. Leen, and K. Müller, eds.), vol. 12, pp. 1008–1014, MIT Press, 2000.

[33] K. Wagner, J. A. Reggia, J. Uriagereka, and G. S. Wilkinson, "Progress in the Simulation of Emergent Communication and Language:," *Adaptive Behavior*, July 2016. Publisher: SAGE Publications.

[34] S. Sukhbaatar, A. Szlam, and R. Fergus, "Learning multiagent communication with backpropagation," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, (Red Hook, NY, USA), pp. 2252–2260, Curran Associates Inc., Dec. 2016.

[35] S. Havrylov and I. Titov, "Emergence of Language with Multi-agent Games: Learning to Communicate with Sequences of Symbols," p. 11, 2017.

[36] R. E. Wang, M. Everett, and J. P. How, "R-MADDPG for Partially Observable Environments and Limited Communication," *arXiv:2002.06684 [cs]*, Feb. 2020. arXiv: 2002.06684.

[37] R. Lowe, Y. WU, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, pp. 6379–6390, Curran Associates, Inc., 2017.

[38] P. Peng, Y. Wen, Y. Yang, Q. Yuan, Z. Tang, H. Long, and J. Wang, "Multiagent Bidirectionally-Coordinated Nets: Emergence of Human-level Coordination in Learning to Play StarCraft Combat Games," *arXiv:1703.10069 [cs]*, Sept. 2017. arXiv: 1703.10069.

[39] A. Mostaani, O. Simeone, S. Chatzinotas, and B. Ottersten, "Learning-based Physical Layer Communications for Multiagent Collaboration," in *2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pp. 1–6, Sept. 2019. ISSN: 2166-9589.

[40] T. Tung and D. Gndz, "SparseCast: Hybrid Digital-Analog Wireless Image Transmission Exploiting Frequency Domain Sparsity," *IEEE Communications Letters*, pp. 1–1, 2018.

[41] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic Policy Gradient Algorithms," p. 9, 2014.

[42] G. E. Uhlenbeck and L. S. Ornstein, "On the Theory of the Brownian Motion," *Physical Review*, vol. 36, pp. 823–841, Sept. 1930.

[43] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, vol. 8, pp. 229–256, May 1992.

[44] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv:1412.6980 [cs]*, Jan. 2017. arXiv: 1412.6980.

[45] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, p. 17351780, Nov. 1997.