Multi-Agent Cooperation Against Adversarial Agents Through Communication

Mihir Patel [*1] **Nikhil Sardana** [*1] **Vinjai Vale** [*1]

Abstract

Communication is a powerful tool for multi-agent challenges, but reinforcement learning techniques still struggle to learn it effectively. In this work, we explore how communication is altered and learned when playing with an adversary who can listen in on the communication. We induce noise and perturb environment factors and observe differences in communication and cooperative behavior. We additionally explore properties of MADDPG, a popular multi-agent cooperative learning algorithm, analyzing how discrete parameterization of communication affects performance.

1. Introduction

People fail to get along . . . because they have not communicated with each other.
– Martin Luther King Jr.

Human efforts, especially in group tasks, critically rely on communication. From work to sports to research, effective communication drives our society. Yet often, we want our communication to be private and secure, indecipherable by potentially malicious third parties. Take, for example, baseball, where the pitcher signals to the catcher the pitch without letting the batter know. Humans are remarkably capable of not only communicating effectively, but also secretly or in codes.

Similarly, in a reinforcement learning setting, we wish to train multiple agents, commonly referenced as multi-agent RL, in a manner where agents can learn effective, secure communication patterns. This can help in a wide variety of environments and problems, especially partially observable or team settings, where messaging allows agents to share and maximize observable information to reach a common goal.

In this paper, we explore communication between cooperating agents in complex multi-agent environments with adversarial agents. Early work by Foerster et al. (2016) was the one of the first papers to look at the role of communication in reinforcement learning. Foerster et al. (2016) proposed ways of having communication channels and how gradients may flow through them.

Building on this, Lowe et al. (2017) explores multi-agent communication in simple environments with adversaries. The "covert communication" environment in Lowe et al. (2017) involves two allied agents who send messages and are rewarded when they decrypt ally messages more successfully than an adversarial agent. In this communication environment, Lowe et al. (2017) found allies were able to successfully decrypt messages 40% more often than an adversary.

However, the covert communication environment in Lowe et al. (2017) is so limited, it is unclear if the positive results will generalize to more complex environments, where communication is only one component. In the covert communication environment, allies have only one action: sending messages. Furthermore, allies can only choose between two possible length-4 messages. There are also only two possible keys, and both allies are given the key at the start of each episode.

Recent survey papers discuss progress in cooperative communication and multi-agent RL, summarizing efforts and showing the state of communication against an adversary has not improved significantly recently (Zaïem & Bennequin, 2019; Nguyen et al., 2018). To the best of our knowledge, the MADDPG (Multi-Agent Deep Deterministic Policy Gradient) algorithm introduced in Lowe et al. (2017) is still state of the art for contested communication.

Lastly, we consider how communication is parameterized. Recent work by OpenAI (Mordatch & Abbeel, 2017) proposes using the Gumbel-Softmax trick in communication as information should be discrete bits. However, it is not clear this process is the best parameterization and the claimed

---

[*]Equal contribution [1]Department of Mathematics, Stanford University, Stanford, California, USA. Correspondence to: Mihir Patel <mihirp@stanford.edu>, Nikhil Sardana <nsardana@stanford.edu>, Vinjai Vale <vinjai@stanford.edu>.

Proceedings of the Winter 2020 offering of Computer Science Two Hundred and Thirty Four, Stanford, California, USA.

benefits of lower gradient variance are not tested.

In this work, we extend upon several of these recent ideas to analyze characteristics of multi-agent communication with adversaries. We present three contributions in this paper:

- We develop and run implementations of the environments in Lowe et al. (2017), for which, to our knowledge, there are no publicly-available weights. We provide some analysis of these results.
- We create two complex communication-based multi-agent environments, incorporating cooperative and adversarial elements. We test MADDPG on these environments and provide a thorough analysis with environment perturbations.
- We analyze parameterizations of the discrete communication and action space, in the process finding an error in the original paper and giving a theoretical improvement on MADDPG.

Our primary motivation is to determine if agents can learn to incorporate communication even when it only indirectly affects the final reward, or payoffs from communicating come far after messages are sent. Previous work has found incentivizing communication difficult, especially when agents can perform reasonably well without communication.

2. Approach

2.1. Contribution I: Building Baselines

We first reproduce results by Lowe et al. (2017) to serve as baselines. This involves setting up the environments described in the paper and using the MADDPG algorithm. These baselines will be used to compare against the modification proposed in Contribution 3.

Specifically, we run baselines for four of the environments described in the original paper. These environments are:

1. `simple_crypto`: Two agents, Alice and Bob, each have a private key and attempt to communicate. Eve, a third adversarial agent, tries to decipher their message.
2. `simple_speaker_listener`: On a map with many landmarks, only the speaker knows which landmark is the goal. The speaker must communicate the goal's identity to a listener who moves towards it.
3. `simple_tag`: A predator-prey environment where faster prey attempt to avoid a greater number of slower predators.
4. `simple_world_comm`: A more advanced predator-prey with food for prey, forests where prey can hide, and a predator leader who can always see the prey.

2.2. Contribution II: New Environments

The current suite of environments is unfortunately not sufficient enough for the problem of communication. These environments, although involving communication, were not designed to primarily test this functionality and are rudimentary in terms of the complexity of the task. As a result, we propose two significantly more challenging tasks which force the agents to combine communication with other actions to reach an optimal reward.

Our two environments are:

1. *comm*`plex_speaker_listener`: An adversarial agent is added to `simple_speaker_listener`. The speaker/listener pair must learn encrypt communication to reach the goal while preventing an adversary from doing the same.
2. *comm*`plex_tag`: Predators are given communication abilities in `simple_tag` but have vision restricted and the environment becomes partially observable.

We augment the traditional speaker-listener environment by adding a random private key, which only the speaker and listener have access to. All agents, including the adversary, have access to the public communication channel.

All agents can also see the n landmarks scattered on a map ($n = 3$ for all our experiments), but only one is the target. As before, only the speaker knows the identity of the target.

At the end of each episode, the speaker and listener receive the same reward, which is proportional to the listener's distance from the target subtracted from adversary's distance to the target. The adversary receives a reward proportional to its distance to the target. Hence, the speaker must learn to use both public communication and the private key to tell the listener the identity of the target without disclosing it to the adversary. An image of our speaker-listener environment is shown in Figure 1.

Crucially, both our speaker-listener and predator-prey environments are partially observable, so communication can be used to maximize information. High reward on either task is nearly impossible without encrypted or effective communication, respectively.

In both of these environments, we conduct thorough ablation studies to push and understand the limits of MADDPG and learning communication capabilities in tricky multi-agent settings. In our speaker-listener environment, we modify the communication bandwidth and induce noise into the adversary's channel. In our predator-prey environment, we vary the vision of predators, communication audience, communication complexity, and number of agents.

All environments and baselines were trained consistent with

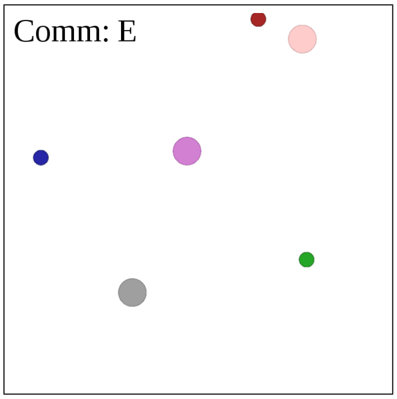


Figure 1. `commplex_speaker_listener`. The small circles are the landmarks. In each episode, a random landmark is chosen to be the target. Only the speaker (grey) knows the target's identity. The speaker cannot move, instead, it can only communicate over a public channel to a listener and adversary (purple). The listener and adversary both move during the episode to minimize their distance to the target. The speaker must use a private key (shared with only the listener) and public communication to tell the listener the identity of the target landmark without disclosing it to the adversary. In the figure above, the listener is colored a light shade of the target. As shown above, when the adversary receives noise instead of communication, it learns to find the centroid of the three landmarks.

the original MADDPG implementation. We used a 3-layer MLP with 64 units in each layer, and a ReLU activation following each layer. The learning rate was initialized to 0.01. We used the Adam optimizer for gradient optimization. The discount factor was set to $\gamma = 0.95$.

2.3. Contribution III: Discrete Parameterizations

In machine learning, many problems require having discrete samples. Often, we would like these samples to be differentiable so that we can flow gradients through them. Naively estimating the gradient of a discrete sample is challenging because we cannot directly differentiate and using Monte Carlo methods results in high variance. Additionally, the reparameterization trick used in continuous samples is not readily available. To address this challenge, Jung et al. (2016) propose the Gumbel-Softmax distribution which can be smoothly annealed to a categorical distribution. This distribution is defined as

$$y_i = \frac{\exp\left(\left(\log\left(\pi_i\right) + g_i\right)/\tau\right)}{\sum_{j=1}^{k} \exp\left(\left(\log\left(\pi_j\right) + g_j\right)/\tau\right)}$$

where the stochasticity is derived from $g \sim$ Gumbel(0,1). We note that g can be efficiently computed with $g = -\log(-\log(u))$. In this setting, we can easily differentiate with respect to π_i and not worry about stochasticity. Note that τ is a temperature variable which is typically annealed and presents a tradeoff between bias and variance.

MADDPG uses the Gumbel-Softmax distribution in order to parameterize communications and actions (Lowe et al., 2017). This is in line with earlier work which pushes gradients through the communication channel (Foerster et al., 2016). With this distribution, communication channels can be changed into discrete variables. However, a closer look reveals this parameterization is unnecessary.

Crucially, MADDPG does *not* push gradients through the communication channel. In fact, the paper states "we do not assume any particular structure on the communication method between agents (that is, we don't assume a differentiable communication channel)," (Lowe et al., 2017). Since MADDPG does not differentiate with respect to the sampling, we do not need to use the Gumbel-Softmax trick and instead can apply a standard softmax.

By using softmax, we might better encourage exploration. The Gumbel-Softmax distribution forces logits which give more uncertain values towards a categorical distribution, penalizing exploration compared to softmax. This is normally countered with a high temperature value that is annealed down. However, in the MADDPG hyperparameters and implementation, there is no annealing.

To measure if using the Gumbel-Softmax distribution under these settings actually results in any gain, we compare results with the two distributions across all environments. In these experiments, the only difference is the final distribution parameterized by the logits, and all hyperparameters remain the same.

Unfortunately, some of the other claims in the paper regarding the discrete parameterization are also wrong. Lowe et al. (2017) claims "to support discrete communication messages, we use the Gumbel-Softmax estimator." When inspecting the code provided for MADDPG, this is not true. The Gumbel-Softmax distribution provides a soft relaxation, so the output vector is not one-hot but is smooth. *The code provided and the results in the paper do not use discrete communication messages, contradicting the claims made in the paper*. Instead, smooth Gumbel-Softmax samples are used.

We find this discrepancy in the public MADDPG GitHub repository, an official release by OpenAI. We verified the communications were not discrete by injecting print statements into the TensorFlow compute graph. Additionally, we find the only point in which the message is discrete is in the renderer, where the display shows a discretized message via `argmax`.

We reached out to the authors of this paper with regards to these two issues. They agreed that while Gumbel-Softmax does provide a smooth approximation to discrete channels, it doesn't actually have discrete communication. They also agreed using softmax would be an interesting exploration as opposed to Gumbel-Softmax. After our communication, Lowe et al. updated the MADDPG arXiv paper on March 14, 2020, to correct the issue of discrete communications.

3. Results

3.1. Contribution I: Building Baselines

For each of the environments we use from the original MADDPG paper, we train on each environment for 60,000 episodes, We report rewards for each one in Table 1.

crypto	speaker listener	tag	world comm
15.773	-54.740	12.441	27.294

Table 1. Baseline results for existing cooperative/competitive environments.

We also visualize each environment and find the performance is generally proficient, indicating that the baseline correctly converged. We provide an example in Figure 2.

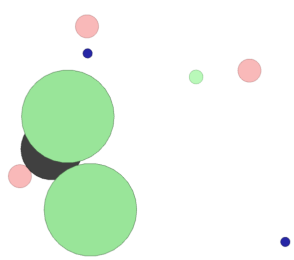


Figure 2. simple_world_comm. The green prey are moving towards blue food, hiding in the large green forests, and avoiding the red predators.

3.2. Contribution II: New Environments

3.2.1. Adversarial Speaker-Listener

MADDPG yields poor results on our adversarial speaker-listener environment. As shown in Figure 3, after training for 120,000 epochs, the speaker and listener consistently average essentially zero reward, meaning the listener and adversary end the same distance from the goal, despite the fact that the listener shares a private key with the speaker. It appears MADDPG is not powerful enough to directly learn encrypted communication for scenarios even slightly more complex than the trivial simple_crypto environment in Lowe et al. (2017).

Upon further inspection, we notice that not only do the speaker and listener not use their private key, they also fail

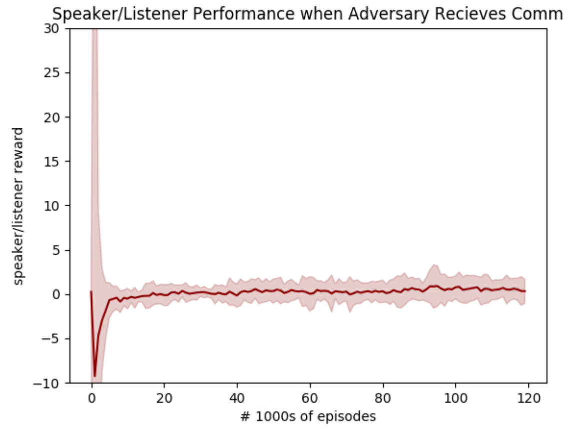


Figure 3. Without modifying the environment or training, the listener does not outperform the adversary. The dark red line is the mean over 10 trials and the shaded region shows $\pm$ 2 standard deviations. Zero reward indicates the listener and adversary ended the episode the same distance from the target.

to use the public communication channel. In most trials, the speaker simply communicates the same letter, regardless of the identity of the target landmark.

In the standard speaker-listener environment, the speaker and listener learn quickly to associate a different letter with each landmark (A for red, B for blue, etc.) We might expect the same behavior in our environment. First, we imagine the speaker and listener learn to use the public communication channel and associate a different letter with each landmark. Then, the speaker and listener learn to use the key to encrypt this signal.

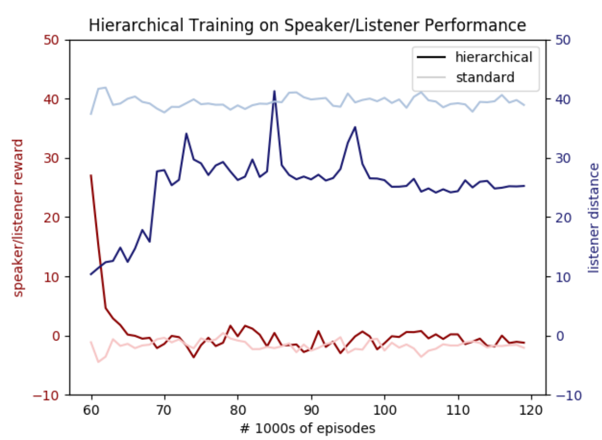
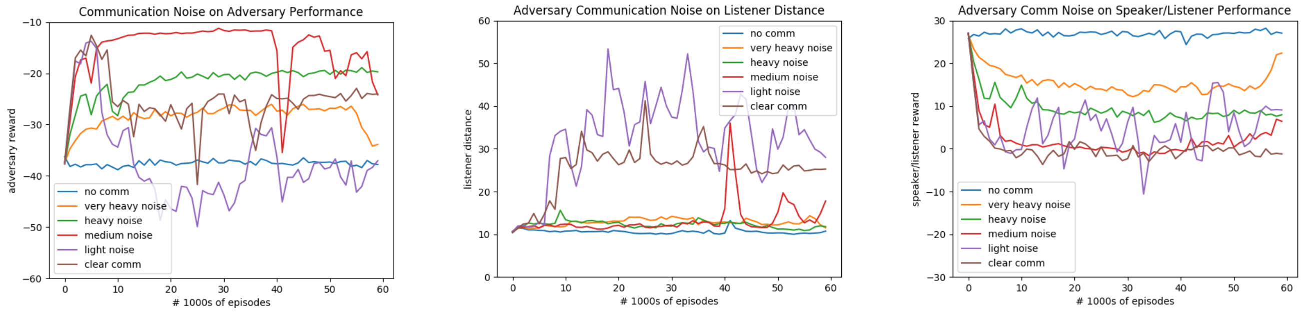


Figure 4. Pre-training the speaker/listener with an adversary receiving noise produces similar reward to standard training (red lines) but a smaller listener distance to target (blue lines).

However, because the speaker and listener are rewarded



(a) Adversary performance varies across noise levels because the speaker adapts.

(b) The listener gets closer to the target when the adversary receives heavy noise.

(c) Increasing adversary's communication noise decreases speaker/listener rewards.

Figure 5. Effects of adding noise to the adversary's communication channel in the speaker-listener environment.

for better performance relative to the adversary, there is no incentive for the speaker to communicate the target publicly. Both the listener and adversary learn using the same algorithm, so any public information will be exploited equally by both parties, providing zero reward to the speaker.

Since we imagine it would be easier for the speaker and listener to learn encrypted communication if they first learn unencrypted communication, we use a technique similar to hierarchical reinforcement learning and learn the easier task first. First, we train a modified version of our speaker-listener environment where the adversary receives noise instead of public communication. After 60,000 episodes, we freeze this model and then resume training for another 60,000 episodes with the adversary receiving communication. Since the adversary receives only noise initially, the speaker and listener are free to communicate without encryption. Only once they have learned to communicate the target freely do we encourage them to encrypt.

We compare this hierarchical training procedure to standard training, where the adversary always receives public communication. As shown in Figure 4, this procedure is only partially successful. Although the hierarchical model's listener ends significantly closer to the target, both models have similar final rewards. In other words, after the adversary gains communication access, the speaker and listener retain most of their ability to communicate the target publicly, but do not learn any encryption protocol.

We further test the effects of inducing noise in the adversary's communication channel. Once again, we train on a modified speaker-listener environment, where the adversary receives pure noise and no communication. Then, we run six different tests, with the adversary receiving either clear communication, pure noise, or varying levels of noisy communication. Specifically, for each dimension of the speaker's communication vector, we sample from a Gaussian $\mathcal{N}(0, \sigma)$ and add this sample to the communication before the adversary observes it. Each element of the speaker's communication vector is in $[0, 1]$. We test on light noise, medium noise, heavy noise, and very heavy noise ($\sigma = 0.1, 0.5, 1.0$, and 2.0, respectively), along with clear communication ($\sigma = 0$) and our baseline, where the adversary continues to receive pure noise (no comm). The listener does not receive any noise in its communication channel. The results are shown in Figure 5 above.

As expected, the speaker/listener reward increases consistently as the adversary's noise increases (Figure 5c). However, the adversary's performance does not decline uniformly, as we might expect (Figure 5a). Intuitively, we expect the adversary to be closest to the target when it receives the least noise, and furthest when it receives no communication at all.

These mixed results are explained by differences in the speaker's behavior. Upon displaying the trained result for $\sigma = 0$ and 0.1, we find the speaker and listener have partially devolved, using only two communication patterns and ignoring one of the landmarks. The adversary's noise is light enough that it can successfully decipher any publicly-communicated information about the target location. So, the speaker has learned that public communication provides limited reward benefit, and thus both the adversary and listener perform worse in these scenarios. On the other hand, when the adversary receives heavy or very heavy noise, it is only partially able to decipher the speaker's public communication. Hence, the speaker and listener have an incentive to retain their public communication, and the adversary performs better as well, since it can still determine the target some of the time. In no environment do we observe any encryption learned between the speaker and listener.

3.2.2. Predator-Prey with Communication

Our main experiments were conducted on a modified predator-prey setting with 4 predators and 1 prey, where one of the predators (the "leader") is granted the ability to send messages, and the other predators (the "followers")

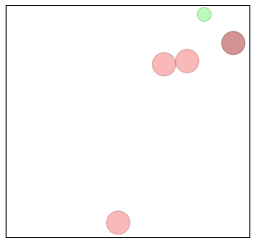


Figure 6. One of the light red followers stays back as a safety, moving to intercept the green prey as it escapes to the left from the dark red leader and two other followers.

can listen. As in the previous section, all predators are also given a common message key every round that they can learn to use. We observe emergent behavior that parallels pack hunting behavior in nature. When predators are granted full vision, we observe the following, regardless of whether communication is enabled or not:

- Predators approach the prey from multiple angles in order to limit its escape paths.
- Predators generally achieve success by cornering the prey along an edge or in one of the four corners of the square map.
- One predator generally stays farther back than the others, acting as a "safety" to chase down a prey in case it escapes between or around the attackers (see Figure 6).

Communication does not appear to significantly improve performance when all predators have full vision. However, we find that communications do yield significant improvement when vision is restricted, as in Figure 7. When all predators have full vision, however, the system doesn't appear to utilize the communication system – in fact, at the end of the training process the leader doesn't even use one of the four communications available to it.

In all the above trials, the leader may send one of four communications in each step: 'A', 'B', 'C', or 'D'. We find that four is a good number of communication channels for this task. In particular, fewer communication channels nullify the benefit of communications in the first place, and the algorithm is unable to utilize more communication channels for better performance. This is exemplified in Figure 8.

Next, we introduce an adversarial element similar to our first environment, to see if that would influence the way the predators behaved or used communications. We find that allowing the prey to eavesdrop on communication does not alter performance, as visualized in Figure 9.

Finally, we attempt to make the environment a bit more complex by increasing the number of agents. When we

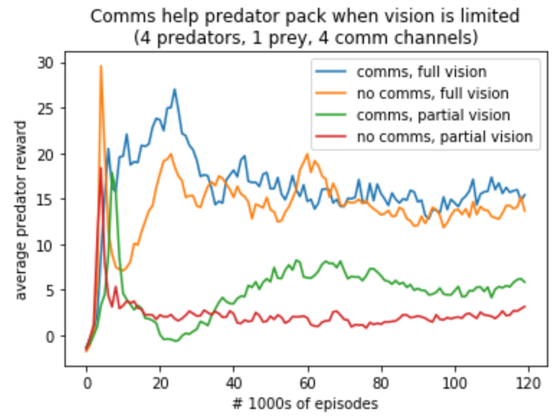


Figure 7. Comparison of how communication helps predators depending on vision. In the partial vision case, the light red followers cannot see the green prey, and only the dark red leader can see the prey.

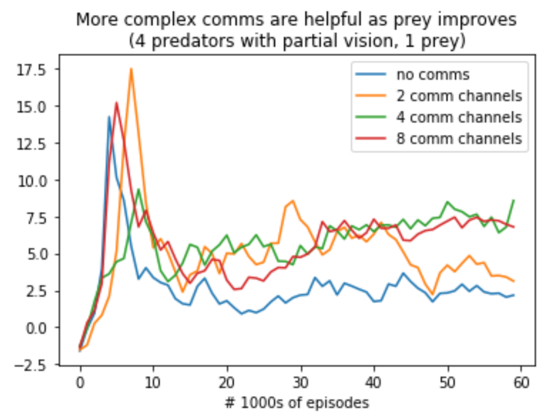


Figure 8. More communication channels yield better performance, but the algorithm is unable to learn to utilize more than four channels. Interestingly, two channels works well up to a certain point, where there is a marked decrease in predator reward around 45k episodes where the prey improves.

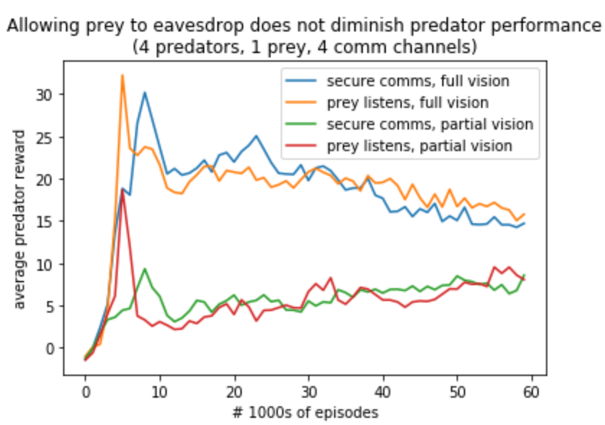
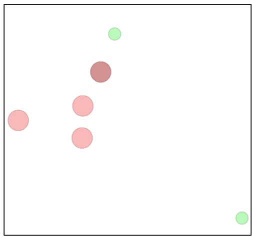
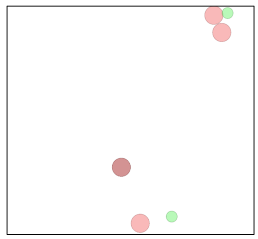


Figure 9. Allowing the prey to eavesdrop on predators does not alter the predators' performance.



(a) All four agents converge on one of the prey in an attempt to corner it. The other prey is ignored and hides from the action.



(b) The predators split up; two corner one prey in the upper right while the other two pursue the prey at the bottom.

Figure 10. Emergent behavior in the multiple-prey setting.

add two prey with partially blind predators that cannot see the prey, we observe some interesting emergent behavior (see Figure 10). Sometimes all four predators converge on a single prey, and sometimes two split off to corner one prey while the other two chase another. There was no clear reason as to why one happened over the other, but it accurately parallels what we see in nature where a pack of predators approach several prey and thus need to coordinate amongst themselves which one to focus on.

3.3. Contribution III: Discrete Parameterizations

We test the two possible parameterizations for communication and action, softmax and Gumbel-Softmax, on the previous environments. For consistency, we use the same hyperparameters in each setting as with the original paper and do not make any modifications besides in the parameterization step.

As Table 2 shows, we see there is no clear winner across all environments. In general, softmax has larger margins when it outperforms Gumbel-Softmax, most noticeably in the speaker-listener environment. By contrast, in the environments where Gumbel-Softmax has higher rewards, the margins are generally small.

When looking at the training curves, we see softmax tends to have more fluctuation then Gumbel-Softmax as highlighted by Figure 11 and Figure 12. We theorize that softmax achieves higher performance in the environments and exhibits this greater fluctuation because it explores more compared to Gumbel-Softmax. In particular, Gumbel-Softmax forces the distribution toward a categorical distribution as it produces a soft one-hot vector, so it puts more weight on one specific value compared to softmax. As softmax is more spread out for the same set of logits (it generally has higher entropy), it encourages greater exploration. However, while there is evidence supporting this hypothesis, we stress that there could be other explanations and this is far from conclusive.

Figure 11. A comparison of softmax (top) and gumbel (bottom) per agent training curves on the crypto environment. In general, we see more fluctuation in the softmax curves.

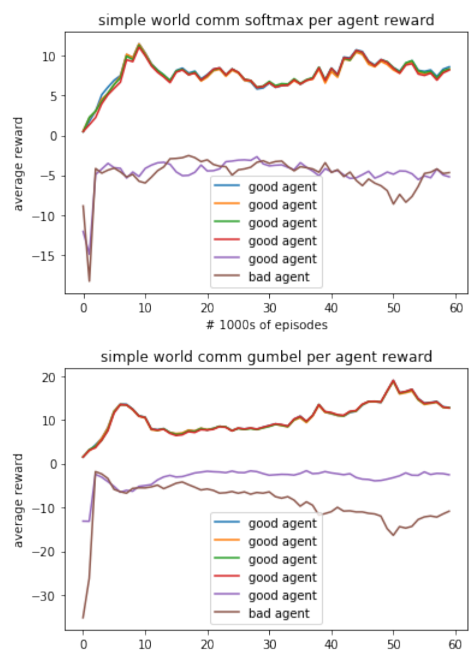


Figure 12. A comparison of softmax (top) and gumbel (bottom) per agent training curves on the world comm environment. In general, we see more fluctuation in the softmax curves.

	crypto	**speaker listener**	**tag**	**world comm**	***comm*plex speaker listener**	***comm*plex tag**
Softmax	11.588	**-33.271**	**20.718**	23.652	**-14.917**	16.644
Gumbel-Softmax	**15.773**	-54.740	12.441	**27.294**	-16.829	**17.295**

Table 2. Final rewards for different action and communication parameterizations. We see softmax tends to perform marginally better overall, winning by larger margins and losing by smaller ones.

We also look at the issue of discrete communication and actions, modifying the code to obtain one hot vectors from instead of a soft approximation as was the paper's original goal. Under this setting, MADDPG fails to converge for every environment in our testing suite. One possible explanation is that it is more difficult for the agents to learn without having soft vectors to communicate uncertainty. It also might be because of some sort of degeneracy in the communication which prevents exploration and diversity in messages.

4. Conclusion

In this work, we explore at multi-agent RL communication, specifically considering complex environments with adversaries. We primarily focus on the MADDPG algorithm, analyzing its properties on four existing environments along with two novel ones. These two novel environments provide cases where communication is required for good performance, but is not directly rewarded, presenting a non-trivial challenge which is not present in existing environments. Lastly, we look at how MADDPG parameterizes its discrete action and communication space, showing both theoretical and practical improvement.

In Section 3.2, we extend the baselines in the paper to more complex and meaningful environments. In our speaker-listener environment, we explore the impact of inducing noise in the adversary's channel, and find the speaker and listener modify their behavior as the utility of public communication changes. However, our main conclusion is that MADDPG is woefully ineffective for learning encryption, even when training on easier tasks first.

In our predator-prey environment, we demonstrate the emergence of natural pack hunting behavior. We also observe some of the limitations of MADDPG, including its inability to utilize more complex communication schemes and to effectively use communication to fully bridge the gap created by introducing partial observability.

Finally, we analyze the parameterizations for actions and communication. We find the use of Gumbel-Softmax is unwarranted given there is no derivative through the sampling, and show softmax is a more principled alternative which often performs better. We also show a discrepancy between the paper and the code, finding the communication and action spaces are not discrete. Lastly, we find MADDPG fails to work with true discrete communication and actions.

After contacting the authors of MADDPG, we received confirmation that these points were correct. They updated their paper to more clearly explain why they chose Gumbel-Softmax and indicate their communication is not actually discrete.

4.1. Future Work

This work provides a thorough analysis of state-of-the-art work in communication in collaborative and competitive environments. Based on our findings, we believe there are several interesting directions to address the many issues we observe.

First, we believe more thorough testing is necessary to determine if MADDPG can solve covert communication. Abadi and Andersen (2016) used a variety of tricks to encourage learning in their cryptography environment, including alternating speaker/listener and adversary training, and using more complex neural networks with convolutional layers.

Additionally, we believe future research is needed on the discrete parameterization of communication. As our work shows, MADDPG does not allow for discrete communication. Directly modifying MADDPG to use discrete communication leads to divergence. We believe deeper changes to MADDPG which allow for differentiablility through the observation channel, such as flowing gradients through communication as in Foerster et al. (2016), might help MADDPG converge. We also suggest researchers explore various differentiable parameterizations, including Gumbel-Softmax and REBAR gradients (Jang et al., 2016; Tucker et al., 2017).

References

Abadi, M. and Andersen, D. G. Learning to protect communications with adversarial neural cryptography. *CoRR*, abs/1610.06918, 2016. URL `http://arxiv.org/abs/1610.06918`.

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

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. *ArXiv*, abs/1611.01144, 2016.

Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., and Mordatch, I. Multi-agent actor-critic for mixed cooperative-competitive environments. *Neural Information Processing Systems (NIPS)*, 2017.

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

Nguyen, T. T., Nguyen, N. D., and Nahavandi, S. Deep reinforcement learning for multi-agent systems: A review of challenges, solutions and applications, 2018.

Tucker, G., Mnih, A., Maddison, C. J., Lawson, J., and Sohl-Dickstein, J. Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models. In *NIPS*, 2017.

Zaïem, M. S. and Bennequin, E. Learning to communicate in multi-agent reinforcement learning : A review, 2019.