Learning to Ground Multi-Agent Communication with Autoencoders

Anonymous Author(s)
Affiliation
Address
email

Abstract

Communication requires having a common language, a lingua franca, between agents. This language could emerge via a consensus process between agents, but this may require many generations of trial and error. Alternatively, the lingua franca can be given by the environment, where agents ground their language in representations of the observed world. We demonstrate a simple way to ground language in learned representations, which facilitates decentralized multi-agent communication and coordination. We find that a standard representation learning algorithm – autoencoding – is sufficient for arriving at a grounded common language. When agents broadcast these representations, they learn to understand and respond to each other's utterances, and achieve surprisingly strong task performance across a variety of multi-agent communication environments.

1 Introduction

The emergence of language was a defining moment in human evolution [30], starting from which human societies began developing into much more sophisticated forms. The study of communication in multi-agent environments is critical for the same reason: it opens up new ways for agents to develop collective intelligence, improving their coordination and cooperation for a much wider range of tasks.

An essential aspect of communication is that each pair of speaker and listener must share a common understanding of the symbols being used before any meaningful conversation can take place. For artificial agents interacting in an environment, with a communication channel but without an agreed upon communication protocol, this raises the question: how can meaningful communication emerge as agents try to maximize their utilities? The communication model that most closely resembles language learning in nature would be a fully decentralized model, where the policies of agents are independently optimized. However, such models perform poorly even in simple communication tasks [13] or with additional inductive biases [7]. As a result, previous works have resorted to using centralized learning or differentiable communication to achieve success in communication tasks [5, 10, 13, 23, 27, 36], making them less applicable to real-world multi-agent coordination problems.

We tackle this challenge by first making the following observations on why emergent communication is difficult in a decentralized multi-agent reinforcement learning setting. A key problem that prevents agents from learning meaningful communication is the lack of a common grounding in communication symbols [2, 7, 14]. In nature, the emergence of a common language is thought to be aided by physical biases and embodiment [24] – we can only produce certain vocalizations, these sounds only can be heard a certain distance away, these sounds bear similarity to natural sounds in the environment, etc – artificial communication protocols are not a priori grounded in aspects of the environment dynamics. This poses a severe exploration problem as the chances of a consistent protocol being found and

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

rewarded is extremely small [13]. Moreover, before a communication protocol is found, the random utterances transmitted between agents add to the already high variance of multi-agent reinforcement learning, making the learning problem even more challenging [7, 23].

To overcome the grounding problem, an important question to ask is: do agents really need to learn the grounding from scratch through random exploration in an environment where success is determined by chance? Perhaps nature has a different answer; previous studies in cognitive science and evolutionary linguistics [15, 31, 34, 35] have provided evidence for the hypothesis that communication first started from sounds whose meaning are grounded in the physical environment, then creatures adapted to make sense of those sounds and make use of them. Inspired by language learning in natural species, we propose a novel framework for grounding multi-agent communication: first ground *speaking* through learned representations of the world, then learn *listening* to interpret these grounded utterances. Surprisingly, even with the simple representation learning task of autoencoding, our approach eases the learning of communication in fully decentralized multi-agent settings and greatly improves agents' performance in multi-agent coordination tasks that are nearly unsolvable without communication.

The contribution of our work can be summarized as follows:

- We formulate communication grounding as a representation learning problem and propose to use observation autoencoding to learn a common grounding across all agents.
- We experimentally validate that this approach is an effective approach for learning decentralized communication in MARL settings: a communication model trained with a simple autoencoder can consistently outperform baselines across various MARL environments.
- We highlight a need to rethink how to address the lack of visual grounding in communication policies, where this work serves as a first step.

2 Related Work

In multi-agent reinforcement learning (MARL), achieving successful emergent communication with decentralized training and non-differentiable communication channel is an important yet challenging task that hasn't been satisfactorily addressed by existing works. Due to the non-stationary and non-Markovian transition dynamics in multi-agent settings, straightforward implementation of standard reinforcement learning methods such as Actor-Critic [17] and DQN [26] perform poorly [13, 23].

Centralized learning is often used to alleviate the problem of high variance in MARL, for example learning a centralized value function that has access to the joint observation of all agents [10, 23]. However, it turns out that MARL models are unable to solve tasks that rely on emergent communication, even with centralized learning and shared policy parameters across the agents [13]. Eccles et al. [7] provides an analysis that illustrates how MARL with communication poses a more difficult exploration problem than standard MARL, which is confirmed by empirical results in [13, 23]: communication exacerbates the sparse reward and high variance in MARL.

Many works therefore resort to differentiable communication [5, 13, 23, 27, 36], where agents are allowed to directly optimize each other's communication policies through gradients. However, this approach imposes a strong constraint on the nature of communication, which limits its applicability to many real-world multi-agent coordination tasks.

Jaques et al. [16] proposes a method that allows independently trained agents to communicate and coordinate. However, the proposed method requires that agent either has access to policies of other agents or stays in close proximity with other agents. These constraints make it difficult for the same method to be applied to a wider range of tasks, such as those in which agents are not embodied or do not observe others directly. Eccles et al. [7] attempts to solve the same issue by introducing inductive biases for positive signaling and positive listening, but implementation requires numerous task-specific hyperparameter tuning, and the effectiveness is limited.

It is also worth noting that, while a large number of existing works on multi-agent communication take structured state information as input [3, 11, 12, 13, 27, 29], we train agents to learn a communication protocol directly from raw pixel observations. This presents additional challenges due to the unstructured and ungrounded nature of pixel data, as shown in [2, 5, 19]. To our knowledge, this

work is the first to effectively use representation learning to aid communication learning from pixel inputs in a wide range of MARL task settings.

3 Preliminaries

We model **multi-agent reinforcement learning (MARL) with communication** as a partially-observable general-sum Markov game [21, 33], where each agent can broadcast information to a shared communication channel. Every time step, each agent receives a partial observation of the underlying world state, including all information communicated in the shared channel. This observation is used to learn an appropriate policy that maximizes the agent's environment reward. In this work, we use parameterize the policy function using a deep neural network.

Formally, a decentralized MARL can be expressed as a partially observable Markov decision process as $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{C}, \mathcal{O}, \mathcal{T}, R, \gamma \rangle$, where N is the number of agents, $\mathcal{S}$ is the set of states, $\mathcal{A} = (\mathcal{A}^1, ..., \mathcal{A}^N)$ is the set of action spaces, $\mathcal{C} = (\mathcal{C}^1, ..., \mathcal{C}^N)$ is the set of communication spaces, and $\mathcal{O} = (\mathcal{O}^1, ..., \mathcal{O}^N)$ is the set of observation spaces.

At time step t, an agent k observes a partial view o_t^k of the underlying true state s_t , and a set of communicated messages from the previous time step $c_{t-1} = (c_{t-1}^1, ..., c_{t-1}^N)$. The agent then chooses an action $a_t^k \in \mathcal{A}^k$ and a subsequent message to broadcast $c_t^k \in \mathcal{C}^k$. Given the joint actions of all N agents $a_t = (a_t^1, ..., a_t^N) \in (\mathcal{A}^1, ..., \mathcal{A}^N)$, the transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A}^1 \times ... \times \mathcal{A}^N \to \Delta(\mathcal{S})$ maps the current state s_t and set of agent actions a_t to a distribution over the next state s_{t+1}. Since the transition function $\mathcal{T}$ is non-determinstic, we denote the probability distributions over $\mathcal{S}$ as $\Delta(\mathcal{S})$. Finally, each agent receives an individual reward $r_t^k \in R(s_t, a_t)$ where $R : \mathcal{S} \times \mathcal{A}^1 \times ... \times \mathcal{A}^N \to \mathbb{R}$.

In our work, we consider a fully cooperative setting in which the objective of each agent is to maximize the total expected return of all agents:

$$\underset{\pi:\mathcal{S}\to\mathcal{A}\times\mathcal{C}}{\text{maximize}} \quad \mathbb{E}\Big[\sum_{t\in T} \sum_{k\in N} \gamma^t R(s_t, a_t) \;\Big|\; (a_t, c_t) \sim \pi, s_t \sim \mathcal{T}(s_{t-1}) \Big] \tag{1}$$

for some finite time horizon T and discount factor γ.

In MARL, the aforementioned objective function is optimized using policy gradient. Specifically, we use asynchronous advantage actor-critic (A3C) [25] with Generalized Advantage Estimation [32] to optimize our policy network. The policy network in A3C outputs a distribution over the actions and the discounted future returns. Any other policy optimization algorithms could have been used in lieu of A3C. We do not use centralized training or self-play, and only consider decentralized training where each agent is parameterized by an independent policy.

4 Grounding Representation for Communication with Autoencoders

The main challenge of learning to communicate in fully decentralized MARL settings is that there is no grounded information to which agents can associate their symbolic utterances. This lack of grounding creates a dissonance across agents and poses a difficult exploration problem. Ultimately, the gradient signals received by agents are therefore largely inconsistent. As the time horizon, communication space, and the number of agents grow, this grounding problem becomes even more pronounced. This difficulty has been highlighted in numerous prior works, with empirical results showing that agents often fail to use the communication channel at all during decentralized learning [2, 7, 14, 23].

To mitigate this issue, we propose a simple yet surprisingly effective approach: using a self-supervised representation learning task to learn a common grounding across all agents. Specifically, we train each agent to independently learn to auto-encode its own observation and use the learned representation for communication. This approach offers the benefits of allowing fully decentralized training without needing additional architectural bias or supervision signal. In Section 5, we show the effectiveness of our approach on a variety of MARL communication tasks.

An overview of our method is provided in Figure 1, which illustrates the communication flow among agents at some arbitrary time step t. All agents share the same individual model architecture, and

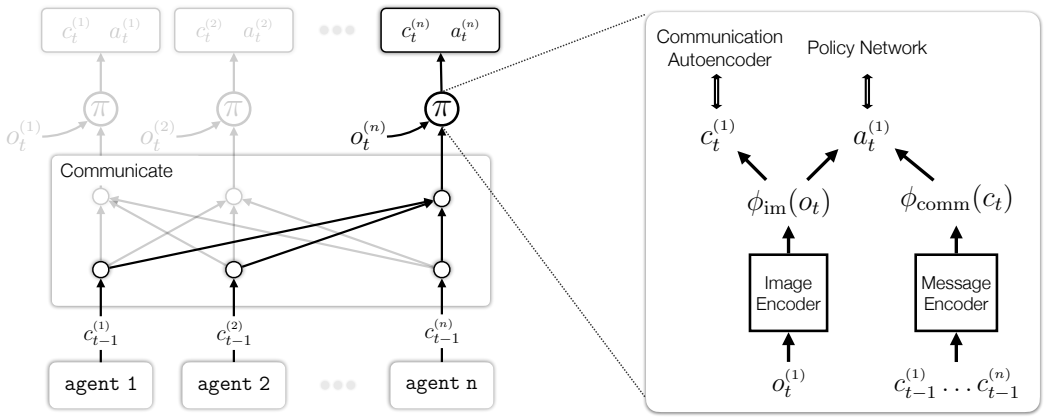


Figure 1: **Overview:** The overall schematic of our multi-agent system that uses autoencoding to ground communication. Each agent broadcasts communication messages to all agents at each environment step. The broadcasted messages are processed through a Message Encoder and concatenated with the image features to predict the next action. The image features are also used to generate the communication messages from the Communication Autoencoder.

each agent consists of two modules: a **speaker module** and a **listener module**. We describe the architecture details of a single agent k below.

4.1 Speaker Module

At each time step t, the speaker module takes in the agent's observation o_t^k and outputs the agent's next communication message c_t^k.

Image Encoder Given the raw pixel observation, the module first uses a image encoder to embed the pixels into a low-dimensional feature $o_t^k \rightarrow \phi_{\text{im}}(o_t^k) \in \mathbb{R}^{128}$. The image encoder is a convolutional neural network with 4 convolutional layers, and the output of this network is spatially pooled.

Communication Autoencoder The goal of the communication autoencoder is to take the current state observation and generate the next subsequent message. We use an autoencoder to learn a mapping from $\phi_{\text{im}}(o_t^k) \rightarrow c_t^k$. The autoencoder consists of an encoder and a decoder, both parameterized by a 3-layer MLP. The decoder tries to reconstruct the input state from the communication message $c_t^k \rightarrow \hat{\phi}_{\text{im}}(o_t^k)$. The communication messages are quantized before being passed through the decoder. We use a straight-through estimator to differentiate through the quantization []. The auxiliary objective function of our model is to minimize the reconstruction loss $\|\phi_{\text{im}}(o_t^k) - \hat{\phi}_{\text{im}}(o_t^k)\|_2^2$. This loss is optimized jointly with the policy gradient loss from the listener module.

4.2 Listener Module

While the goal of the speaker module is to output grounded communication based on the agent's private observation o_t^k, the goal of the listener module is to learn an optimal action policy based on both the observation o_t^k and communicated messages c_{t-1}. At each time step t, the listener module outputs the agent's next action a_t^k.

Message Encoder The message encoder linearly projects all messages communicated from the previous time step c_{t-1} using a shared embedding layer. The information across all agent message embeddings are combined through concatenation and passed through 4-layer MLP. The resulting message feature has a fixed dimension of 128, i.e. $\phi_{\text{comm}}(c_t) \in \mathbb{R}^{128}$.

Policy Network Each agent uses an independent policy head, which is a standard GRU [6] policy with a linear layer. The GRU policy concatenates the encoded image features and the message features $\phi = \phi_{\text{im}}(o_t^k) \circ \phi_{\text{comm}}(c_t)$, and predicts a distribution over the actions $a \sim \pi(\phi)$ and the corresponding expected returns. The predicted action distribution and expected returns are used for computing the policy gradient loss. This loss is jointly optimized with the autoencoder reconstruction loss from the speaker module.

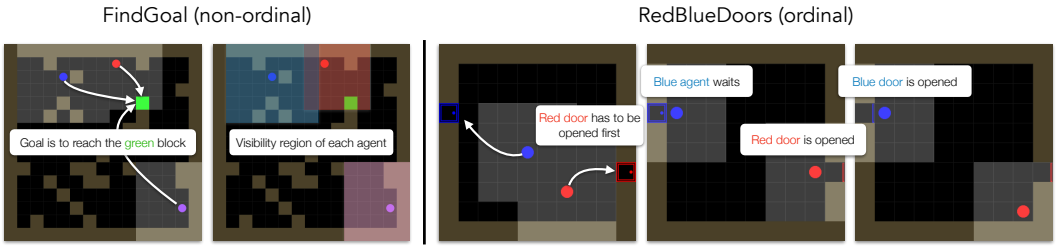


Figure 2: **MarlGrid Environment:** We introduce two new grid environments: `FindGoal` (left) and `RedBlueDoors` (right). These environments are adapted from the GridWorld environment [4, 28]. Environment states are randomized at every episode and are partially observable to the agents. In `FindGoal`, the task is to reach the green goal location. Each agent receives a reward of 1 when they reach the goal, and an additional reward of 1 when all 3 agents reach the goal within the time frame. In `RedBlueDoors`, the task is ordinal, where the ordering of actions matter. A reward of 1 is given to both agents if and only if the red door is opened first and then the blue door.

The same setup is used for all experiments. The exact details of the network architecture and the corresponding training details are in the Appendix.

5 Experiments

In this section, we demonstrate that autoencoding is a simple yet effective representation learning algorithm to ground communication in MARL. We evaluate our method on a various suite of multi-agent environments and qualitatively show that our method outperforms baseline methods. We then provide further ablations and analyses on the learned communication. Code for the environments as well as the experiments will be released.

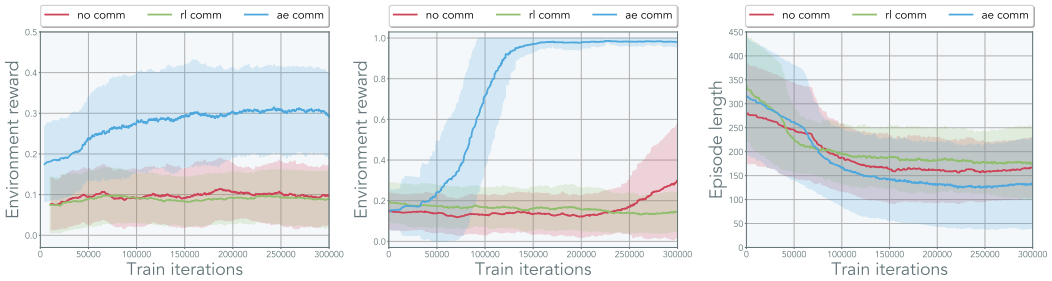
5.1 Environments

We introduce three multi-agent communication environments: `CIFAR Game`, `FindGoal`, and `RedBlueDoors`. In this paper, we focus on cooperative scenarios, though these environments can also be extended to competitive or mixed scenarios.

Our environments cover a wide range of communication task settings, including (1) referential or non-referential, (2) ordinal or non-ordinal, and (3) two-agent versus generalized multi-agent. A referential game, often credited to Lewis signaling game [20], refers to a setup in which agents communicate through a series of message exchanges to solve a task. In contrast to non-referential games, constructing a communication protocol is critical to solving the task – where one can only arrive at a solution through communication. Referential games are referred to as a grounded learning environment, and therefore, communication in MARL have been studied mainly through the lens of referential games [9, 19]. The difference between ordinal and non-ordinal settings is illustrated in Figure 2. We now describe the environments used in our work in more detail:

CIFAR Game We design `CIFAR game` following the setup of `Multi-Step MNIST Game` in [13], but with `CIFAR-10` dataset [18] instead. This is a non-ordinal, two-agent, referential game. In `CIFAR game`, each agent independently observes a randomly drawn image from the `CIFAR-10` dataset, and the goal is to communicate the observed image to the other agent within 5 environment time steps. At each time step, each agent broadcasts a set of communication symbols of length l. At the final time step, each agent must choose a class label from the 10 possible choices. At the end of the episode, an agents receives a reward of 0.5 for each correctly guessed class label, and both agent receives a reward of 1 only when both images are classified correctly.

MarlGrid Environments The second and third environments we consider are: `FindGoal` (Figure 2 left) and `RedBlueDoors` (Figure 2 right). Both environments are adapted from the GridWorld environment [4, 28] and environment states are randomized at every episode.



(a) `CIFAR dialogue` ↑ (b) `RedBlueDoors` ↑ (c) `FindGoal` ↓

Figure 3: **Results with communication grounding:** Comparison between our method that uses an autoencoded communication (`ae-comm`), a baseline that is trained without communication (`no-comm`) and another baseline where communication policy is trained using reinforcement learning (`rl-comm`). For `FindGoal`, we visualize the amount of time it takes for all agents to reach the goal, as all methods can reach the goal within the time frame.

`FindGoal` is a non-ordinal, multi-agent, non-referential game. We use $N = 3$ agents, and at each time step, each agent observes a partial view of the environment centered at its current position. The task of agents is to reach the green goal location as fast as possible. Each agent receives an individual reward of 1 for completing the task and an additional reward of 1 when all agents have reached the goal. Hence, the optimal strategy of an agent is to communicate the goal location once it observes the goal. If all agents learn a sufficiently optimized search algorithm, they can maximize their reward without communication.

`RedBlueDoors` is an ordinal, two-agent, non-referential game. The environment consists of a red door and a blue door, both initially closed. The task of agents is to open both doors, but unlike in the previous two games, the ordering of actions executed by agents matters. A reward of 1 is given to both agents if and only if the red door is opened first and then the blue door. This means that any time the blue door is opened first, both agents receive a reward of 0, and the episode ends immediately. Hence, the optimal strategy for agents is to convey the information that the red door was opened. Since it is possible to solve the task through visual observation or by a single agent that opens both doors, communication is not necessary.

Compared to `CIFAR Game`, the MarlGrid environments have a higher-dimensional observation space and a more complex action space. The fact that these environments are non-referential exacerbates the visual-language grounding problem since communication can only exist in the form of *cheap talk* (i.e., costless communication that has no direct effect on the game state and agent payoffs). We hope to show from this set of environments that autoencoders can be used as a surprisingly simple and adaptable representation learning task to ground communication. It requires little effort to implement and almost no change across environments. Most importantly, as we will see in Section 5.4, autoencoded representation shows an impressive improvement over communication trained with reinforcement learning.

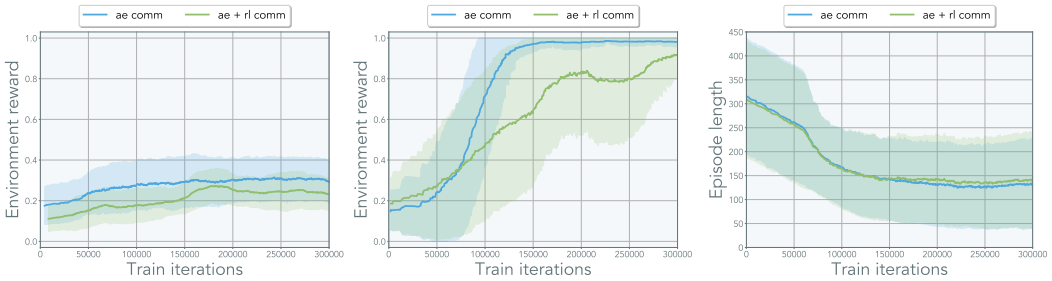
5.2 Baselines

To evaluate the effectiveness of grounded communication, we compare our method (`ae-comm`) against the following baselines: (1) a `no-comm` baseline, which stands for no communication; (2) a `rl-comm` baseline, where communication policy is learned by an additional policy network in listener module instead of the speaker module; (3) a `ae-rl-comm` baseline, where communication policy is learned by an additional policy network trained on top of the autoencoded representation in speaker module.

5.3 The Effectiveness of Grounded Communication

In Figure 3, we compare task performance of `ae-comm` agents with performance of baseline agents.

In `CIFAR Game` environment, both `no-comm` and `rl-comm` baselines could only obtain an average reward close to that of random guesses throughout the training process. In comparison, `ae-comm` agents achieve a much higher reward on average. Since this environment is a referential game, our



(a) `CIFAR dialogue` ↑ (b) `RedBlueDoors` ↑ (c) `FindGoal` ↓

Figure 4: **Representation learning with reinforcement learning:** Comparison between a speaker module trained with only an autoencoding task (`ae-comm`) and another one trained with both autoencoding task and reinforcement learning (`ae-rl-comm`). We observed that further training a policy on top of the autoencoder representation degrades performance across all environments.

results directly indicate that `ae-comm` agents learn to communicate more effectively than the baseline agents.

`FindGoal` environment poses a challenging multi-agent coordination problem since the reward is extremely sparse. As shown in Figure 3(b), neither of the baseline agents was able to learn a successful coordination strategy. In contrast, `ae-comm` agents converge to an optimal strategy after 150k of training.

In `RedBlueDoors` environment, agents are able to solve the task without communication, but their performance can be improved with communication. Therefore, we use episode length instead of reward as the performance metric for this environment. While all agents are able to obtain full rewards, Figure 3(c) shows that `ae-comm` agents are able to complete the episode much faster than other agents. We further verify that this improvement is indeed a result of the successful communication by providing further analysis in Section 5.5.

Our results indicate that a communication model trained with autoencoding tasks consistently outperforms the baselines across all environments. The observation that communication does not work well with reinforcement is consistent with observations made in prior works [7, 13, 23]. Furthermore, our results with autoencoders – a task that is often considered trivial – highlight that we as a community may have overlooked a critical representation learning component in MARL communication.

5.4 The Role of Autoencoding

Given the success of agents trained with autoencoders, it is natural to ask whether a better communication protocol can emerge from the speaker module by jointly training it with a reinforcement learning policy. To this end, we train a GRU policy on top of the autoencoded representation (`ae-rl-comm`) and compare it against our previous model that was trained just with an autoencoder (`ae-comm`). The communication policy head is independent of the environment action policy head.

Surprisingly, we observed in Figure 4 that the model trained jointly with reinforcement learning consistently performed worse. We hypothesize that the lack of correlation between visual observation and the communication rewards hurts the agents' performance. This lack of visual-reward grounding could introduce conflicting gradient updates to the action policy and thereby exacerbate the high-variance problem that already exists in reinforcement learning. Our observation suggests that specialized reward design and training at the level of [1] might be required for decentralized MARL communication. This prompts us to rethink how to address the lack of visual grounding in communication policy, where this work serves as a first step in this direction.

5.5 Analyzing the Effects of Communication Signals on Agent Behavior

Communication Embedding To analyze whether the agents have learned a meaningful visual grounding for communication, we first visualize the communication embedding. In Figure 5, we

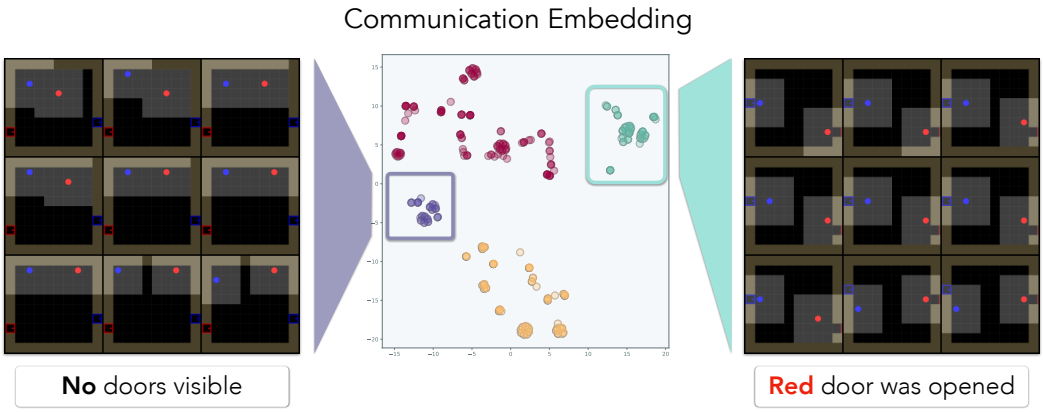


Figure 5: **Communication clusters:** 4096 communication messages are embedded into low-dimensional representation using t-SNE [37] and is clustered using DBSCAN [8]. We visualize the images corresponding to the communication messages. We observed that the message clusters correspond to various meaningful phases throughout the task. The communication symbol of the purple cluster corresponds to when no doors are visible by either agent, and the light green cluster corresponds to when the red door is opened.

visualize the communication symbols transmitted by the agents trained on `RedBlueDoors`. The communication symbols are discrete with a length of $l = 10$ (1024 possible embedding choices), and we use approximately 4096 communication samples across 10 episodic runs. We embed the communication symbols using t-SNE [38] and further cluster them using DBSCAN [8]. In the figure, we visualize clusters by observing the correspondence between image states and communication symbols produced by agents in those states. For example, we observed that a specific communication cluster corresponded to an environment state when the red door was opened; this suggests a communication action where one agent signals the other agent to open the blue door and complete the task.

Positive Signaling and Positive Listening We informally investigate the two metrics suggested by [22] for measuring effectiveness of communication, *positive signaling* and *positive listening*. First, we examine the reward curve with respect to the representation learning task loss. Since `ae-comm` agents have to communicate their learned representation, the presence of representation learning task loss means that `ae-comm` agents are intrinsically optimized for *positive signaling* (i.e., sending messages that are related with their observation or action). In Figure 6, we observe that agent task performance improves as representation learning task loss decreases, indicating *positive listening* (i.e., communication messages influence the behavior of agents).

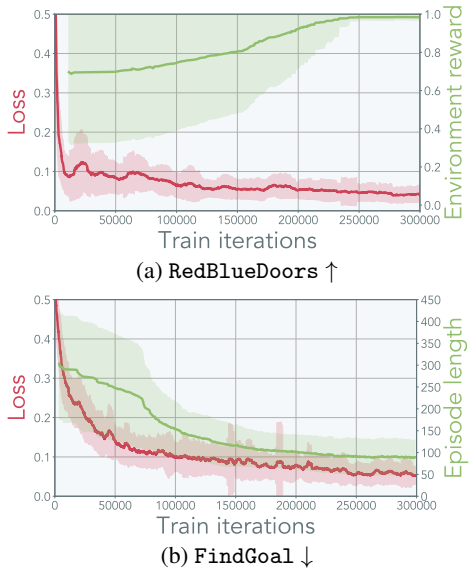


Figure 6: **Performance and representation:** Both agent performance and autoencoder loss improve over the course of learning.

Entropy of Action Distribution To measure whether communicated information directly influences other agents' actions, we visualize the entropy of action distribution during episode rollouts. Suppose one agent shares information that is vital to solving the task. In that case, a decrease in entropy should be observed in the action distributions of other agents, as they act more deterministically towards solving the task.

As shown in Figure 7, we visualize the entropy of action distribution across 256 random episodic runs using policy parameters from a fully trained `ae-comm` model. The entropies are aligned using

(a) `FindGoal` (b) `RedBlueDoors`

Figure 7: **Policy entropy with communication:** We visualize the entropy of the action policy throughout the task (lower is better). The graph is generated with 256 random episodes. For `FindGoal`, the entropy is measured on the last agent to enter the goal, and for `RedBlueDoors` the entropy is measured on the agent that opens the blue door (the second door). All 256 runs are aligned to the *dotted red lines* which corresponds to the time in which the first and second agent enters the goal for (`FindGoal`), and the time in which the red and the blue doors are opened for (`RedBlueDoors`). The model trained with an auto-encoder transmits messages that are effectively used by other agents.

environment milestone events: for `FindGoal`, this is when the first agent reaches the goal; for `RedBlueDoors`, this is when the red door is opened. Since the identity of the agents that solve the task first does not matter, entropy plots are computed with respect to the *listener* agents (i.e., agents that receive vital information from others). In `FindGoal`, this corresponds to the last agent to reach the goal; in `RedBlueDoors`, this corresponds to the agent opening the blue door. For both environments, we see a sharp fall-off in entropy as soon as the first agents finish the task. In contrast, agents trained without autoencoding act randomly regardless of whether other agents have completed the task. This reaffirms that the agents trained with an autoencoder can effectively transmit information to other agents.

6 Discussion and Societal Impacts

We present a framework for grounding multi-agent communication through autoencoding, a simple self-supervised representation learning task. Our method allows agents to learn non-differentiable communication in fully decentralized settings, and does not impose constraints on input structures (e.g. state inputs or pixel inputs) or task nature (e.g. referential or non-referential). Our results demonstrate that, agents trained with the proposed method achieve much better performance on a suite of coordination tasks compared to baselines.

We believe this work on multi-agent communication is of importance to our society for two reasons. First, it extends a computational framework under which scientific inquiries concerning language acquisition, language evolution, and social learning can be made. Second, it opens up new ways for artificial learning agents to improve their coordination and cooperative skills, increasing their reliability and usability when deployed to real-world tasks and interacting with humans.

One limitation of this work, which is also one concern we have regarding its potential negative societal impact, is that the environments we consider are cooperative. If the communication method we present in this work is to be deployed to the real world, we need to either make sure the environment is rid of adversaries, or conduct additional research to come up with robust counter-strategies in the face of adversaries, which could use better communication policies as a way to lie, spread misinformation, or maliciously manipulate other agents.

References

[1] C. Berner, G. Brockman, B. Chan, V. Cheung, P. Dębiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019. 7

[2] B. Bogin, M. Geva, and J. Berant. Emergence of communication in an interactive world with consistent speakers. *arXiv preprint arXiv:1809.00549*, 2018. 1, 2, 3

[3] K. Cao, A. Lazaridou, M. Lanctot, J. Z. Leibo, K. Tuyls, and S. Clark. Emergent communication through negotiation. *arXiv preprint arXiv:1804.03980*, 2018. 2

[4] M. Chevalier-Boisvert, L. Willems, and S. Pal. Minimalistic gridworld environment for openai gym. https://github.com/maximecb/gym-minigrid, 2018. 5

[5] E. Choi, A. Lazaridou, and N. de Freitas. Compositional obverter communication learning from raw visual input. *arXiv preprint arXiv:1804.02341*, 2018. 1, 2

[6] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. 4

[7] T. Eccles, Y. Bachrach, G. Lever, A. Lazaridou, and T. Graepel. Biases for emergent communication in multi-agent reinforcement learning. *arXiv preprint arXiv:1912.05676*, 2019. 1, 2, 3, 7

[8] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996. 8

[9] K. Evtimova, A. Drozdov, D. Kiela, and K. Cho. Emergent communication in a multi-modal, multi-step referential game. *arXiv preprint arXiv:1705.10369*, 2017. 5

[10] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 1, 2

[11] J. Foerster, F. Song, E. Hughes, N. Burch, I. Dunning, S. Whiteson, M. Botvinick, and M. Bowling. Bayesian action decoder for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, pages 1942–1951. PMLR, 2019. 2

[12] J. N. Foerster, Y. M. Assael, N. de Freitas, and S. Whiteson. Learning to communicate to solve riddles with deep distributed recurrent q-networks. *arXiv preprint arXiv:1602.02672*, 2016. 2

[13] J. N. Foerster, Y. M. Assael, N. De Freitas, and S. Whiteson. Learning to communicate with deep multi-agent reinforcement learning. *arXiv preprint arXiv:1605.06676*, 2016. 1, 2, 5, 7

[14] S. Harnad. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1-3):335–346, 1990. 1, 3

[15] J. R. Hurford. Biological evolution of the saussurean sign as a component of the language acquisition device. *Lingua*, 77(2):187–222, 1989. 2

[16] N. Jaques, A. Lazaridou, E. Hughes, C. Gulcehre, P. Ortega, D. Strouse, J. Z. Leibo, and N. De Freitas. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In *International Conference on Machine Learning*, pages 3040–3049. PMLR, 2019. 2

[17] V. R. Konda and J. N. Tsitsiklis. Actor-critic algorithms. In *Advances in neural information processing systems*, pages 1008–1014. Citeseer, 2000. 2

[18] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[19] A. Lazaridou, K. M. Hermann, K. Tuyls, and S. Clark. Emergence of linguistic communication from referential games with symbolic and pixel input. *arXiv preprint arXiv:1804.03984*, 2018. 2, 5

[20] D. Lewis. *Convention: A philosophical study*. John Wiley & Sons, 2008. 5

[21] M. L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*, pages 157–163. Elsevier, 1994. 3

[22] R. Lowe, J. Foerster, Y.-L. Boureau, J. Pineau, and Y. Dauphin. On the pitfalls of measuring emergent communication. *arXiv preprint arXiv:1903.05168*, 2019. 8

[23] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *arXiv preprint arXiv:1706.02275*, 2017. 1, 2, 3, 7

[24] B. MacWhinney. *The emergence of language from embodiment*. Psychology Press, 2013. 1

[25] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937. PMLR, 2016. 3

[26] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013. 2

[27] I. Mordatch and P. Abbeel. Emergence of grounded compositional language in multi-agent populations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 1, 2

[28] K. Ndousse. marlgrid. https://github.com/kandouss/marlgrid, 2020. 5

[29] M. Noukhovitch, T. LaCroix, A. Lazaridou, and A. Courville. Emergent communication under competition. *arXiv preprint arXiv:2101.10276*, 2021. 2

[30] M. A. Nowak and D. C. Krakauer. The evolution of language. *Proceedings of the National Academy of Sciences*, 96(14):8028–8033, 1999. 1

[31] D. K. Roy and A. P. Pentland. Learning words from sights and sounds: A computational model. *Cognitive science*, 26(1):113–146, 2002. 2

[32] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015. 3

[33] L. S. Shapley. Stochastic games. *Proceedings of the national academy of sciences*, 39(10):1095–1100, 1953. 3

[34] L. Steels. The synthetic modeling of language origins. *Evolution of communication*, 1(1):1–34, 1997. 2

[35] L. Steels and F. Kaplan. Aibo's first words: The social learning of language and meaning. *Evolution of communication*, 4(1):3–32, 2000. 2

[36] S. Sukhbaatar, A. Szlam, and R. Fergus. Learning multiagent communication with backpropagation. *arXiv preprint arXiv:1605.07736*, 2016. 1, 2

[37] L. van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. 8

[38] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 8

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]