Semantic Encoders Enable Robust Communication-Aware Reinforcement Learning Policies

Siddharth Srikanth[1], Varun Bhatt[1], Darius Mahjoob[1],
Sophie Hsu[1], Aaquib Tabrez[2], Stefanos Nikolaidis[1],
[1]Thomas Lord Department of Computer Science, University of Southern California
[2]Sibley School of Mechanical and Aerospace Engineering, Cornell University
{ssrikant,vsbhatt,dmahjoob,yachuanh,nikolaid}@usc.edu, aaquibtabrez@cornell.edu

Abstract

Natural language serves as a powerful medium for coordination, information sharing, instruction, and building a theory of mind in teams. However, training agents to interpret such communication often relies on either rigid, templated, or symbolic messages that are not robust, or on large language models (LLMs), which introduce significant inference delays. We address this with a framework to bridge the gap between high-dimensional unrestricted natural language messages and low-dimensional representations suited for training communication-aware reinforcement learning (RL) agents. Our approach follows a two-stage training process: (1) training an encoder on diverse communication logs generated by LLM-powered agents to learn a low-dimensional representation of messages, and (2) integrating this encoder to train RL agents in multi-agent collaboration scenarios. We evaluate our framework in Lunar Lander and Merge, two long-horizon environments, and show improved performance with communication. Furthermore, we show that the trained RL agents are capable of understanding messages even when worded in unseen ways, demonstrating the robustness of our framework.

1 Introduction

Natural language enables humans to share information, adapt plans, and build a theory of mind in collaborative settings, making it ideal for robot teammates to also understand natural language messages. Large language models (LLMs) make it feasible to parse varied phrasing (e.g., “bring me water” vs. “get me a glass of water”) that a user might say. However, in dynamic situations, e.g., “don’t turn left; a child just stepped out”, their longer inference time can slow the decision loop. Moreover, these models are not trained to be optimal or collaborative, especially when interacting with human teammates. On the other hand, reinforcement learning (RL) agents can act faster with smaller policies, but typically lack the capability to understand and act on unconstrained natural language (Luketina et al. 2019). In this work, we address the problem of *training communication-aware RL policies for human-machine teaming* to maintain low-latency decision-making while equipping agents with an understanding of natural language messages.

Agents capable of collaborating with explicit communication have previously been studied in the context of multi-agent reinforcement learning (Lazaridou and Baroni 2020; Zhu, Dastani, and Wang 2024). However, such works focused on symbolic communication, in which agents communicated via symbols that were not necessarily grounded in natural language (Evtimova et al. 2018; Havrylov and Titov 2017; Kottur et al. 2017; Lazaridou et al. 2018). Subsequent works also extended symbolic communication to partially observable domains and showed that communication was key to bridging the information gap between agents (Jaques et al. 2018; Eccles et al. 2019).

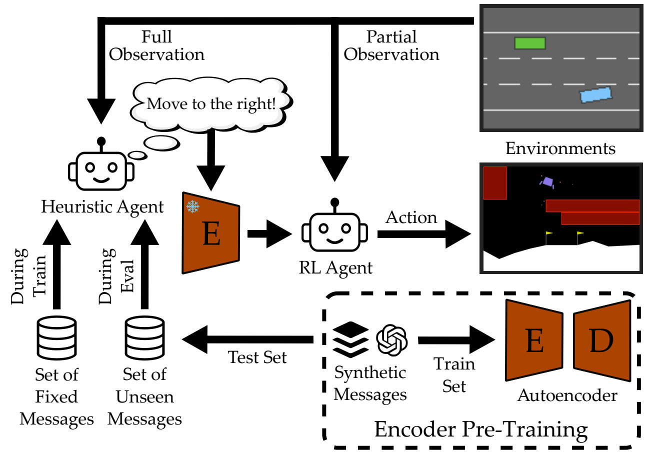

Figure 1: Summary of our proposed framework. We pretrain an autoencoder with synthetic communication data generated by an LLM. Then, we integrate the encoder into RL training to obtain a communication-aware RL policy capable of understanding unseen messages as well.

One way to integrate symbolic and natural language communication is to manually design symbols and train RL policies that can interpret and communicate using them. However, this process is tedious and does not scale well (Tellex et al. 2020; Tabrez, Leonard, and Hayes 2025). Another approach to training such agents would be to collect teaming and communication behaviors through large-scale human-human or human-robot data collection, but this is expensive and challenging (Rogers and Marshall 2017). However, recent work has shown that LLM-powered agents exhibit human-like behavior (Zhou et al. 2024; Li et al. 2023; Xie et al. 2024; Yang et al. 2024; Srikanth et al. 2025), making them a good proxy for humans.

Our key insight is that *by learning to encode synthetic communication data, we can train RL policies capable of understanding messages in natural language*. We achieve this via a two-step process. First, we pre-train an autoencoder on the communication data to obtain an encoder that converts high-dimensional natural language messages to a low-dimensional embedding. Then, we integrate this encoder into RL training to obtain a message-conditioned policy. Our results show that the learned RL policy generalizes robustly to novel, unseen messages, as it was exposed to diverse communication data during training.

2 Method

Stage 1: Learning Low-Dimensional Representations of Natural Language Messages Directly learning an RL policy conditioned on a high-dimensional message input is challenging, as it requires the policy to simultaneously learn a good representation of the message and a good mapping to actions. Hence, we employ a Variational Autoencoder (VAE) (Kingma and Welling 2014) to convert the high-dimensional message inputs to low-dimensional representations that are more suitable as observations to the RL agent. First, we query an LLM to generate a set of diverse phrasings of messages an agent could send in the domain, based on the available actions. Then, we obtain the sentence embeddings, i.e., a high-dimensional representation, of these messages by passing them through Sentence-BERT (Reimers and Gurevych 2019). Finally, we train the VAE with a low-dimensional latent space to encode and reconstruct the sentence embeddings. The diverse messages in its training set enable the VAE to encode incoming natural language messages into their corresponding low-dimensional representation during RL agent execution.

Stage 2: Training Communication-Aware Policies We assume a training setup with a communication-aware RL agent paired with a fixed heuristic agent sending natural language messages selected from a message set. During training, the pre-trained encoder converts the received message to its low-dimensional representation, which the RL agent receives as an additional input along with observations from the environment. We then train the RL policy, now conditioned on both messages and observation, to maximize the discounted return. While our framework makes no assumptions about the RL algorithm, we use Proximal Policy Optimization (PPO) (Schulman et al. 2017) in our experiments.

3 Results

Table 1: Performance comparison across domains with and without communication. Communication, even with unseen wording, results in significant performance improvements. Effect size reported as Cohen's *d*.

Domain	w/ Comm	w/o Comm	Effect Size
Lander	$-\mathbf{3.36}_{0.41}$	$-7.40_{0.46}$	0.466
Merge	$\mathbf{18.17}_{0.58}$	$13.98_{0.75}$	0.312

Domains. We evaluated the communication-aware policies in two domains modified to include communication: Lunar Lander (Brockman et al. 2016) and Merge (Leurent 2018). In the Lunar Lander domain, the objective is to land the lander while avoiding certain areas of space marked as "danger zones", which are not observable to the RL agent and would reduce the RL agent's reward when entering. The heuristic agent provides language instructions on where the danger zones are located to the RL agent. In the Merge domain (Leurent 2018), the RL agent's task is to avoid collisions with merging traffic whose merge intent is unobserved by the RL agent. The heuristic agent indicates to the agent from which side the merging vehicle is approaching.

Results. Table 1 compares the performance of RL agents with and without communication. Adding communication understanding capabilities significantly improves the agent's performance ($p < 0.01$), highlighting the benefits of our framework. By training an autoencoder on diverse communication logs, our framework enables RL agents to infer the latent intent of their partners from unrestricted natural language input.

3.1 Proposed Future Evaluation.

We propose the following future extensions for evaluation of our framework:

Unseen scenarios. We will evaluate our framework on unseen scenarios in both of our domains. For Lunar Lander, this constitutes evaluating on danger zone configurations unseen during training. For the Merge domain, we will modify where the unseen vehicles merge onto the highway.

Multi-agent domain. We propose evaluating our framework on a multi-agent collaborative domain, Overcooked AI (ove 2018; Carroll et al. 2019). We will modify this domain to have both agents communicate with each other in natural language and train them with our framework.

4 Discussion

Future work Here, we outline our proposed improvements and extensions. Prior work has explored training losses to facilitate communication in RL agents (Eccles et al. 2019), and we plan to incorporate similar strategies into our framework. Additionally, we assume that the heuristic agent sends perfect messages. In human-agent teaming scenarios, messages may be noisy or even adversarial (e.g., deceptive or misleading), necessitating mechanisms to detect and filter such inputs. We plan to integrate learnable message filtering in our framework in the future. By leveraging ideas from prior work (e.g., (Strouse et al. 2021)) to generate a population of partners that vary in terms of their messaging behavior, the RL agent can learn a robust filtering mechanism.

Conclusion We present a framework for training RL agents that understand natural language messages. These agents can adapt to unseen messages during evaluation in two long-horizon environments. Such communication-aware RL agents enable effective ToM as they can leverage private knowledge and intent shared by their partners in natural language.

Acknowledgments

This work was supported by NSF CAREER #2145077.

References

2018. Overcooked 2.

Bernstein, D. S.; Givan, R.; Immerman, N.; and Zilberstein, S. 2002. The Complexity of Decentralized Control of Markov Decision Processes. *Math. Oper. Res.*, 27(4): 819–840.

Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; and Zaremba, W. 2016. OpenAI Gym.

Carroll, M.; Shah, R.; Ho, M. K.; Griffiths, T.; Seshia, S.; Abbeel, P.; and Dragan, A. 2019. On the utility of learning about humans for human-ai coordination. *Advances in neural information processing systems*, 32.

Eccles, T.; Bachrach, Y.; Lever, G.; Lazaridou, A.; and Graepel, T. 2019. Biases for Emergent Communication in Multi-agent Reinforcement Learning. In *Advances in Neural Information Processing Systems 32*, 13111–13121.

Evtimova, K.; Drozdov, A.; Kiela, D.; and Cho, K. 2018. Emergent Communication in a Multi-Modal, Multi-Step Referential Game. In *6th International Conference on Learning Representations, ICLR*.

Havrylov, S.; and Titov, I. 2017. Emergence of Language with Multi-agent Games: Learning to Communicate with Sequences of Symbols. In *Advances in Neural Information Processing Systems 30*, 2149–2159.

Huang, S.; Dossa, R. F. J.; Ye, C.; Braga, J.; Chakraborty, D.; Mehta, K.; and Araújo, J. G. 2022. CleanRL: High-quality Single-file Implementations of Deep Reinforcement Learning Algorithms. *Journal of Machine Learning Research*, 23(274): 1–18.

Jaques, N.; Lazaridou, A.; Hughes, E.; Gülçehre, Ç.; Ortega, P. A.; Strouse, D.; Leibo, J. Z.; and de Freitas, N. 2018. Intrinsic Social Motivation via Causal Influence in Multi-Agent RL. *CoRR*, abs/1810.08647.

Kingma, D. P.; and Welling, M. 2014. Auto-Encoding Variational Bayes. *International Conference on Learning Representations*.

Kottur, S.; Moura, J. M. F.; Lee, S.; and Batra, D. 2017. Natural Language Does Not Emerge 'Naturally' in Multi-Agent Dialog. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP*, 2962–2967.

Lazaridou, A.; and Baroni, M. 2020. Emergent Multi-Agent Communication in the Deep Learning Era. *CoRR*, abs/2006.02419.

Lazaridou, A.; Hermann, K. M.; Tuyls, K.; and Clark, S. 2018. Emergence of Linguistic Communication from Referential Games with Symbolic and Pixel Input. In *6th International Conference on Learning Representations, ICLR*.

Leurent, E. 2018. An Environment for Autonomous Driving Decision-Making. https://github.com/eleurent/highway-env.

Li, H.; Chong, Y. Q.; Stepputtis, S.; Campbell, J.; Hughes, D.; Lewis, M.; and Sycara, K. 2023. Theory of mind for multi-agent collaboration via large language models. *arXiv preprint arXiv:2310.10701*.

Luketina, J.; Nardelli, N.; Farquhar, G.; Foerster, J.; Andreas, J.; Grefenstette, E.; Whiteson, S.; and Rocktäschel, T. 2019. A survey of reinforcement learning informed by natural language. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence.

Reimers, N.; and Gurevych, I. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Rogers, Y.; and Marshall, P. 2017. *Research in the Wild*. Morgan & Claypool Publishers.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal Policy Optimization Algorithms.

Srikanth, S.; Bhatt, V.; Zhang, B.; Hager, W.; Lewis, C. M.; Sycara, K. P.; Tabrez, A.; and Nikolaidis, S. 2025. Algorithmic Prompt Generation for Diverse Human-like Teaming and Communication with Large Language Models. arXiv:2504.03991.

Strouse, D.; McKee, K.; Botvinick, M.; Hughes, E.; and Everett, R. 2021. Collaborating with humans without human data. *Advances in Neural Information Processing Systems*, 34: 14502–14515.

Tabrez, A.; Leonard, R.; and Hayes, B. 2025. Single-shot policy explanation to improve task performance via semantic reward coaching. *Neural Computing and Applications*, 1–23.

Tellex, S.; Gopalan, N.; Kress-Gazit, H.; and Matuszek, C. 2020. Robots that use language. *Annual Review of Control, Robotics, and Autonomous Systems*, 3(1): 25–55.

Xie, C.; Chen, C.; Jia, F.; Ye, Z.; Lai, S.; Shu, K.; Gu, J.; Bibi, A.; Hu, Z.; Jurgens, D.; Evans, J.; Torr, P.; Ghanem, B.; and Li, G. 2024. Can Large Language Model Agents Simulate Human Trust Behavior? arXiv:2402.04559.

Yang, Z.; Zhang, Z.; Zheng, Z.; Jiang, Y.; Gan, Z.; Wang, Z.; Ling, Z.; Chen, J.; Ma, M.; Dong, B.; et al. 2024. Oasis: Open agents social interaction simulations on one million agents. *arXiv preprint arXiv:2411.11581*.

Zhou, X.; Zhu, H.; Mathur, L.; Zhang, R.; Yu, H.; Qi, Z.; Morency, L.-P.; Bisk, Y.; Fried, D.; Neubig, G.; and Sap, M. 2024. SOTOPIA: Interactive Evaluation for Social Intelligence in Language Agents. arXiv:2310.11667.

Zhu, C.; Dastani, M.; and Wang, S. 2024. A survey of multi-agent deep reinforcement learning with communication. *Autonomous Agents and Multi-Agent Systems*, 38(1): 4.

A Domains

We address the problem of training communication-aware RL policies in collaborative sequential decision-making environments. We formulate the environment as a decentralized Partially Observable Markov Decision Process (dec-POMDP (Bernstein et al. 2002)) $\langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, O, \gamma \rangle$ with N agents, where $\mathcal{S}$ is the state space, $\mathcal{A} = \Pi_i^N \mathcal{A}_i$ is the joint action space of all agents, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the common reward function that all agents receive, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ is the transition function, O is the observation function, and γ is the discount factor. The agents' goal is to maximize the discounted sum of rewards, $J = \Sigma_t \gamma^t r_t$, where r_t is the reward obtained at timestep t.

A.1 Lunar Lander

Each policy is trained for 5,000,000 timesteps, with a maximum episode duration of 600 timesteps. The dimension of the message vector is 2.

A.2 Merge

Each policy was trained for 200,000 timesteps, with a maximum episode duration of 256 timesteps. The dimension of the message vector is 32.

B Algorithm

B.1 Pseudocode

We train the RL agent with PPO (Schulman et al. 2017), using the implementation and default hyperparameters from CleanRL (Huang et al. 2022).

Algorithm 1: PPO with Communication

```
Input: POMDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, O, \gamma \rangle$; initial policy
       parameters $\theta_0$; episode horizon $T$; total
       iterations $N$
Output: Trained policy $\pi_\theta$
$\theta \leftarrow \theta_0$
for $i \in \{1 \dots N\}$ do
    Get initial state $s_0$
    $o_0 \leftarrow O(s_0)$
    for $t \in \{0 \dots T\}$ do                // Rollout
        $m_t \leftarrow \text{heuristic\_agent\_message}(o_t)$
        $\hat{m}_t \leftarrow \text{encoder}(m_t)$
        $\tilde{o}_t \leftarrow [\, o_t \,;\, \hat{m}_t \,]$      // Concat message
        $a_t \sim \pi_\theta(a_t | \tilde{o}_t)$
        $r_t \sim \mathcal{R}(s_t, a_t)$
        $s_{t+1} \sim \mathcal{P}(s_t, a_t)$
        $o_{t+1} \leftarrow O(s_{t+1})$
    end
    Update $\theta$ with PPO
end
```