# Extended Abstract

**Motivation** Multi-agent path planning plays a critical role in robotics applications such as drone coordination, warehouse logistics, and search-and-rescue operations. While prior work has demonstrated that inter-agent communication can improve performance, direct agent-to-agent messaging is not always feasible—especially in domains with limited bandwidth, stealth requirements, or decentralized constraints. Inspired by biological systems and the principle of stigmergy, we investigate whether communication via the environment (rather than between agents directly) can yield similar benefits. Our research question is: Can path planning performance in cooperative multi-agent systems be improved using agent-environment-agent communication instead of direct agent-to-agent communication?

**Method** We explore this question using the Simple Spread environment from the PettingZoo MARL benchmark. The task requires N agents to reach N landmarks cooperatively. We propose a mechanism where agents share minimal information—specifically, the location of their nearest landmark—not by direct messaging but by writing to and reading from the environment state. This form of communication is centralized through the environment itself, avoiding the need for bandwidth-heavy or explicit agent-to-agent signaling. Our learning framework is based on Proximal Policy Optimization (PPO), selected for its stability and effectiveness in multi-agent settings.

**Implementation** Our architecture consists of decentralized PPO agents that optionally receive augmented observations containing communication vectors broadcast through the environment. The observation space includes agent position, velocity, and relative landmark/agent positions. If communication is enabled, agents also receive other agents' nearest-landmark positions. The PPO implementation uses shared policy networks across agents, with separate training per agent to preserve decentralization. We ablate across three factors: (1) number of agents, (2) size of the critic network, and (3) presence or absence of communication. Each training epoch includes three episodes of 40 steps each, with actor and critic updated twice per batch.

**Results** Our quantitative experiments compare agent performance in terms of average return and success rate per epoch, with and without communication. We observe that agent-environment-agent communication yields modest but consistent gains—especially in setups with 10 agents and larger critic networks. These configurations showed improved success rates and higher average returns compared to their no-communication counterparts. However, when using smaller critic networks or fewer agents, performance improvements diminished and in some cases reversed due to potential overfitting.

**Discussion** The results suggest that while indirect communication can enhance performance, its benefits depend on the model capacity and task complexity. Larger teams benefit more from communication, possibly because the environment becomes harder to coordinate without shared cues. However, as input dimensionality increases with communication, smaller networks may experience an information bottleneck, leading to degraded performance. Additionally, tuning models under multiple ablation dimensions was time-intensive, and we suspect further improvements may be possible by refining the actor network or adopting learned communication policies.

**Conclusion** This work demonstrates that indirect communication through the environment can improve multi-agent path planning performance in cooperative tasks, even in the absence of explicit inter-agent messaging. While the performance gains are not dramatic, they are consistent and point toward a promising, low-bandwidth alternative to traditional communication strategies. Future work will explore learned communication mechanisms, actor network tuning, and broader generalization across different multi-agent environments.

# Improving Multi-Agent Path Planning via Indirect Communication in Cooperative Tasks

**Abhinav Shaw**
Department of Computer Science
Stanford University
abshaw@stanford.edu

## Abstract

This paper investigates whether communication can improve path planning performance in multi-agent systems under the constraint of no direct agent-to-agent communication. Using the Simple Spread environment, we implement a communication mechanism where agents share information via the environment, inspired by stigmergy. We train agents using Proximal Policy Optimization (PPO) and compare performance across multiple configurations, including varying critic network sizes and agent counts. Our experiments show modest but consistent improvements in success rate and return when communication is enabled. The results suggest potential for future work in learned communication and actor network tuning.

## 1 Introduction

Multi Agent Systems has been a hot field for research and engineering in a the past decade. In the recent years after the advancements in Language Modeling and Reinforcement Learning we have seen a significant increase in interest in Multi Agent Systems. There are several techniques that can be used to design and train learned systems for Multi Agent Systems. The environments for Multi Agent tasks can be broadly categorized into the following **fully cooperative, fully competitive and mixed**. Reinforced Learning is emerging as one of the most promising techniques to learn agents that can tackle these environments. Orthogonally path planning has been one of the most challenging robotics problem with its own rich literature ranging from optimal control, discretized planning to distributed planning. As the field matures new avenues for combining techniques and solving problems becomes possible. While researchers have studied multi agent cooperative tasks in the context of improvement and emergent communication we couldn't find any literature that tackles multi agent cooperative tasks with a focus of path planning and agent-env-agent communication.

Path planning for multi agents systems sees significant application in distributed robotics with it being a precursor for much more complicated tasks such as search and rescue. Inspired by this potential gap and the limitless application of a system that can perform path planning for multi agent systems this project attempts to address the gap by improving performance for a multi agent cooperative task in the simple spread environment. The simple spread environment is a simplified analogue to a complex real world scenario. We believe if we can improve performance in the simple spread environment the improvements would carry over to similar real world tasks.

Multi agent systems can be complex and often require intricate architectures and design to achieve good performance. Bui et al. (2025) researchers find that limited communication can improve path finding task using agent-agent communication. While agent to agent communication has potential it requires to maintain a communication channel between agents which may or may not be desirable for certain tasks. In particular in defense tasks where the task is to search and destroy, emitting radiation from unmanned UAVs can render them susceptible to detection or anti radiation munitions.

Guided by the application, the research question that we ask is **can we improve the path planning problem with communication by limit ourselves to non agent-agent communication?** Drawing inspiration from Aras et al. (2004) who is in turn inspired by communication in nature which is agent - env - agent communication we attempt to improve the path planning problem through agent - env -agent communication providing a proof of concept in the simple spread environment.

## 2 Related Work

Multi Agent RL is a thriving research sub-domain under RL. With high impact papers like Lowe et al. (2017) in which researchers introduce the centralized training and decentralized execution paradigm of agents for mixed cooperative and competitive tasks. In Rashid et al. (2018) authors revisit centralized vs decentralized training of RL agents proposing QMix, a non linear mixing network to combine agent-specific value estimates into a global Q functions, conditioned on the full state. While our work touches some aspects of centralized vs decentralized training, our project research objective is different from theirs.

Within the sub-domain of Multi Agent RL, there are some researchers who are interested in understanding communication protocols for better performance. Orthogonally, a small section of RL research focuses on bringing new ideas to the field. In Particular, Aras et al. (2004) describe how certain aspects of biological phenomena of stigmergy can be imported into multi-agent reinforcement learning with the purpose of better coordination of agent actions and speeding up learning. We drive motivation for agent-env-agent communication from Stigmergy but apply the concepts from this work in a path planning problem for a multi agent cooperative task.

While some of the research is exploratory, some are interested in finding what is the impact of communication when applied to Multi Agent RL systems. In Foerster et al. (2016) are able to demonstrate demonstrate end to-end learning of communication protocols in complex environments inspired by communication riddles and multi-agent computer vision problems with partial observability. They proposes two approaches for these: Reinforced Inter-Agent learning and differentiable Inter-Agent Learning (DIAL). While the agents communicate through a limited discrete bandwidth (same as the proposed project), the communication is relayed agent to agent which differs from the proposed approach. Further more, the empirical results are shown for environment, i.e. Switch Riddle which has a discrete state and action space and the second environment, i.e. MNIST Color digit and the Grey Scale variant which has a discrete action space but a continuous state space since inputs are images. Not only do these tasks differ from the propose task, they are also lower in dimensionality due to their discrete nature. The proposed project aims to use the simple spread petting zoo environment which is a multi agent path planning problem.

Moreover, Eccles et al. (2019) study the Biases for Emergent Communication in MARL. They propose inductive biases for positive signaling and listening which eases the limitations of adopting communication techniques in multi-agent RL settings without centralized training. They demonstrate the value of these biases empirically on two tasks first, Sum of digits in MNIST which has a continuous state space but a discrete action space as the results are integers and second, Treasure hunt where the agents need to learn to communicate to obtain treasure which, this environment has a discrete state space and the action space wasn't clear either from the paper's main section or the supplementary material. Both these settings differ from our proposed approach with ours being continuous state space, continuous action space and agent to environment to agent communication (broadcaster listener paradigm). Additionally, our task has a general practical application and addresses a path planning problem under communication and cost constraints.

Orthogonally some research groups improve multi agent path planning in a grid based discrete setting by training multiple agents using deep reinforcement learning in Çetinkaya (2021) and take the field in promising direction. Ours differs from this as we use a continuous where the actor network outputs the quantity of force that needs to be applied in each direction which is a continuous value.

## 3 Method

Our main approach is to improve performance on a multi agent cooperative path planning task using communication. This kind of system could be used to control swarms of drones that need to move into certain shapes or formations. For example, in drone light shows, path planning is currently handled using specific algorithms. But path planning for many agents is not just for light shows — it's also important in many areas of distributed robotics.

We use Proximal Policy Optimization Schulman et al. (2017) and online RL algorithm to learn the policy and the critic. The key idea in PPO is to use a clipped surrogate objective instead of a hard KL constraint in Trust Region Policy Optimization Schulman et al. (2015). The goal is to maximize the expected return in (1) with the gradient estimated by (2).

$$J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{T} \gamma^t R_t\right] \tag{1}$$

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a_t \mid s_t) \cdot A_t\right] \tag{2}$$

Although PPO is an online algorithm, there are multiple critic and actor updates for each batch of collected data. As training progresses, the policy being trained can diverge from the policy used to collect that data. To accommodate this, PPO uses Importance Weighting by computing an importance weight as $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$. We then compute the advantage as the current state value subtracted by the sum of discounted rewards which is the Monte Carlo estimate of the Advantage. The Mote Carlo estimate of the advantage is sufficient enough as the reward structure is dense and the multi agent task can reset often with short horizons.

We compute the clipped surrogate object in (3) which becomes the actor loss, this is stark contrast to Trust Region Policy Optimization where a hard KL penalty is added to avoid drifting from the original distribution of the policy. The critic is trained using discounted sum of all rewards which is the estimated target value. This is again the Monte Carlo estimate of the expected rewards. To summarize we use PPO as our main RL algorithm for its simplicity and effectiveness in Multi Agent RL tasks.

$$L^{\text{CLIP}}(\theta) = \mathbb{E}_t\left[\min\left(r_t(\theta)\hat{A}_t,\ \text{clip}(r_t(\theta),\ 1-\epsilon,\ 1+\epsilon)\hat{A}_t\right)\right] \tag{3}$$

The training infrastructure has been divided into five major parts along with some other utilities, 1. PPO Actor (contains the policy), 2. PPO Critic (contains the critic network), 3. PPO Agent (responsible for training, collecting data) and 4. Replay buffer (holds the data for the episode) 5. Environment abstraction with monkey patching for the environment observation space for communication. This structure is common for RL training and helps us abstract different components increases re-usability and debuggability. The training process resembles a standard online RL training paradigm. The training loop comprises of collecting data and adding the data to the replay buffer which is cleared and reset after every epoch. We collect 3 episodes for each training epoch. After collecting the data, we update both the actor and the critic twice by computing the respective target objectives. We choose to update the actor and critic only two full iterations per batch of data collected so that we do not significantly diverge from the policy used to collect the data.

All of our experiments are performed using decentralized training. In decentralized training, there are two variants 1. Shared network for all agents but trained separately per agent i.e. each batch input to the network is just the observation for the given agent. 2. Different networks for each agent. We use the first variant to improve our data efficiency.

We treat number of agents, communication and size of the critic network as bells and whistles and ablate across these dimensions. We report our results on two different metrics that we identify important for the given task to discover different patterns arising and support our hypothesis.

## 4 Experimental Setup

Simple Spread is a multi agent cooperative task in which there are N agents and N landmarks. The agents need to occupy the landmarks. The environment given a reward equivalent to the negative of the agent's distance to the respective landmark for each step, the agents get a reward of 1 if they reach the landmark.

The observation space comprises of **[agent vel x, agent vel y, agent pos x, agent pos y, landmark relative pos, N * other agents relative pos, (N - 1) * communication(optional)]**. The observation vector is a continuous space which is consumed by the actor and the critic network. The actions are a vector of size five **[left, right, down, up, nothing]** with floating point value denoting how much force needs to be applied in each direction. The actions also comprises of the communication vector if communication is enabled. For all our experiments for agent-env-agent communication each agent outputs the nearest landmark to it. The environment will collect this communication from different agents and append it to the observation vector of other agents enabling agent-env-agent communication. We don't use a neural network to compute this information as this is deterministic and can be computed using a deterministic policy. We collect three episodes worth of data with each episode being 40 trajectory steps long.

Our baseline is the policy with no communication among agents and our candidate is the policy that outputs communication which in our case is the location of of the nearest landmarks for the respective agents. We report our results on primarily two kinds of metrics, 1. Avg return across different Epochs which is computed using the total discounted reward per epoch / total episodes per epoch 2. Avg success rate per epoch which is computed as the total count of successes (agent comes 0.5 distance to the landmark) / total episodes per epoch.

## 5 Results

We report our results while changing the critic network size, number of agents and communication with consistent signs improvement communication and use the qualitative results as evidence to support our hypothesis.

### 5.1 Quantitative Evaluation

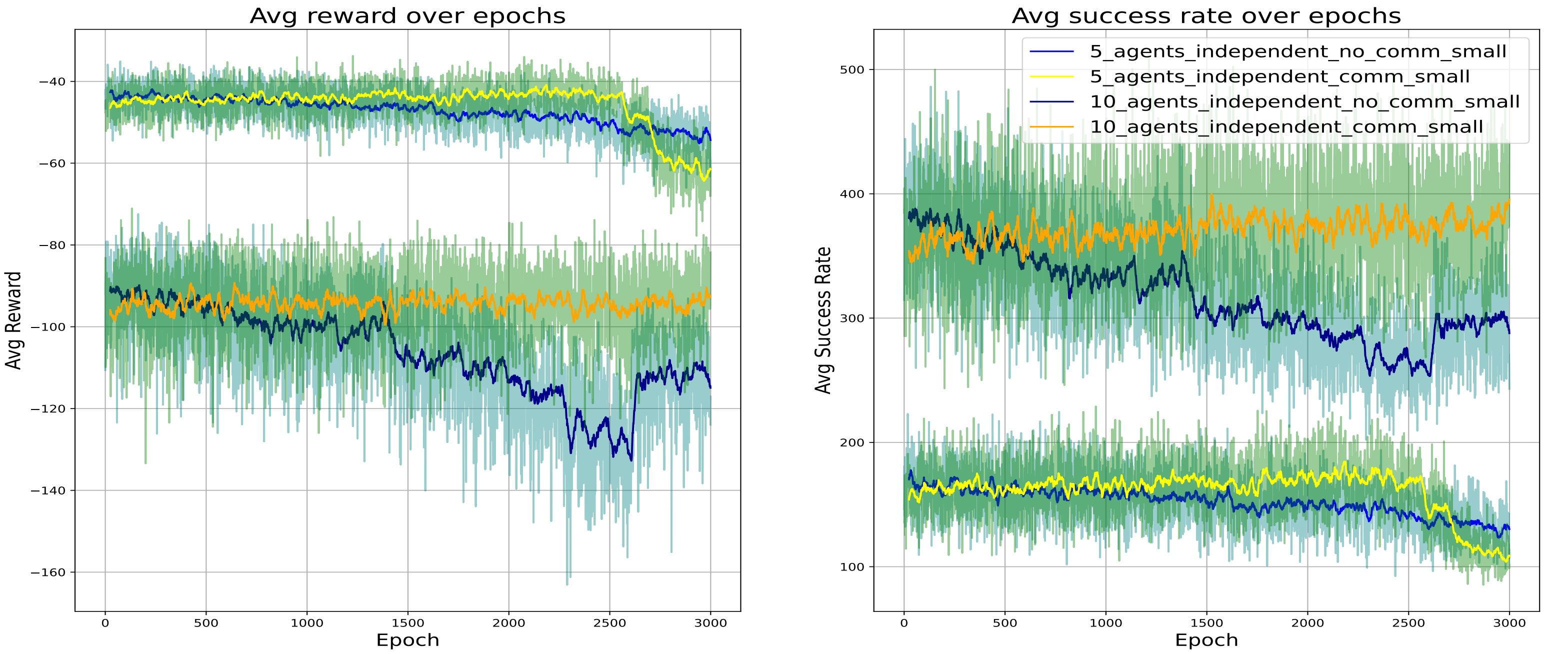

Figure 1: Avg Returns and Success rate for small Critic.

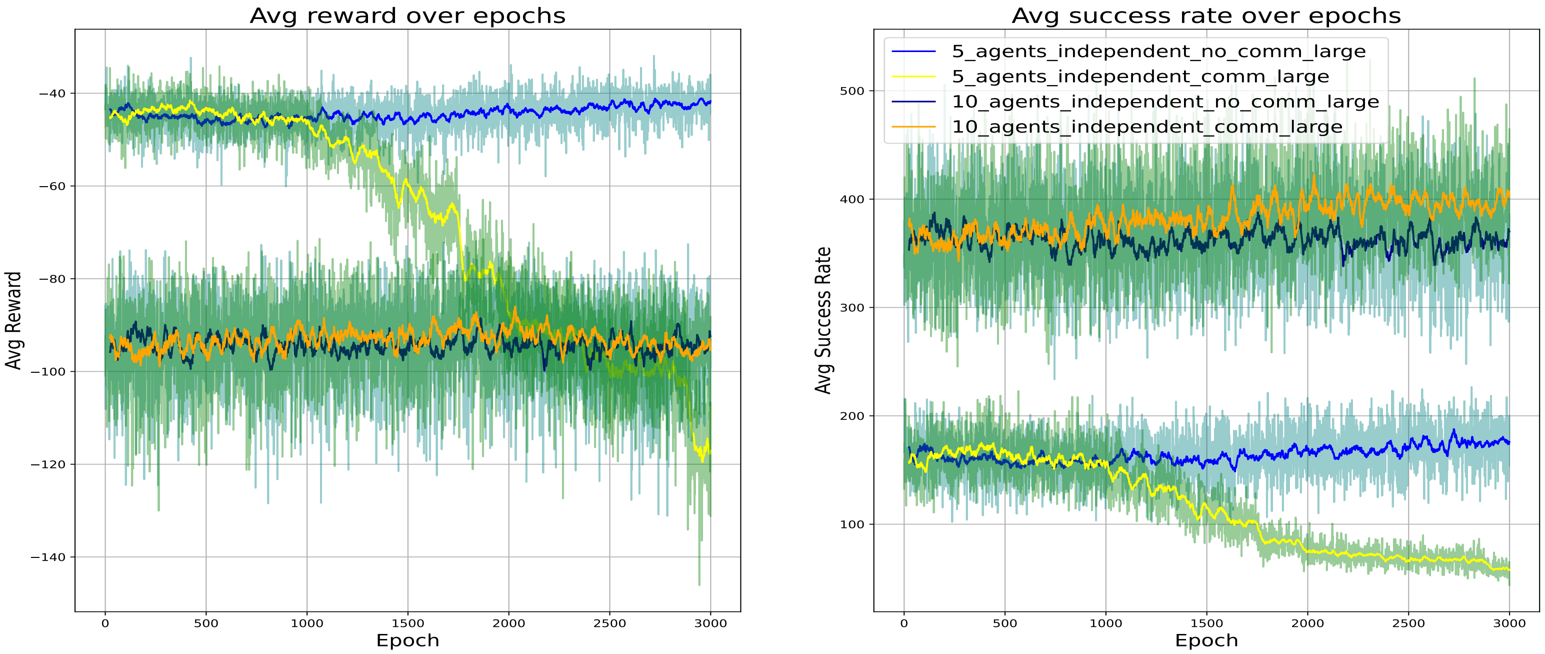

Figure 2: Avg Returns and Success rate for large critic.

We conduct eight experiments where we report avg success rate and avg returns for up-to 3000 epochs with different number of agents and critic network sizes. All the critic networks have three fully connect layers with ReLu activation function but vary in width.

We see a clear trend in the smaller critic network the communication variant is drastically significantly better for both 5 and 10 agents and outperforms the variant with no communication both in avg success rate and avg return. The trend also continues as we increase the size of the critic network. As training continues the performance of no-comm versions continue to degrade showing signs of underfitting or an issue with the input features where the features are not enough for the critic to learn a meaningful reward function. For the comm variant we see a consistent increase in performance for both metrics which eventually collapses for the 5 agents. We hypothesize that the collapse is due too over fitting of the neural nets.

For the experiment with 10 agents the the communication variant significantly outperforms the no-comm variant in avg success rate, with minor improvement in avg success rate. While we see an improved performance for 10 agents, the trend is reversed for the case where there are only 5 agents in the task. In this case, the avg reward and avg success rate show signs of modest performance gains in the earlier epochs but as training continues both the avg. returns and avg. success rate degrade drastically. We hypothesize that this is a sign of overfitting to the data which results in a collapse in performance as training continues.

The network can infer where the nearest landmark is corresponding to different agents, it has all the information in the observation. However this doesn't work in practice as the function that needs to be learned by the network is too complex and it won't be able to do so with the given data without overfitting to the data.

### 5.2 Qualitative Analysis

In this section we analyze the qualitative improvement for communication vs no communication. The blue blobs are agents and the black blobs are landmarks. For qualitative analysis we pick small critic network and 5 agents. On investigating Fig 3 which is the no-communication variant we notice that some agents do move towards the the landmarks but as the episode progresses we see that they drift away from the landmarks and clump up together. While, for in Fig 4 it is evident that for the communication variant some of the agents move towards the landmark and eventually occupy the landmark while one of them stays at the landmark before max number of cycles is reached. This observation is also supported by the evidence of increasing success rate for the given ablation case. While we see signs of improvement, we also notice that some of the agents are not able to find a good policy to move towards a landmark and remain stagnant. This behavior is a common challenge in Multi-Agent Reinforcement Learning and we leave further investigation for future work. We see similar patterns for 10 agents as well, particularly when the the avg success rate increases as training progresses.

We posit that the agent stagnation occurs due to the credit assignment problem. Since, the reward is global and obtained after actions of all the agents are collected and used to step into the environment with a dictionary obtaining global reward for the actions of all the agents. When using this collected data for training there is no understanding of the which actions lead to a higher reward and if some agents were stalling while others were moving towards the landmarks obtaining higher rewards, the network still learns to keep some agents stationary.

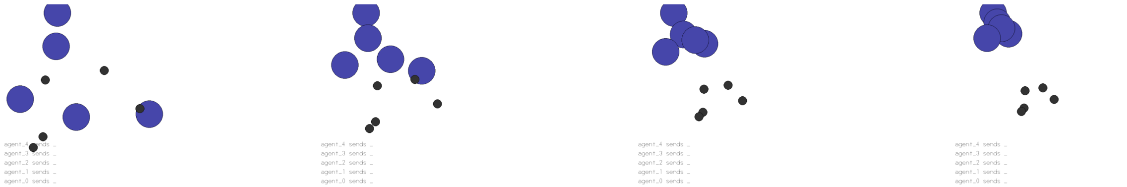

Figure 3: Training progression for 5 agents no - communication for 0, 10, 20 and 30 step idx.

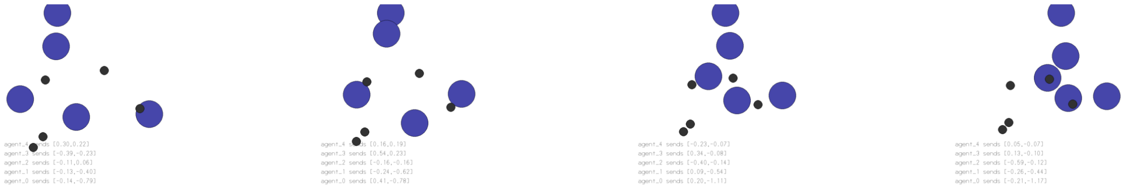
agent_4 sends [0.30,0.22]
agent_3 sends [-0.39,-0.23]
agent_2 sends [-0.11,0.06]
agent_1 sends [-0.13,-0.40]
agent_0 sends [-0.14,-0.79]
agent_4 sends [0.16,0.19]
agent_3 sends [0.54,0.23]
agent_2 sends [-0.16,-0.16]
agent_1 sends [-0.24,-0.62]
agent_0 sends [0.41,-0.78]
agent_4 sends [-0.23,-0.07]
agent_3 sends [0.34,-0.08]
agent_2 sends [-0.40,-0.14]
agent_1 sends [0.09,-0.54]
agent_0 sends [0.20,-1.11]
agent_4 sends [0.05,-0.07]
agent_3 sends [0.13,-0.10]
agent_2 sends [-0.59,-0.12]
agent_1 sends [-0.26,-0.44]
agent_0 sends [-0.21,-1.17]

Figure 4: Training progression for 5 agents communication for 0, 10, 20 and 30 step idx.

## 6 Discussion

From our results we observe only minor improvements in select cases particularly when the number of agents are ten and the critic network is large. We attribute the minor improvement due to the difficulty in tuning the model in different cases as the dimensionality of the input changes with communication vectors appended. This difficulty is exacerbated as we have to limit the width of the network to get an apples-apples comparison between the policy that uses communication and that does not. Remember, as number of agents increase the size of the inputs also increase. Additionally, when communication is enabled, the dimensionality of the input increases by (num agents - 1) * 2 but the width of the network remains the same. This could lead to an informational bottleneck.

The simple spread environment environment had a bug there by default the observation space has communication appended with a dimension of two and a default value of zero even though the agents are not sending any communication in their actions. We had to manually monkey-patch the environment to get the correct dimension of the observation space when there is no communication. We believe this technique could be applied to other environments with similar set up and would translate well to the real world with minimum tuning.

## 7 Conclusion

Revisiting our core research question—can communication improve multi-agent path planning when limited to non-agent-to-agent mechanisms—our quantitative analysis provides encouraging evidence of modest gains, as reflected across key evaluation metrics. While the improvements are not dramatic, they are consistent with qualitative observation and suggest potential.

Given the multi-dimensional nature of the ablation studies, exploring different architectural variants proved both challenging and time-intensive. One promising direction left partially unexplored is tuning the actor network, which may further clarify the role of communication in enhancing performance.

In summary, while our results indicate incremental progress, we acknowledge the scope for future extensions, including the integration of learned communication mechanisms and more targeted ablation studies—particularly those focusing on the actor's design and optimization.

## 8 Team Contributions

- **Group Member 1:** Responsible for researching and formulating the problem statement. Independently implemented PPO, supporting infrastructure and environment integration. Abhinav ran all experiments and wrote the report.

**Changes from Proposal** The project initially proposed to study emergent communication behavior, but the project pivoted to improving the performance.

## References

Raghav Aras, Alain Dutech, and François Charpillet. 2004. Stigmergy in Multi-Agent Reinforcement Learning. In *Proceedings of the Fourth International Conference on Hybrid Intelligent Systems (HIS'04)*. IEEE, 468–469. `https://doi.org/10.1109/ICHIS.2004.87`

Hoang-Dung Bui, Erion Plaku, and Gregory J. Stein. 2025. Multi-Agent Path Finding under Limited Communication Range Constraint via Dynamic Leading. *arXiv preprint arXiv:2501.02770* (2025). `https://arxiv.org/abs/2501.02770`

Tom Eccles, Yoram Bachrach, Guy Lever, Angeliki Lazaridou, and Thore Graepel. 2019. Biases for Emergent Communication in Multi-agent Reinforcement Learning. In *Advances in Neural Information Processing Systems*, Vol. 32. `https://proceedings.neurips.cc/paper_files/paper/2019/hash/fe5e7cb609bdbe6d62449d61849c38b0-Abstract.html`

Jakob N Foerster, Yannis M Assael, Nando de Freitas, and Shimon Whiteson. 2016. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in neural information processing systems*, Vol. 29. `https://papers.nips.cc/paper_files/paper/2016/file/2a65f2a5d4f312b52d7d4880b5dc1e74-Paper.pdf`

Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. 2017. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 30. 6379–6390.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder de Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. 2018. QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. In *Proceedings of the 35th International Conference on Machine Learning*. PMLR, 4295–4304.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. 2015. Trust Region Policy Optimization. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*. PMLR, 1889–1897.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347* (2017).

Mert Çetinkaya. 2021. Multi-Agent Path Planning Using Deep Reinforcement Learning. *arXiv preprint arXiv:2110.01460* (Oct 2021). `https://doi.org/10.48550/arXiv.2110.01460` Preprint.