Collaborative Auto-Curricula Multi-Agent Reinforcement Learning with Graph Neural Network Communication Layer for Open-ended Wildfire-Management Resource Distribution

Philipp D. Siedler
Independent Researcher
London, UK
{p.d.siedler}@gmail.com

Abstract

Most real-world domains can be formulated as multi-agent (MA) systems. Intentionality sharing agents can solve more complex tasks by collaborating, possibly in less time. True cooperative actions are beneficial for egoistic and collective reasons. However, teaching individual agents to sacrifice egoistic benefits for a better collective performance seems challenging. We build on a recently proposed Multi-Agent Reinforcement Learning (MARL) mechanism with a Graph Neural Network (GNN) communication layer. Rarely chosen communication actions were marginally beneficial. Here we propose a MARL system in which agents can help collaborators perform better while risking low individual performance. We conduct our study in the context of resource distribution for wildfire management. Communicating environmental features and partially observable fire occurrence help the agent collective to pre-emptively distribute resources. Furthermore, we introduce a procedural training environment accommodating auto-curricula and open-endedness towards better generalizability. Our MA communication proposal outperforms a Greedy Heuristic Baseline and a Single-Agent (SA) setup. We further demonstrate how auto-curricula and openendedness improves generalizability of our MA proposal.

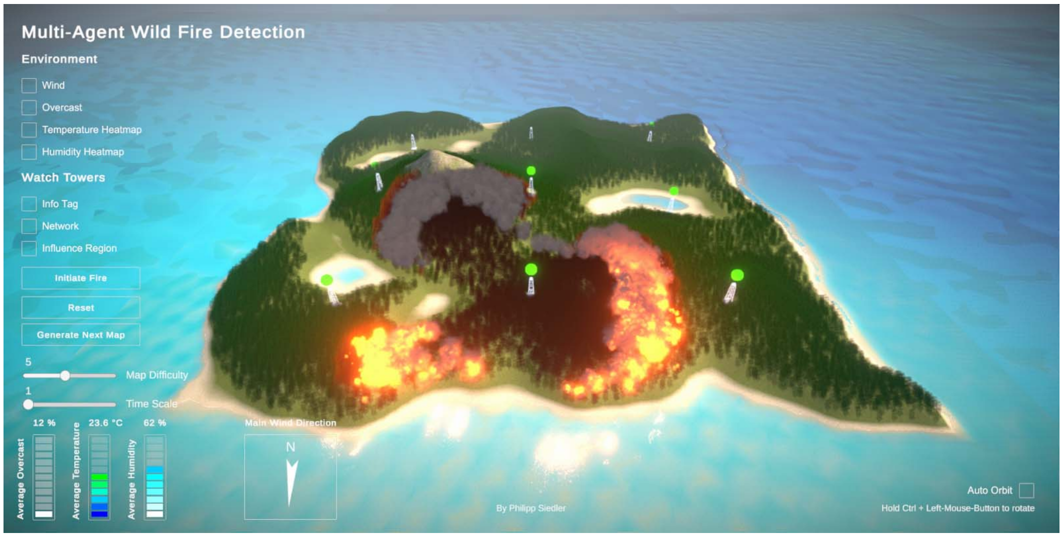


Figure 1: Dashboard of multi-agent wildfire environment in inference mode. A web browser application can be found at: `https://philippds-pages.github.io/RL-Wild-Fire_WebApp/`.

1 Introduction

1.1 Motivation

The human ability to communicate motives and intentions is the basis for our society's success. If the intersection of such is large enough, communication can lead to collaboration between involved parties. A collaboration exists if all collaborators behave beneficially for themselves and the collective of parties. A collective party can represent a group or an individual with an agency; therefore, we can consider entities with specific interests agents. Many, if not all, domains in society involve multiple agents and can be described as MA systems. While human intelligence is the highest observed in nature, not only human societies strive through collaboration. Insects and Mammals collaborate to procreate, care, protect and ultimately survive as species. Bees, i.e. assign each other tasks, request help and fight off predators (Bonabeau et al., 1999). The highly cooperative Meerkats organise themselves in groups of up to fifty members, while some breed, others help raise the young offspring by foraging and keeping watch (Clutton-Brock, 2002). Some monkeys and lions even kill the young of revivals, i.e. to maintain food source saturation (Hoogland, 1985).

In the context of nature, Charles Darwin argues for the survival of the fittest (Darwin, 1977) and, therefore, the occurrence of competition. While in Artificial Intelligence, the majority of significant work on MA systems consider two opposing agents only, the problems of interest of this work are cooperative MA systems, where groups of agents act together to achieve higher individual and collective goals (Cohen et al., 1997; Guestrin et al., 2002; Decker, 1987; Panait & Luke, 2005; MATARIC, 1998). Just like in human society or the animal world, individuals have unique or mixtures of motives. However, we can define agents with mixed or identical motives in a MA environment simulation. Assuming shared intentionality leaves us with the question of how to collaborate. Communication can play a crucial role to collaborate successfully. Human society uses language as communication medium (Barón Birchenall, 2016). Agents can send signals of various types as a form of language. Nevertheless, observing others' behaviour can be a form of communication. Body language, a tail-wagging dog, or the red colour of an octopus can communicate internal states and intention. But we can also design agents that directly share policies - state action transitions - or memory data of past experiences. Core questions we ask: Can agents in a MA system learn the importance of communication? Subsequently: Can agents learn to use the communicated information to take actions beneficial for themselves and the collective? And finally: Can agents learn to ask for help, form temporary alliances to encounter a high-stress state? Answering these questions will be the challenge of the experiments conducted and presented in this paper.

1.2 Contribution

In this work, we study communication in the context of a distributed wildfire lookout tower grid. Wildfires occur continuously and globally as part of the Earth's ecosystem (Bond & Keeley, 2005). Furthermore, climate change increases the likelihood of extreme wildfire conditions worldwide (Goss et al., 2020; Coogan et al., 2019). Approximately 420 Mha is the total estimated area burned annually. While humans initiate 90% of wildfires and only 10% by lightning, environmental conditions, topography and fuel composition can suppress or enhance occurrence and growth. Satellites can detect fires, but only once they are already too large, and some areas are not well covered by cell phone networks. We propose unmanned lookout towers equipped with environmental sensors, cameras and local processing units. LoRa (long-range) is a low-power wide-area network modulation technique that can send signals with low power requirements in ranges of up to 15 kilometres (Corporation, 2020). As outlined in the introduction, we chose the wildfire problem context based on its high impact and to minimise the gap between the testbed environment and the real world. By working closely on a real-world problem and its complexity, we believe we can achieve better MA collaboration systems, which is the main focus of this paper (Küttler et al., 2020). We use a graph to organize our proposed MA lookout tower grid. Each lookout tower agent is represented by a node and has three closest neighbours to exchange local information. Such information exchange can help predict fire growth and fire management resource distribution across the lookout tower grid. Proposed communication mechanisms consist of two strands: 1. streaming of information between neighbours and 2. requesting help and answering a help request as part of the agent action space. The environment includes multiple environmental conditions, topology, and distributed fuel in the form of forest volume. To minimise simulation inaccuracies and increase generalizability, we can procedurally generate an infinite amount of environment conditions with varying difficulty levels (Jaderberg et al., 2021). Furthermore, the environment design allows for auto-curricula, in which the MA collective can advance automatically from one difficulty level to another to improve training (Baker et al., 2020).

We present a message passing (Gilmer et al., 2017) GNN (Scarselli et al., 2009) based communication layer on top of a MARL mechanism (Zhang et al., 2021). We use a Proximal Policy Optimization (PPO) algorithm (Schulman et al., 2017) for our Reinforcement Learning (RL) agents. We demonstrate how our collaboration mechanism using communication can help agents organise themselves to surpass a Greedy Heuristic, SA baselines. Furthermore, how our environment design, accomodating for openendedness and auto-curricula, can help our MA system advance further to become more robust and generally perform well, even in unseen environments.

2 Related Work

While wildfire science is not the main domain of this paper, we still think it is important to set the stage and point to some relevant work in the field we have drawn inspiration and insight from. Generally, we found that RL approaches in wildfire applications are vastly underrepresented (Jain et al., 2020). Only a few works use RL algorithms, such as advanced actor-critic (A3C) and Monte Carlo tree search (MCTS) methods, addressing topics related to fire behaviour prediction, more specifically fire spread and growth (Subramanian & Crowley, 2017; Ganapathi Subramanian & Crowley, 2018). Nevertheless, we have been pinpointing aspects from various applications in wildfire science that help us develop a better wildfire simulation environment. One interesting aspect resonating with our communication approach is remote data sensing in possibly hard to reach terrain, spread across a network of agents organised in proximity neighbourhoods and directed graphs (Huot et al., 2021). While we are looking at a highly distributed stationary MA system, there has been work on networks (Haksar & Schwager, 2018) of autonomous unmanned aerial vehicles (UAV) to monitor and predict wildfire growth (Julian & Kochenderfer, 2019; Afghah et al., 2019). Finally, we want to mention work on predicting wildfire using climate data, which is part of our agents sensing abilities (Xiong et al., 2020).

RL is a powerful learning paradigm consisting of an agent interacting with an environment to learn from experiences through positive or negative rewards. From the perspective of an individual agent in a MA system, all other agents are part of the environment. Therefore SARL (Single-Agent Reinforcement Learning) is building the foundation for MARL (Wang & Raj, 2017). RL does not require any data; consequently the learning environment plays a crucial role in providing enough and diverse experiences (Jaderberg et al., 2021) in conjunction with carefully crafted reward signals. Many domains are interested in RL research and applications (Leitão & Karnouskos, 2015), such as game theory and distributed systems, but also optimal control, autonomous cars (Shalev-Shwartz et al., 2016) and robotics (Kober et al., 2013; Gupta et al., 2017; Ismail & Sariff, 2018). Games have been part of one of three main historical threads of RL development (Sutton & Barto, 2015). Therefore naturally, MARL has been studied using a diversity of games, including multiple competing and cooperating players. Traditional two-player tabletop games such as GO (Silver et al., 2016; 2017), Chess (Campbell et al., 2002), Shogi (Silver et al., 2018) and Hex (Anthony et al., 2017), recent work on multi-player games such as Poker (Moravčík et al., 2017; Brown & Sandholm, 2018) and Diplomacy (Anthony et al., 2022; Calhamer, 1959), but also computer games including Atari games (Mnih et al., 2015), Dota (Berner et al., 2019), Starcraft (Vinyals et al., 2019) and overcooked (Fontaine et al., 2021) have significantly shaped developments in AGI and RL research. Game engines such as unity include realistic physics (Ward et al., 2020), ideal for digital twins of real-world scenarios.

While there is work on various methods on collaboration without active communication (Matignon et al., 2012; Panait & Luke, 2005) such as gradient-based distributed policy search (Peshkin et al., 2000), reward function sharing (Lauer & Riedmiller, 2000), memory sharing (Lowe et al., 2017; Pesce & Montana, 2020; Hernandez-Leal et al., 2019) and parameter sharing (PS) (Gupta et al., 2017; Hernandez-Leal et al., 2019), we are interested in communication as part of the agents' action space (Xuan et al., 2001). Active communication requires a protocol and a medium. A protocol describes the rule of communication. The medium could be anything from low-level binary data (Berna-Koes et al., 2004), discrete or continuous, text, numbers-based or a combination of such as message packages. In our work, vectors of observations are part of the messages sent (MATARIC, 1998), but other work proposes transferring more complex information, such as intentions or policy gradients (Foerster et al., 2016). Our work can be classified as a decentralised, partially observable Markov decision process (Dec-POMDP) (Oliehoek, 2012) in combination with a GNN (Scarselli et al., 2009) message passing (Gilmer et al., 2017) communication layer. The proposed MA communication mechanism builds on a combination of previous work, including communication as part of the agents' action space (Foerster et al., 2016), enabling the agent to send help requests to members of its neighbourhood and information broadcasting as an extension of the lookout towers local sensing capabilities (Sukhbaatar et al., 2016). Our message passing GNN is structuring incoming neighbourhood information, while work by (Almasan et al., 2020) implemented a GNN, as part of the main training feedback. Recently published work on MARL and GNN communication layer is demonstrating how communication can improve the collective performance, and the importance of shared information (Siedler, 2021). Here we advance the communication protocol and the agents ability to raise and answer help requests actively. Requesting help, as well as helping is part of the agents action space.

3 Background

3.1 Proximal Policy Optimisation

All agents are trained using state of the art algorithm Proximity Policy Optimization (PPO). Two main concepts distinguish PPO. Firstly, PPO estimates a trust region to take safe learning steps while performing gradient ascent. Secondly, Advantage estimates how good an action is compared to the average action in a specific state. Many other RL algorithms, such as Asynchronous Advantage Actor Critic (A3C), use this concept (Udacity-DeepRL, 2019). **Advantage:** Advantage can be described as the difference of the Q Function and the Value Function: $A(s, a) = Q(s, a) - V(s)$, where s is the state and a the action (Zychlinski, 2019). The Q Value (Q Function), denoted as $Q(s, a)$, measures the overall expected reward given state s, performing action a. Assuming the agent continues playing until the end of the episode following policy π. The Q is abbreviated from the word Quality, and denoted as: $\mathcal{Q}(s, a) = \mathbb{E}\left[\sum_{n=0}^{N} \gamma^n r_n\right]$. The State Value Function, denoted as $V(s)$, measures, similar to the Q Function, overall expected reward, with the difference that the State Value is calculated after the action has been taken and is denoted as: $\mathcal{V}(s) = \mathbb{E}\left[\sum_{n=0}^{N} \gamma^n r_n\right]$. The Q Value $V(s)$, with $n = 0$, is the expected reward r^0 in state s, before action a was taken, while the Q Value measures the expected reward r^0 after a was taken. **Trust Region:** After some experience samples $\pi_{\theta_k}(a_t|s_t)$ have been collected, the trust region can be calculated as the quotient of the current policy to be refined $\pi_\theta(a_t|s_t)$ and the previous policy as follows $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)} = \frac{current\ policy}{old\ policy}$. This is a simplified gradient ascent objective function with limited deviation between the current and old policies (Achiam, 2018).

$$\mathcal{L}_{\theta_k}^{CLIP}(\theta) = \underset{s,a\sim\theta_k}{\mathbb{E}}\left[\min\left(r_t(\theta)A^{\theta_k}(s, a), g(\epsilon, A^{\theta_k}(s, a))\right)\right],$$

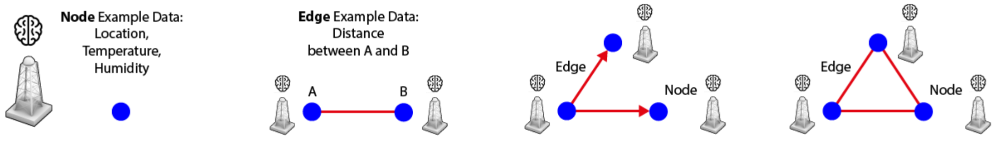
where

$$g(\epsilon, A) = \begin{cases} (1+\epsilon)A, & \text{if } A \geq 0 \\ (1-\epsilon)A, & \text{otherwise} \end{cases}$$

The advantage function will be clipped to the value at $(1-\epsilon)$ or $(1+\epsilon)$, if the probability ratio between the current and the previous policy is outside the range of $(1+\epsilon)$ and $(1-\epsilon)$. This also means that the advantage will never exceed the clipped values. In the original PPO paper by (Schulman et al., 2017) ϵ was set to 0.2. Finally, the policy that yields the highest sum over all Advantage estimates A_t in range of max time step T of a trajectory $\tau \in \mathbb{D}_k$ will be used to override the old policy θ_{old} (OpenAI, 2021): $\theta_{k+1} = \underset{\theta_k}{argmax} \frac{1}{|\mathbb{D}_k|T} \sum_{\tau\in\mathbb{D}_k} \sum_{t=0}^{T} \min\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)}, g(\epsilon, A^{\theta_k}(s, a))\right)$.

3.2 Graph Neural Network

Many flavours of GNNs exist (Li et al., 2017; Veličković et al., 2018; Defferrard et al., 2017), but (Scarselli et al., 2009) fundamentally introduced them. A graph is a data structure based on nodes or vertices and edges. Nodes are objects holding arbitrary features. Edges represent the relationships between nodes. Edges can be directed from node A to B 2c, or undirected, from node A to B and vice versa 2d.



(a) Graph Node (b) Graph Edge (c) Directed Graph (d) Undirected Graph

Figure 2: Graph $\mathcal{G}$ consisting of vertices $\mathcal{V}$ (blue dots) and edges $\mathcal{E}$ (red lines): $\mathcal{G} = (\mathcal{V}, \mathcal{E})$

The basic functionalities of GNNs are graph, node and edge classification. Features of a node can be predicted using edges or the existence of edge connections using node features. Graphs as a whole can be classified i.e. using node features and the graphs topology. However, the simplest form of a GNN is the message passing framework proposed by (Gilmer et al., 2017) using the network architecture introduced by (Battaglia et al., 2018), utilising a "graph-in, graph-out" architecture. The input graph topology is not modified but its loaded feature embeddings.

Node states can be denoted as v, edges connecting with node v as $x_{co[v]}$. The state of a node h_v consists of n-dimensional vector features. Adjacencies between a node and its neighbours are the mapped transition of the node, denoted as $h_{ne[v]}$, including all neighbouring node features, denoted as $x_{ne[v]}$. The transition function f is used to embed each node on a n-dimensional space (Zhou et al., 2020): $h_v = f(x_v, x_{co[v]}, h_{ne[v], x_{ne[v]}})$
While the two most popular algorithms to define neighbourhoods on graphs are Breadth-First Search (BFS) (Burkhardt, 2021), Depth-First Search (DFS) (Kaur & Garg, 2012) and random walk based DeepWalk (Perozzi et al., 2014), we define neighbourhoods by finding Euclidean distance based n nearest neighbours. Passing state h_v and feature x_v to the GNN outputs the result of function g: $o_v = g(h_v, x_v)$. A basic last step is applying gradient descent to formulate loss using the ground truth t_v as well as the output o_v of node v: $loss = \sum_{i=1}^{p}(t_i - o_i)$. In our approach we are using gradient ascent and a reward function utilised by PPO.

3.3 Multi-Agent Communication

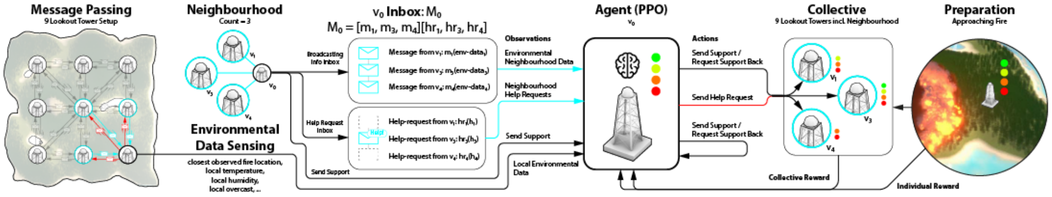
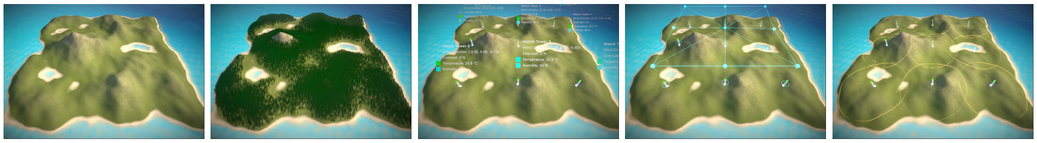


Figure 3: GNN Message Passing Communication Diagram: Neighbourhood graph (n=3); Observations: Inbox environmental data, inbox help requests and local environmental data; Agent (PPO); Actions: send support, request support back, send a help request to others, support self; Rewards: individual and collective reward for preparedness in case of fire. A zoomed-in version of this diagram can be found in the appendix: Figure 15.

Agent communication is possible between an agent and the agents in its neighbourhood. Two main functionalities define communication. Firstly, each agent can send help request hr messages as part of its action space. Help request messages are sent at time step t and received by all neighbours u_v of v at time step $t + 1$. The neighbouring agents can now collaborate and react to the help request hr_t by sending resources, if available, at time step $t + 2$ to support v. If the sent resources help v, u_v gets a small positive bonus reward of $+0.1$ for helping, no other agent can receive a bonus reward for reacting to hr_t thereafter. A help request message consists of a boolean signal, $hr_t = [false/true]$. An agent can receive multiple help requests as part of its help request inbox $hri[hr_1, hr_2, hr_3]$ and needs to decide which to react to if at all. And secondly, information broadcasting between each agent v and its neighbours u_v. All neighbours receive broadcasted messages in an inbox $ibi[ib_1, ib_2, ib_3,]$. Received messages $ib[cof_{pos}(x, y, z), temp, hum, prep, oc]$ including information such as closest observed fire location (if existing) cof_{pos}, in the form of a 3-dimensional vector $cof_{pos}(x, y, z)$, local temperature $temp$, humidity hum, the current preparation value $prep$ and the percentage of overcast oc at the lookout tower location as scalars. The graph-structured communication is the input to the neural network of the GNN. The agent has to learn how to reason about the broadcasting information and whether to support its neighbours or itself in preparation for approaching fire. While helping a neighbour might yield a bonus reward, there is a risk for unpreparedness of its own lookout tower, which might result in low rewards.

4 Methodology

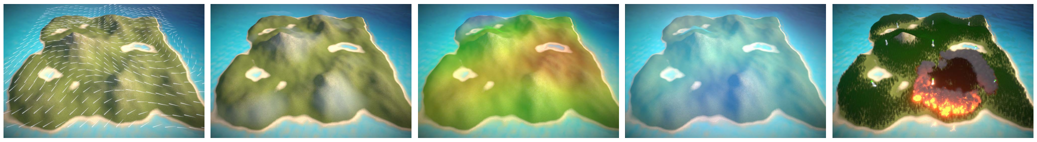
4.1 Environment



(a) Terrain (b) Forest (c) Lookout Towers (d) Network (e) Lookout Region

Figure 4: Static environment features. Zoomed-in version in the appendix: I.

We now describe the 3-D Wildfire Lookout Tower environment, developed in the game engine unity, used for training and evaluating the Greedy Heuristic Baseline, Single- and Multi-Agent experiments. The scenario is a procedurally generated landmass with a distributed network of lookout towers, forest and environmental condition features. Static features of the environment are shown in Figure 4. A sample of the terrain is shown in Figure 4a. Terrain height values influence the distribution of trees forming the forest volume (4b). Nine lookout towers are distributed in a fixed three by three grid. The placement of a lookout tower also determines the neighbourhood, consisting of the three closest lookout towers. Each lookout tower has a fixed observation region (Figure 4e, 16).

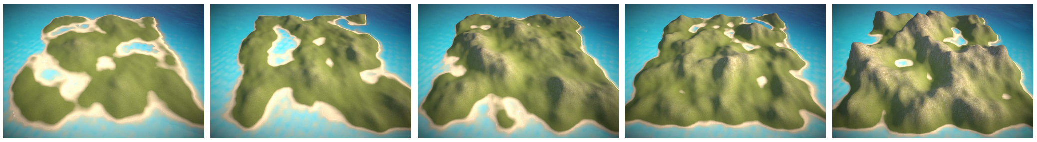


(a) Wind (b) Overcast (c) Temperature (d) Humidity (e) Fire

Figure 5: Dynamic environment features. Zoomed-in version in the appendix: I.

The dynamic features of the environment (Figure 5) are based on perlin noise (Perlin, 1985) and a main wind direction, including a wind field (5a), overcast (5b), temperature (5c) and humidity (5d). Wildfire is also part of the dynamic environment features. The fire's initiation and growth are based on environmental features and probability. At the beginning of an episode the location on the terrain with the lowest overcast, highest temperature and lowest humidity is chosen to ignite a wild fire. Fire can not spread to the next tree when the distance is larger than ten meters. However, if the distance is lower, the following conditions add twenty percent each to the probability of fire spreading: if the angle between the wind direction vector and the target vector is lower than 45 degrees; if the target is at a higher location; if the target temperature is higher than 21 Celsius degree; if the target humidity is higher than 50 percent and if overcast is zero. When all conditions are true, the chance of fire spreading to the target is 100 percent. Once the fire has spread, a tree will burn ten time-steps. At this point it is important to stress that the focus of this paper is not on simulating hyper-realistic environmental conditions or fire behaviour, rather a close adjustable abstraction.

While we are aware of the fact that simulation will always differ from the real world, we try to close the simulation to reality gap by following a strategy explained by Thore Graepel, creating as much diversity in our simulation environment as possible (Fry, 2022). We aim towards achieving open-endedness through procedurally generating a virtually infinite amount of terrain scenarios. Additionally, ten difficulty levels modify the height of the terrain, resulting in less observable regions



(a) seed=0, diffic.=1 (b) seed=1, diffic.=3 (c) seed=2, diffic.=5 (d) seed=3, diffic.=7 (e) seed=4, diffic.=10

Figure 6: Open-ended environment and increasing difficulty. Zoomed-in version in the appendix: F.

for each agent and, therefore, higher communication necessity and overall higher difficulty predicting fire growth. Furhter terrain samples can be found in the difficulty vs seed matrix in the Appendix (F).

4.2 Performance Evaluation Method:

Each lookout tower has a resource reserve rr of value 1.0. Initially, all lookout towers support value sv is at 0.0. Resources from the resource reserve rr can be distributed in 0.1 increments. Distribution targets can be self or lookout towers in the neighbourhood. In order to distribute resources to a target, the resource reserve rr needs to be larger or equal to 0.1. If the resource reserve is empty, resources have to be deducted in 0.1 decrements from self or lookout towers in the neighbourhood to free up resources for redistribution. If there is no self-need for resources, agents can collect reward fractions for distributing resources to neighbouring lookout towers in need. An example scenario could be: 0.5 resources are distributed to self, and 0.5 resources at a neighbouring lookout tower. There is no observed fire near self, but fire is approaching at the neighbouring location. The agent now gets 0.5 times the performance of the neighbouring tower, but none for the distributed resources at target self. Performance is calculated using a broken power law function using $\beta = -1$, $s = 2$, $x_n = 270$, $a = 5$ and $x = \frac{\text{distance to closest observed fire}}{\text{influence region distance}}$ leading to $F(x, x_n, a, s, \beta) = \left(1 + [\frac{x*1000}{x_n}]^a\right)^{\frac{-1}{s}}$, where

$$x = \begin{cases} x \text{ remapped from domain 0 to 1, to domain 0.5 to 0,} & \text{if fire moves towards tower} \\ x \text{ remapped from domain 0 to 1, to domain 0.5 to 1,} & \text{otherwise.} \end{cases}$$

The reward function will yield the highest possible reward if a tower is prepared well and has a high support value before fire crossing the influence region. Each tower relies on environmental data and observed fire locations to predict how much support preparation is needed.

4.3 Experiments

Table 1: Experiment setups and parameters for lookout tower grids of 9. Auto-curriculum (AC), seed (s), environment terrain with seed 0 (s=0); infinite environment terrain scenarios (s=inf).

				Observation(s)			
Setup	Agent Count	Tower Count	Neighbour Count	At each Tower	Total	Stack Size	Action(s) per Agent
Greedy Heuristic	0	9	0	0	0	0	0
Single-Agent (s=0)	1	9	3	7	63	2	36
Multi-Agent (s=0)	9	9	3	32	32	2	5
Multi-Agent (s=inf,AC)	9	9	3	32	32	2	5

All agent experiments have been trained and tested for 500 time-steps per episode. Each agent can take multiple decisions at every time step, depending on its action space. **Greedy Heuristic Baseline:** The hand-designed greedy heuristic baseline keeps all resources to itself and does not support neighbours. Naturally, no training is required. **Single-Agent (seed 0):** The agent in the single-agent setup controls resource distribution of all lookout towers. The egoistic reward is gained by distributed resources, multiplied by performance value p, and the average collective reward cr consisting of the average performance ap over all lookout towers. The environment scenario with seed 0 is used for training. **Multi-Agent (seed 0):** In the multi-agent setup each lookout tower is controlled by an individual agent. All agents receive egoistic and collective rewards, as explained in the single-agent setup description. Additionally in the multi-agent setup, reacting to a help request first yields a small bonus reward. The environment scenario with seed 0 is used for training. **Multi-Agent (Openended & Autocurricula):** Finally, to show the strength of openendedness (seed=inf) and auto-curricula (AC), we trained a multi-agent setup, but with changing environment scenario for each episode as well as rising difficulty level. The difficulty level advances if the agent has achieved a certain cumulative reward threshold, over 100 past episodes. Further details on the curricula design can be found in the Hyperparameters Appendix B.3.

4.4 Results

Table 2: Experiment results while training and inference mode, including training time. Inference: Mean reward and performance for environment with seed 0 and seed inf.

	Training Time	Inference: Mean Reward		Inference: Performance	
Setup	5e7 step(s) (↓ better)	seed=0 (↑ better)	seed=inf (↑ better)	seed=0 (↑ better)	seed=inf (↑ better)
Greedy Heuristic	-	122.3±46.7	111.2±36.1	0.117±0.039	0.109±0.034
Single-Agent (s=0)	787e3(sec)	**1073.1±457.3**	589.9±283.3	**0.189±0.070**	0.111±0.043
Multi-Agent (s=0)	132e3(sec)	995.5±172.2	876.1±131.2	0.178±0.035	0.154±0.226
Mutli-Agent (s=inf,AC)	**128e3**(sec)	967.5±164.17	**907.0±142.7**	0.171±0.034	**0.158±0.027**

Results show that our Multi-Agent proposal surpasses the Greedy Heuristic, Single-Agent and Multi-Agent setup in unseen environments (seed=inf). While the Single-Agent setup can also achieve high rewards, the training-time is almost 10 times higher. We further show that setups that have been trained on a single environment only return low cumulative rewards on unseen environments (seed=inf). While the Multi-Agent setup, trained on multiple environments (seed=inf) with an auto-curricula (AC), yields lower mean rewards on the seed 0 environment, it outperforms the other setups - not trained without further environments and auto-curricula - by a wide margin. Additional data on training (D) and inference (E), including how communication helps our approach to achieve higher performance and cumulative rewards, can be found in the Appendix.

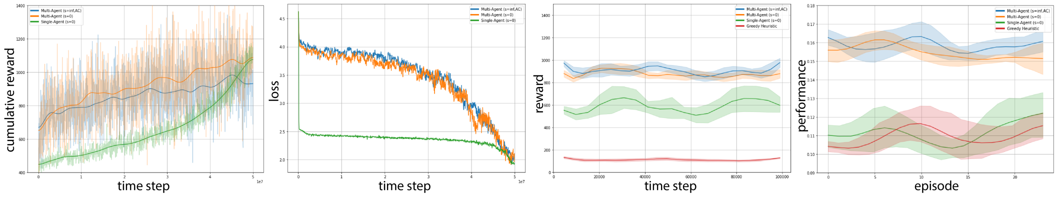


Figure 7: First and second diagram: Training over 5e7 total time steps: cumulative rewards and loss. Third and fourth diagram: Inference on various environments with difficulty level 8: reward vs time step and performance vs episode.

5 Discussion and Future Work

There are three interesting directions to develop our work further: Firstly, we define neighbourhoods as the three nearest lookout towers, using Euclidean distance. There are further strategies to define neighbourhoods, such as Breath-First-Search (BFS) (Burkhardt, 2021). BFS allows to define multi-layered neighbourhoods. Furthermore incoming messages could be pooled and weighted depending on the distance or neighbourhood-layer of the lookout towers the message is coming from. Secondly, instead of rewarding helping with a bonus reward, we could turn this around and slightly weight rewards higher for actions that benefit agents self first. And lastly we could threshold the support amount one lookout tower is able to receive. In our current approach the centre most lookout tower could hold full support of all eight neighbours, which might not be beneficial for the collective but yield temporarily high egoistic rewards.

ACKNOWLEDGMENTS

We want to thank Serkan Cabi, Panagiotis Tigas for their guidance and Ian Gemp for igniting the wild fire context idea. We also want to thank the reviewing committee for their efforts and critic. Further information, video material and an interactive web-app can be found at: `https://ai.philippsiedler.com/iclr2022-gnn-marl-wild-fire-management-resource-distribution/`.

REFERENCES

Joshua Achiam. Simplified PPO-Clip Objective, July 2018. URL `https://drive.google.com/file/d/1PDzn9RPvaXjJFZkGeapMHbHGiWWW20Ey/view?usp=sharing&usp=embed_facebook`.

Fatemeh Afghah, Abolfazl Razi, Jacob Chakareski, and Jonathan Ashdown. Wildfire Monitoring in Remote Areas using Autonomous Unmanned Aerial Vehicles. *arXiv:1905.00492 [cs, eess]*, April 2019. URL `http://arxiv.org/abs/1905.00492`. arXiv: 1905.00492.

Paul Almasan, José Suárez-Varela, Arnau Badia-Sampera, Krzysztof Rusek, Pere Barlet-Ros, and Albert Cabellos-Aparicio. Deep Reinforcement Learning meets Graph Neural Networks: exploring a routing optimization use case. *arXiv:1910.07421 [cs]*, February 2020. URL `http://arxiv.org/abs/1910.07421`. arXiv: 1910.07421.

Thomas Anthony, Zheng Tian, and David Barber. Thinking Fast and Slow with Deep Learning and Tree Search. *arXiv:1705.08439 [cs]*, December 2017. URL `http://arxiv.org/abs/1705.08439`. arXiv: 1705.08439.

Thomas Anthony, Tom Eccles, Andrea Tacchetti, János Kramár, Ian Gemp, Thomas C. Hudson, Nicolas Porcel, Marc Lanctot, Julien Pérolat, Richard Everett, Roman Werpachowski, Satinder Singh, Thore Graepel, and Yoram Bachrach. Learning to Play No-Press Diplomacy with Best Response Policy Iteration. *arXiv:2006.04635 [cs, stat]*, January 2022. URL `http://arxiv.org/abs/2006.04635`. arXiv: 2006.04635.

Bowen Baker, Ingmar Kanitscheider, Todor Markov, Yi Wu, Glenn Powell, Bob McGrew, and Igor Mordatch. Emergent Tool Use From Multi-Agent Autocurricula. *arXiv:1909.07528 [cs, stat]*, February 2020. URL `http://arxiv.org/abs/1909.07528`. arXiv: 1909.07528.

Leonardo Barón Birchenall. Animal Communication and Human Language: An overview. *International Journal of Comparative Psychology*, 29(1), 2016. ISSN 0889-3675. URL `https://escholarship.org/uc/item/3b7977qr`.

Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Caglar Gulcehre, Francis Song, Andrew Ballard, Justin Gilmer, George Dahl, Ashish Vaswani, Kelsey Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matt Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks. *arXiv:1806.01261 [cs, stat]*, October 2018. URL `http://arxiv.org/abs/1806.01261`. arXiv: 1806.01261.

M. Berna-Koes, I. Nourbakhsh, and K. Sycara. Communication efficiency in multi-agent systems. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, volume 3, pp. 2129–2134 Vol.3, April 2004. doi: 10.1109/ROBOT.2004.1307377. ISSN: 1050-4729.

Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, Rafal Józefowicz, Scott Gray, Catherine Olsson, Jakub Pachocki, Michael Petrov, Tim Salimans, Jeremy Schlatter, Jonas Schneider, Szymon Sidor, Ilya Sutskever, Jie Tang, Filip Wolski, and Susan Zhang. Dota 2 with Large Scale Deep Reinforcement Learning. pp. 66, December 2019.

Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm Intelligence: From Natural to Artificial Systems*. Santa Fe Institute Studies on the Sciences of Complexity. Oxford University Press, New York, 1999. ISBN 978-0-19-513158-1. doi: 10.1093/oso/9780195131581.001.0001. URL https://oxford.universitypressscholarship.com/10.1093/oso/9780195131581.001.0001/isbn-9780195131581.

William J. Bond and Jon E. Keeley. Fire as a global 'herbivore': the ecology and evolution of flammable ecosystems. *Trends in Ecology & Evolution*, 20(7):387–394, July 2005. ISSN 0169-5347. doi: 10.1016/j.tree.2005.04.025. URL https://www.cell.com/trends/ecology-evolution/abstract/S0169-5347(05)00132-1. Publisher: Elsevier.

Noam Brown and Tuomas Sandholm. Superhuman AI for heads-up no-limit poker: Libratus beats top professionals. *Science*, 359(6374):418–424, January 2018. doi: 10.1126/science.aao1733. URL https://www.science.org/doi/10.1126/science.aao1733. Publisher: American Association for the Advancement of Science.

Paul Burkhardt. Optimal algebraic Breadth-First Search for sparse graphs. *ACM Transactions on Knowledge Discovery from Data*, 15(5):1–19, June 2021. ISSN 1556-4681, 1556-472X. doi: 10.1145/3446216. URL http://arxiv.org/abs/1906.03113. arXiv: 1906.03113.

AB Calhamer. Diplomacy (game), 1959. URL https://dbpedia.org/page/Diplomacy_(game).

Murray Campbell, A. Joseph Hoane, and Feng-hsiung Hsu. Deep Blue. *Artificial Intelligence*, 134 (1):57–83, January 2002. ISSN 0004-3702. doi: 10.1016/S0004-3702(01)00129-1. URL https://www.sciencedirect.com/science/article/pii/S0004370201001291.

Tim Clutton-Brock. Breeding Together: Kin Selection and Mutualism in Cooperative Vertebrates. *Science*, 296(5565):69–72, April 2002. doi: 10.1126/science.296.5565.69. URL https://www.science.org/doi/abs/10.1126/science.296.5565.69. Publisher: American Association for the Advancement of Science.

Philip Cohen, Hector Levesque, and Ira Smith. On Team Formation. In *Contemporary Action Theory. Synthese*, pp. 87–114. Kluwer Academic Publishers, 1997.

Sean C.P. Coogan, François-Nicolas Robinne, Piyush Jain, and Mike D. Flannigan. Scientists' warning on wildfire — a Canadian perspective. *Canadian Journal of Forest Research*, 49(9): 1015–1023, September 2019. ISSN 0045-5067. doi: 10.1139/cjfr-2019-0094. URL https://cdnsciencepub.com/doi/10.1139/cjfr-2019-0094. Publisher: NRC Research Press.

Semtech Corporation. LoRa and LoRaWAN: Technical overview | DEVELOPER PORTAL, February 2020. URL https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan/.

Charles Darwin. On the origin of species by means of natural selection, or, The preservation of favoured races in the struggle for life., 1977. URL https://www.loc.gov/item/06017473/.

Keith S. Decker. Distributed problem-solving techniques: A survey. *IEEE Transactions on Systems, Man, & Cybernetics*, 17(5):729–740, 1987. ISSN 0018-9472(Print). doi: 10.1109/TSMC.1987.6499280. Place: US Publisher: Institute of Electrical & Electronics Engineers Inc.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. *arXiv:1606.09375 [cs, stat]*, February 2017. URL http://arxiv.org/abs/1606.09375. arXiv: 1606.09375.

Jakob Foerster, Ioannis Alexandros Assael, Nando de Freitas, and Shimon Whiteson. Learning to Communicate with Deep Multi-Agent Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://papers.nips.cc/paper/2016/hash/c7635bfd99248a2cdef8249ef7bfbef4-Abstract.html.

Matthew C. Fontaine, Ya-Chuan Hsu, Yulun Zhang, Bryon Tjanaka, and Stefanos Nikolaidis. On the Importance of Environments in Human-Robot Coordination. *arXiv:2106.10853 [cs]*, June 2021. URL `http://arxiv.org/abs/2106.10853`. arXiv: 2106.10853.

Hannah Fry. DeepMind: The Podcast - Better together, January 2022. URL `https://podcasts.google.com/feed/aHR0cHM6Ly9mZWVkcy5zaW1wbGVjYXN0LmNvbS9KVDZwYlBrZw`.

Sriram Ganapathi Subramanian and Mark Crowley. Using Spatial Reinforcement Learning to Build Forest Wildfire Dynamics Models From Satellite Images. *Frontiers in ICT*, 5, 2018. ISSN 2297-198X. URL `https://www.frontiersin.org/article/10.3389/fict.2018.00006`.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural Message Passing for Quantum Chemistry. *arXiv:1704.01212 [cs]*, June 2017. URL `http://arxiv.org/abs/1704.01212`. arXiv: 1704.01212.

Michael Goss, Daniel L. Swain, John T. Abatzoglou, Ali Sarhadi, Crystal A. Kolden, A. Park Williams, and Noah S. Diffenbaugh. Climate change is increasing the likelihood of extreme autumn wildfire conditions across California. *Environmental Research Letters*, 15(9):094016, August 2020. ISSN 1748-9326. doi: 10.1088/1748-9326/ab83a7. URL `https://doi.org/10.1088/1748-9326/ab83a7`. Publisher: IOP Publishing.

Carlos Guestrin, Michail Lagoudakis, and Ronald Parr. Coordinated Reinforcement Learning. In *In Proceedings of the ICML-2002 The Nineteenth International Conference on Machine Learning*, pp. 227–234, 2002.

Jayesh K. Gupta, Maxim Egorov, and Mykel Kochenderfer. Cooperative Multi-agent Control Using Deep Reinforcement Learning. In Gita Sukthankar and Juan A. Rodriguez-Aguilar (eds.), *Autonomous Agents and Multiagent Systems*, volume 10642, pp. 66–83. Springer International Publishing, Cham, 2017. ISBN 978-3-319-71681-7 978-3-319-71682-4. doi: 10.1007/978-3-319-71682-4_5. URL `http://link.springer.com/10.1007/978-3-319-71682-4_5`. Series Title: Lecture Notes in Computer Science.

Ravi N. Haksar and Mac Schwager. Distributed Deep Reinforcement Learning for Fighting Forest Fires with a Network of Aerial Robots. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1067–1074, Madrid, October 2018. IEEE. ISBN 978-1-5386-8094-0. doi: 10.1109/IROS.2018.8593539. URL `https://ieeexplore.ieee.org/document/8593539/`.

Pablo Hernandez-Leal, Michael Kaisers, Tim Baarslag, and Enrique Munoz de Cote. A Survey of Learning in Multiagent Environments: Dealing with Non-Stationarity. *arXiv:1707.09183 [cs]*, March 2019. URL `http://arxiv.org/abs/1707.09183`. arXiv: 1707.09183.

John L. Hoogland. Infanticide in Prairie Dogs: Lactating Females Kill Offspring of Close Kin. *Science*, 230(4729):1037–1040, November 1985. doi: 10.1126/science.230.4729.1037. URL `https://www.science.org/doi/10.1126/science.230.4729.1037`. Publisher: American Association for the Advancement of Science.

Fantine Huot, R. Lily Hu, Nita Goyal, Tharun Sankar, Matthias Ihme, and Yi-Fan Chen. Next Day Wildfire Spread: A Machine Learning Data Set to Predict Wildfire Spreading from Remote-Sensing Data. *arXiv:2112.02447 [cs]*, December 2021. URL `http://arxiv.org/abs/2112.02447`. arXiv: 2112.02447.

Zool Hilmi Ismail and Nohaidda Sariff. *A Survey and Analysis of Cooperative Multi-Agent Robot Systems: Challenges and Directions*. IntechOpen, November 2018. ISBN 978-1-78985-756-6. doi: 10.5772/intechopen.79337. URL `https://www.intechopen.com/chapters/63854`. Publication Title: Applications of Mobile Robots.

Max Jaderberg, Michael Mathieu, Nat McAleese, Nathalie Bradley-Schmieg, Nathaniel Wong, Nicolas Porcel, Steph Hughes-Fitt, Valentin Dalibard, and Wojciech Marian Czarnecki. Open-Ended learning leads to Generally Capable Agents. pp. 54, 2021.

Piyush Jain, Sean C.P. Coogan, Sriram Ganapathi Subramanian, Mark Crowley, Steve Taylor, and Mike D. Flannigan. A review of machine learning applications in wildfire science and management. *Environmental Reviews*, 28(4):478–505, December 2020. ISSN 1181-8700, 1208-6053. doi: 10.1139/er-2020-0019. URL https://cdnsciencepub.com/doi/10.1139/er-2020-0019.

Kyle D. Julian and Mykel J. Kochenderfer. Image-based Guidance of Autonomous Aircraft for Wildfire Surveillance and Prediction. *arXiv:1810.02455 [cs]*, March 2019. URL http://arxiv.org/abs/1810.02455. arXiv: 1810.02455.

N. Kaur and D. Garg. Analysis of the Depth First Search Algorithms. *undefined*, 2012. URL https://www.semanticscholar.org/paper/Analysis-of-the-Depth-First-Search-Algorithms-Kaur-Garg/ac85ed7c59e3d43990d0a510eb037ecd07d2b269.

Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement Learning in Robotics: A Survey. *The International Journal of Robotics Research 32(11):1238-1274*, pp. 38, 2013.

Heinrich Küttler, Nantas Nardelli, Alexander H. Miller, Roberta Raileanu, Marco Selvatici, Edward Grefenstette, and Tim Rocktäschel. The NetHack Learning Environment. *arXiv:2006.13760 [cs, stat]*, December 2020. URL http://arxiv.org/abs/2006.13760. arXiv: 2006.13760.

Martin Lauer and Martin Riedmiller. An Algorithm for Distributed Reinforcement Learning in Cooperative Multi-Agent Systems. In *In Proceedings of the Seventeenth International Conference on Machine Learning*, pp. 535–542. Morgan Kaufmann, 2000.

Paulo Leitão and Stamatis Karnouskos. *Industrial Agents: Emerging Applications of Software Agents in Industry*. Elsevier, March 2015. ISBN 978-0-12-800341-1.

Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated Graph Sequence Neural Networks. *arXiv:1511.05493 [cs, stat]*, September 2017. URL http://arxiv.org/abs/1511.05493. arXiv: 1511.05493.

Ryan Lowe, YI WU, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://papers.nips.cc/paper/2017/hash/68a9750337a418a86fe06c1991a1d64c-Abstract.html.

MAJA J. MATARIC. Using communication to reduce locality in distributed multiagent learning. *Journal of Experimental & Theoretical Artificial Intelligence*, 10(3):357–369, July 1998. ISSN 0952-813X. doi: 10.1080/095281398146806. URL https://doi.org/10.1080/095281398146806. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/095281398146806.

Laëtitia Matignon, Guillaume J. Laurent, and Nadine Le Fort-Piat. Independent reinforcement learners in cooperative Markov games: a survey regarding coordination problems. *Knowledge Engineering Review*, 27(1):1–31, March 2012. doi: 10.1017/S026988891200057. URL https://hal.archives-ouvertes.fr/hal-00720669. Publisher: Cambridge University Press (CUP).

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, February 2015. ISSN 1476-4687. doi: 10.1038/nature14236. URL https://www.nature.com/articles/nature14236. Bandiera_abtest: a Cg_type: Nature Research Journals Number: 7540 Primary_atype: Research Publisher: Nature Publishing Group Subject_term: Computer science Subject_term_id: computer-science.

Matej Moravčík, Martin Schmid, Neil Burch, Viliam Lisý, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael Bowling. DeepStack: Expert-Level Artificial

Intelligence in No-Limit Poker. *Science*, 356(6337):508–513, May 2017. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.aam6960. URL http://arxiv.org/abs/1701.01724. arXiv: 1701.01724.

Frans A. Oliehoek. Decentralized POMDPs. In Marco Wiering and Martijn van Otterlo (eds.), *Reinforcement Learning: State-of-the-Art*, Adaptation, Learning, and Optimization, pp. 471–503. Springer, Berlin, Heidelberg, 2012. ISBN 978-3-642-27645-3. doi: 10.1007/978-3-642-27645-3_15. URL https://doi.org/10.1007/978-3-642-27645-3_15.

Spinning Up OpenAI. Proximal Policy Optimization — Spinning Up documentation, 2021. URL https://spinningup.openai.com/en/latest/algorithms/ppo.html.

Liviu Panait and Sean Luke. Cooperative Multi-Agent Learning: The State of the Art. *Autonomous Agents and Multi-Agent Systems*, 11(3):387–434, November 2005. ISSN 1387-2532, 1573-7454. doi: 10.1007/s10458-005-2631-2. URL http://link.springer.com/10.1007/s10458-005-2631-2.

Ken Perlin. An image synthesizer. *ACM SIGGRAPH Computer Graphics*, 19(3):287–296, July 1985. ISSN 0097-8930. doi: 10.1145/325165.325247. URL https://doi.org/10.1145/325165.325247.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. DeepWalk: Online Learning of Social Representations. *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 701–710, August 2014. doi: 10.1145/2623330.2623732. URL http://arxiv.org/abs/1403.6652. arXiv: 1403.6652.

Emanuele Pesce and Giovanni Montana. Improving Coordination in Small-Scale Multi-Agent Deep Reinforcement Learning through Memory-driven Communication. *Machine Learning*, 109(9-10): 1727–1747, September 2020. ISSN 0885-6125, 1573-0565. doi: 10.1007/s10994-019-05864-5. URL http://arxiv.org/abs/1901.03887. arXiv: 1901.03887.

Leonid Peshkin, Kee-Eung Kim, Nicolas Meuleau, and Leslie Pack Kaelnling. Learning to Cooperate via Policy Search, 2000. URL https://arxiv.org/ftp/arxiv/papers/1408/1408.1484.pdf.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 2009.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms. *arXiv:1707.06347 [cs]*, August 2017. URL http://arxiv.org/abs/1707.06347. arXiv: 1707.06347.

Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. Safe, Multi-Agent, Reinforcement Learning for Autonomous Driving. *arXiv:1610.03295 [cs, stat]*, October 2016. URL http://arxiv.org/abs/1610.03295. arXiv: 1610.03295.

Philipp Dominic Siedler. The Power of Communication in a Distributed Multi-Agent System. *arXiv:2111.15611 [cs]*, December 2021. URL http://arxiv.org/abs/2111.15611. arXiv: 2111.15611.

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, January 2016. ISSN 1476-4687. doi: 10.1038/nature16961. URL https://www.nature.com/articles/nature16961. Bandiera_abtest: a Cg_type: Nature Research Journals Number: 7587 Primary_atype: Research Publisher: Nature Publishing Group Subject_term: Computational science;Computer science;Reward Subject_term_id: computational-science;computer-science;reward.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis.

Mastering the game of Go without human knowledge. *Nature*, 550(7676):354–359, October 2017. ISSN 1476-4687. doi: 10.1038/nature24270. URL https://www.nature.com/articles/nature24270. Bandiera_abtest: a Cg_type: Nature Research Journals Number: 7676 Primary_atype: Research Publisher: Nature Publishing Group Subject_term: Computational science;Computer science;Reward Subject_term_id: computational-science;computer-science;reward.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 362(6419):1140–1144, December 2018. doi: 10.1126/science.aar6404. URL https://www.science.org/doi/10.1126/science.aar6404. Publisher: American Association for the Advancement of Science.

Sriram Subramanian and Mark Crowley. Learning Forest Wildfire Dynamics from Satellite Images Using Reinforcement Learning. January 2017.

Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. Learning Multiagent Communication with Backpropagation. *arXiv:1605.07736 [cs]*, October 2016. URL http://arxiv.org/abs/1605.07736. arXiv: 1605.07736.

Richard S Sutton and Andrew G Barto. Reinforcement Learning: An Introduction. pp. 352, 2015.

Udacity-DeepRL. An Introduction to Proximal Policy Optimization (PPO) in Deep Reinforcement Learning, April 2019. URL https://www.youtube.com/watch?v=vQ_ifavFBkI.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. *arXiv:1710.10903 [cs, stat]*, February 2018. URL http://arxiv.org/abs/1710.10903. arXiv: 1710.10903.

Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, Laurent Sifre, Trevor Cai, John P. Agapiou, Max Jaderberg, Alexander S. Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom L. Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, November 2019. ISSN 1476-4687. doi: 10.1038/s41586-019-1724-z. URL https://www.nature.com/articles/s41586-019-1724-z. Number: 7782 Primary_atype: Research Publisher: Nature Publishing Group Subject_term: Computer science;Statistics Subject_term_id: computer-science;statistics.

Haohan Wang and Bhiksha Raj. On the Origin of Deep Learning. *arXiv:1702.07800 [cs, stat]*, March 2017. URL http://arxiv.org/abs/1702.07800. arXiv: 1702.07800.

Tom Ward, Andrew Bolt, Nik Hemmings, Simon Carter, Manuel Sanchez, Ricardo Barreira, Seb Noury, Keith Anderson, Jay Lemmon, Jonathan Coe, Piotr Trochim, Tom Handley, and Adrian Bolton. Using Unity to Help Solve Intelligence. *arXiv:2011.09294 [cs]*, November 2020. URL http://arxiv.org/abs/2011.09294. arXiv: 2011.09294.

Yujian Xiong, Jie Wu, and Zizhan Chen. Machine Learning Wildfire Prediction based on Climate Data. pp. 8, 2020.

Ping Xuan, Victor Lesser, and Shlomo Zilberstein. Communication decisions in multi-agent cooperation: model and experiments. In *Proceedings of the fifth international conference on Autonomous agents*, AGENTS '01, pp. 616–623, New York, NY, USA, May 2001. Association for Computing Machinery. ISBN 978-1-58113-326-4. doi: 10.1145/375735.376469. URL https://doi.org/10.1145/375735.376469.

Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms. *arXiv:1911.10635 [cs, stat]*, April 2021. URL http://arxiv.org/abs/1911.10635. arXiv: 1911.10635.

Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020. ISSN 26666510. doi: 10.1016/j.aiopen.2021.01.001. URL https://linkinghub.elsevier.com/retrieve/pii/S2666651021000012.

Shaked Zychlinski. The Complete Reinforcement Learning Dictionary, November 2019. URL https://towardsdatascience.com/the-complete-reinforcement-learning-dictionary-e16230b7d24e.

A Appendix

B Hyperparameters

B.1 Single Agent Training Hyperparameters

```
behaviors:
  WT_SA:
    trainer_type: ppo
    hyperparameters:
      batch_size: 128
      buffer_size: 2048
      learning_rate: 0.0003
      beta: 0.01
      epsilon: 0.2
      lambd: 0.95
      num_epoch: 3
      learning_rate_schedule: linear
    network_settings:
      normalize: false
      hidden_units: 512
      num_layers: 2
      vis_encode_type: simple
    reward_signals:
      extrinsic:
        gamma: 0.99
        strength: 1.0
      curiosity:
        gamma: 0.99
        strength: 0.02
        encoding_size: 256
        learning_rate: 0.0003
    keep_checkpoints: 5
    max_steps: 50000000
    time_horizon: 128
    summary_freq: 40500
    threaded: true
```

B.2 Multi Agent Training Hyperparameters

```
behaviors:
  WT_MA:
    trainer_type: ppo
    hyperparameters:
      batch_size: 128
      buffer_size: 2048
      learning_rate: 0.0003
      beta: 0.01
      epsilon: 0.2
      lambd: 0.95
      num_epoch: 3
      learning_rate_schedule: linear
    network_settings:
      normalize: false
      hidden_units: 512
      num_layers: 2
      vis_encode_type: simple
    reward_signals:
      extrinsic:
        gamma: 0.99
        strength: 1.0
      curiosity:
        gamma: 0.99
        strength: 0.02
        encoding_size: 256
        learning_rate: 0.0003
    keep_checkpoints: 5
    max_steps: 5000000
    time_horizon: 128
    summary_freq: 24300
    threaded: true
```

B.3 Multi Agent Auto Curriculum Training Hyperparameters

```
behaviors:
  WT_MA:
    trainer_type: ppo
    hyperparameters:
      batch_size: 128
      buffer_size: 2048
      learning_rate: 0.0003
      beta: 0.01
      epsilon: 0.2
      lambd: 0.95
      num_epoch: 3
      learning_rate_schedule: linear
    network_settings:
      normalize: false
      hidden_units: 512
      num_layers: 2
      vis_encode_type: simple
    reward_signals:
      extrinsic:
        gamma: 0.99
        strength: 1.0
      curiosity:
        gamma: 0.99
        strength: 0.02
        encoding_size: 256
        learning_rate: 0.0003
    keep_checkpoints: 5
    max_steps: 100000000
    time_horizon: 128
    summary_freq: 40500
    threaded: true

environment_parameters:
  difficulty:
```

```
curriculum:
  - name: Lesson1
    completion_criteria:
      measure: reward
      behavior: WT_MA
      signal_smoothing: true
      min_lesson_length: 100
      threshold: 900
    value: 1
  - name: Lesson2
    completion_criteria:
      measure: reward
      behavior: WT_MA
      signal_smoothing: true
      min_lesson_length: 100
      threshold: 950
    value: 2
  - name: Lesson3
    completion_criteria:
      measure: reward
      behavior: WT_MA
      signal_smoothing: true
      min_lesson_length: 100
      threshold: 1000
    value: 3
  - name: Lesson4
    completion_criteria:
      measure: reward
      behavior: WT_MA
      signal_smoothing: true
      min_lesson_length: 100
      threshold: 1050
    value: 4
  - name: Lesson5
    completion_criteria:
      measure: reward
      behavior: WT_MA
      signal_smoothing: true
      min_lesson_length: 100
      threshold: 1100
    value: 5
  - name: Lesson6
    completion_criteria:
      measure: reward
      behavior: WT_MA
      signal_smoothing: true
      min_lesson_length: 100
      threshold: 1150
    value: 6
  - name: Lesson7
    completion_criteria:
      measure: reward
      behavior: WT_MA
      signal_smoothing: true
      min_lesson_length: 100
      threshold: 1200
    value: 7
  - name: Lesson8
    completion_criteria:
      measure: reward
      behavior: WT_MA
      signal_smoothing: true
      min_lesson_length: 100
      threshold: 1250
    value: 8
  - name: Lesson9
    completion_criteria:
      measure: reward
      behavior: WT_MA
      signal_smoothing: true
      min_lesson_length: 100
      threshold: 1300
    value: 9
  - name: Lesson10
    value: 10
```

B.4 Hyperparameter Description

Hyperparameter	Typical Range	Description
Gamma	$0.8 - 0.995$	discount factor for future rewards
Lambda	$0.9 - 0.95$	used when calculating the Generalized Advantage Estimate (GAE)
Buffer Size	$2048 - 409600$	how many experiences should be collected before updating the model
Batch Size	$512-5120$ (continuous), $32-512$ (discrete)	number of experiences used for one iteration of a gradient descent update.
Number of Epochs	$3 - 10$	number of passes through the experience buffer during gradient descent
Learning Rate	$1e - 5 - 1e - 3$	strength of each gradient descent update step
Time Horizon	$32 - 2048$	number of steps of experience to collect per-agent before adding it to the experience buffer
Max Steps	$5e5 - 1e7$	number of steps of the simulation (multiplied by frame-skip) during the training process
Beta	$1e - 4 - 1e - 2$	strength of the entropy regularization, which makes the policy "more random"
Epsilon	$0.1 - 0.3$	acceptable threshold of divergence between the old and new policies during gradient descent updating
Normalize	$true/false$	weather normalization is applied to the vector observation inputs
Number of Layers	$1 - 3$	number of hidden layers present after the observation input
Hidden Units	$32 - 512$	number of units in each fully connected layer of the neural network
Intrinsic Curiosity Module		
Curiosity Encoding Size	$64 - 256$	size of hidden layer used to encode the observations within the intrinsic curiosity module
Curiosity Strength	$0.1 - 0.001$	magnitude of the intrinsic reward generated by the intrinsic curiosity module

C Pseudocode

PPO-CLIP pseudocode (OpenAI, 2021; Schulman et al., 2017):

Algorithm 1 PPO-Clip

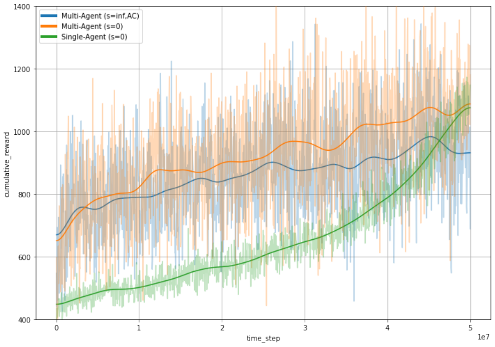
```
Input: initial policy parameters $\theta_0$, initial value function parameters $\phi_0$
**for** $k = 0, 1, 2, \ldots$ **do**
    Collect set of trajectories $\mathcal{D}_k$ = $\{\tau_i\}$ by running policy $\pi_k = \pi(\theta_k)$ in the environment.
    Compute rewards-to-go $\hat{R}_t$.
    Compute advantage estimates, $\hat{A}_t$ (using any method of advantage estimation) based on the
    current value function $V_{\phi_k}$
    Update the policy by maximizing the PPO-Clip objective:
    $\theta_{k+1} = \underset{\theta}{argmax} \frac{1}{|\mathcal{D}_k|T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \min \left( \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)} A^{\pi_{\theta_k}}(s_t, a_t), g(\epsilon, A^{\pi_{\theta_k}}(s_t, a_t)) \right)$,
    typically via stochastic gradient ascent with Adam.
    Fit value function by regression on mean-squared error:
    $\phi_{k+1} = \underset{\phi}{argmin} \frac{1}{|\mathcal{D}_k|T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \left( (V_\phi(s_t) - \hat{R}_t \right)$
    typically via some gradient descent algorithm.
**end for**
```

Simple Multi-Agent PPO pseudocode:

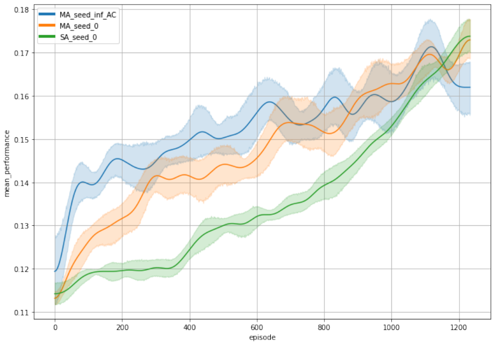
Algorithm 2 Multi-Agent PPO

```
**for** $iteration = 1, 2, \ldots$ **do**
    **for** $actor = 1, 2, \ldots, N$ **do**
        Run policy $\pi_{\theta_{old}}$ in environment for $T$ time steps
        Compute advantage estimates $\hat{A}_1, \ldots, \hat{A}_T$
    **end for**
    Optimize surrogate $L$ wrt. $\theta$, with $K$ epochs and minibatch size $M \leq NT$
    $\theta_{old} \leftarrow \theta$
**end for**
```

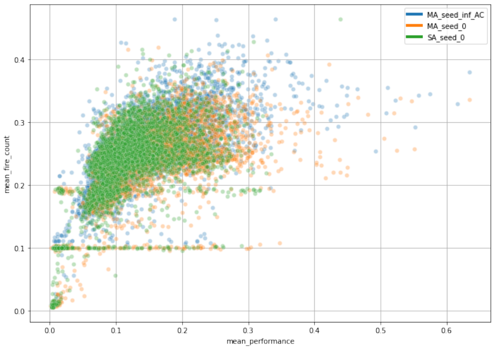
D Training Data Visualisation



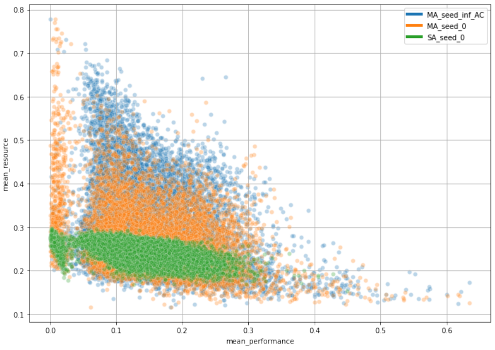
(a) cumulative rewards vs time step



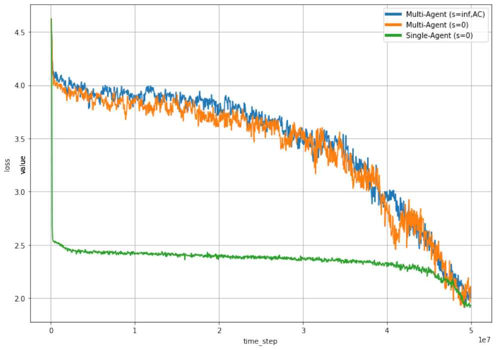
(b) mean performance vs episode



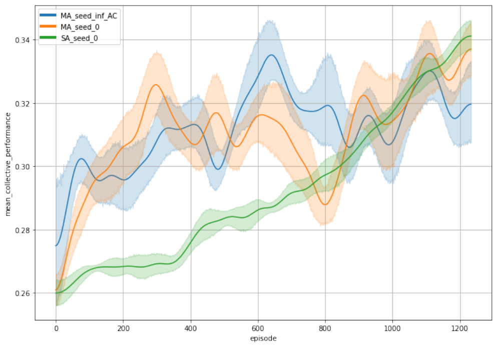
(c) mean fire count vs mean performance



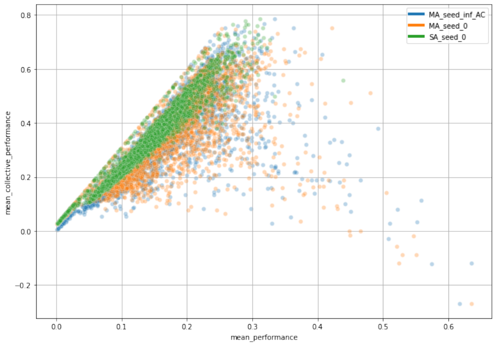
(d) mean resource vs mean performance



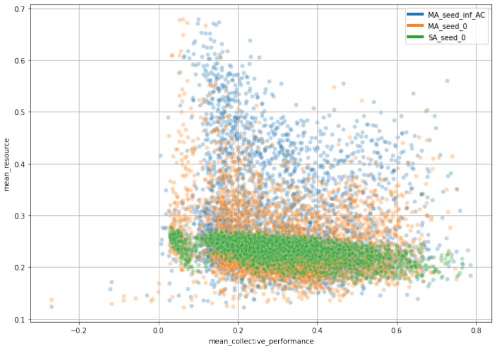
(e) loss vs time step



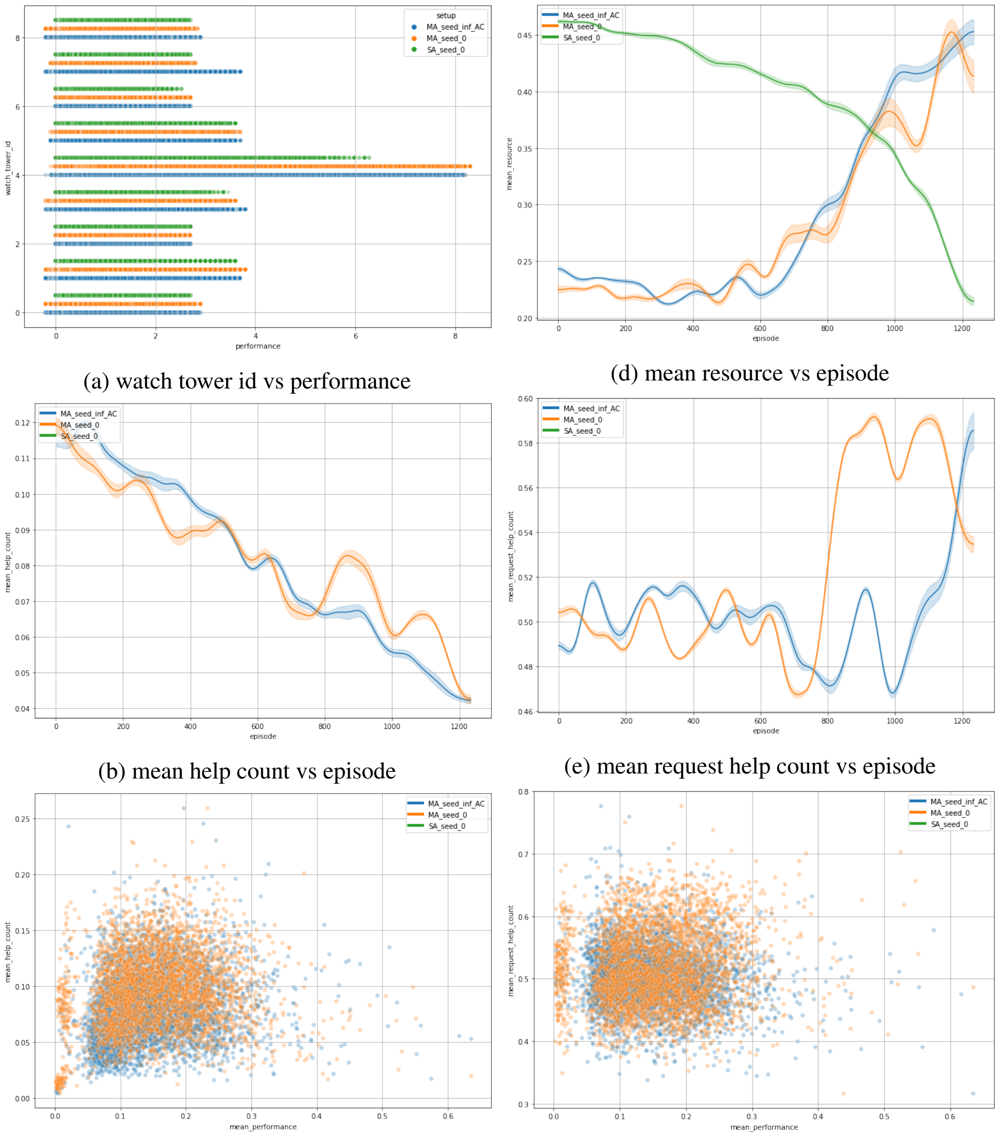
(f) mean collective performance vs episode



(g) m. collective performance vs mean performance



(h) mean resource vs mean collective performance



(a) watch tower id vs performance

(d) mean resource vs episode

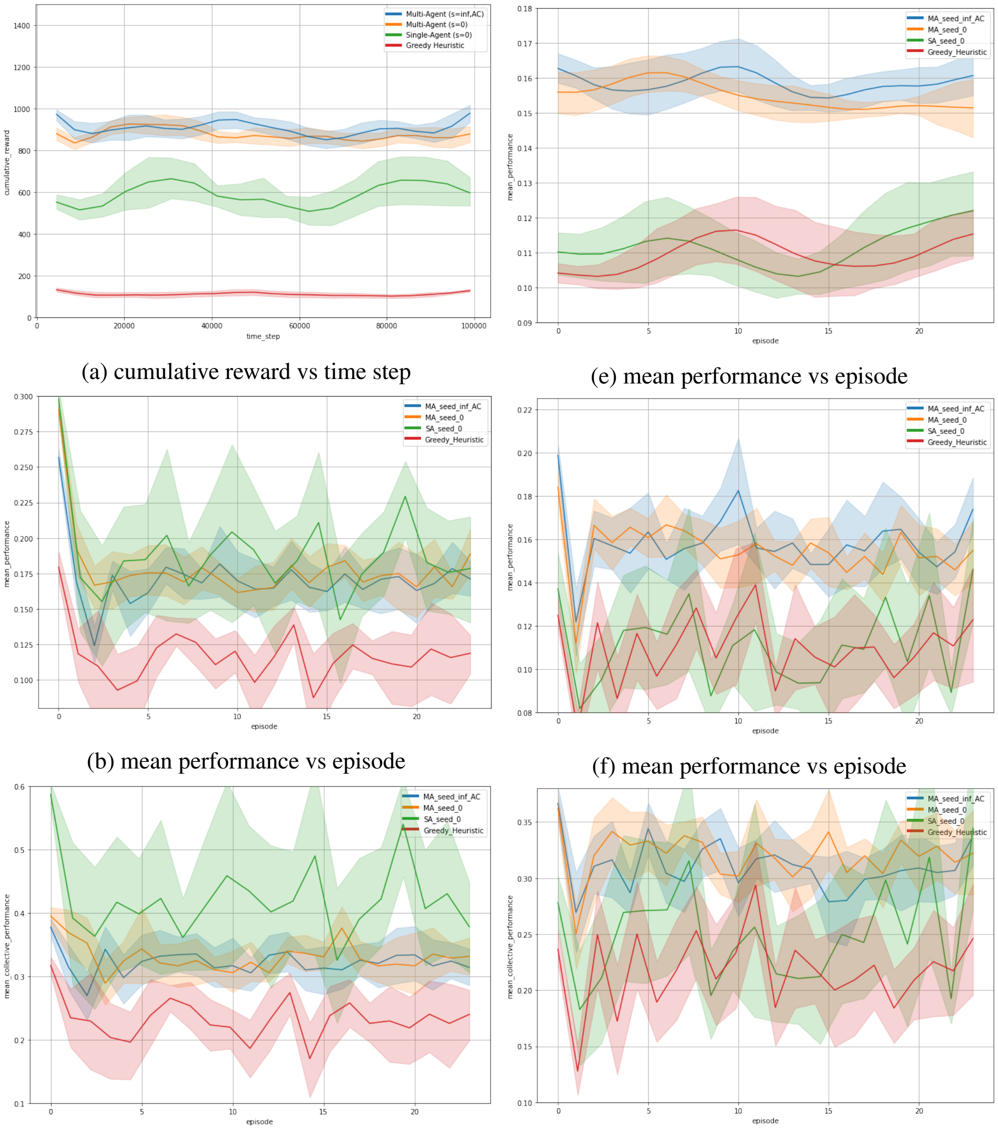
(b) mean help count vs episode

(e) mean request help count vs episode

(c) mean help count vs mean performance

(f) mean request help count vs mean performance

E Inference Data Visualisation (Left Column: Seed = 0, Right Column: Seed = inf, Auto-Curriculum; starting from figure 10 (b)



(a) cumulative reward vs time step

(e) mean performance vs episode

(b) mean performance vs episode

(f) mean performance vs episode

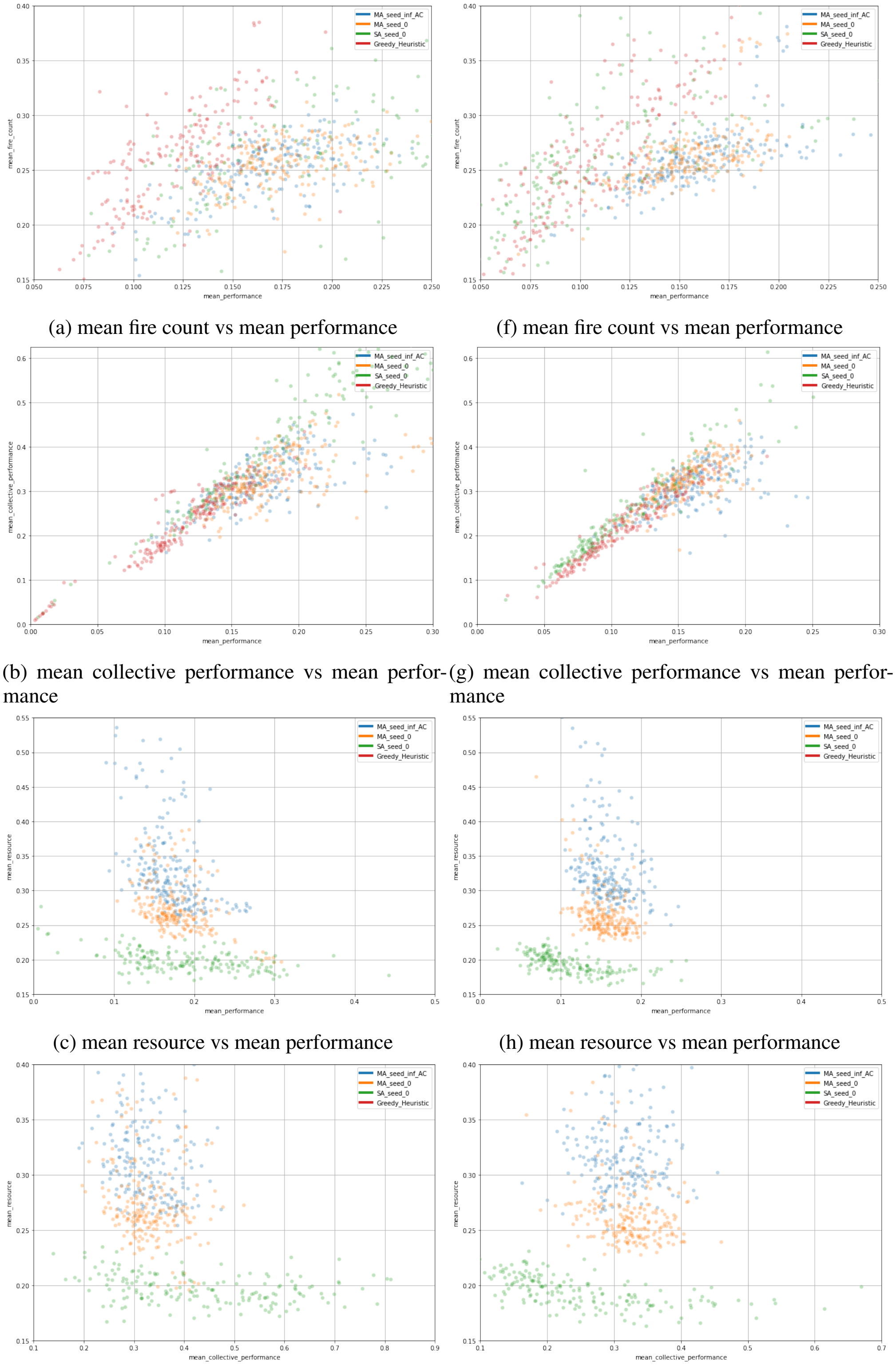
(c) mean collective performance vs episode

(g) mean collective performance vs episode

(d) Seed = 0

(h) Seed = inf, Auto-Curriculum

Figure 10



(a) mean fire count vs mean performance

(f) mean fire count vs mean performance

(b) mean collective performance vs mean performance

(g) mean collective performance vs mean performance

(c) mean resource vs mean performance

(h) mean resource vs mean performance

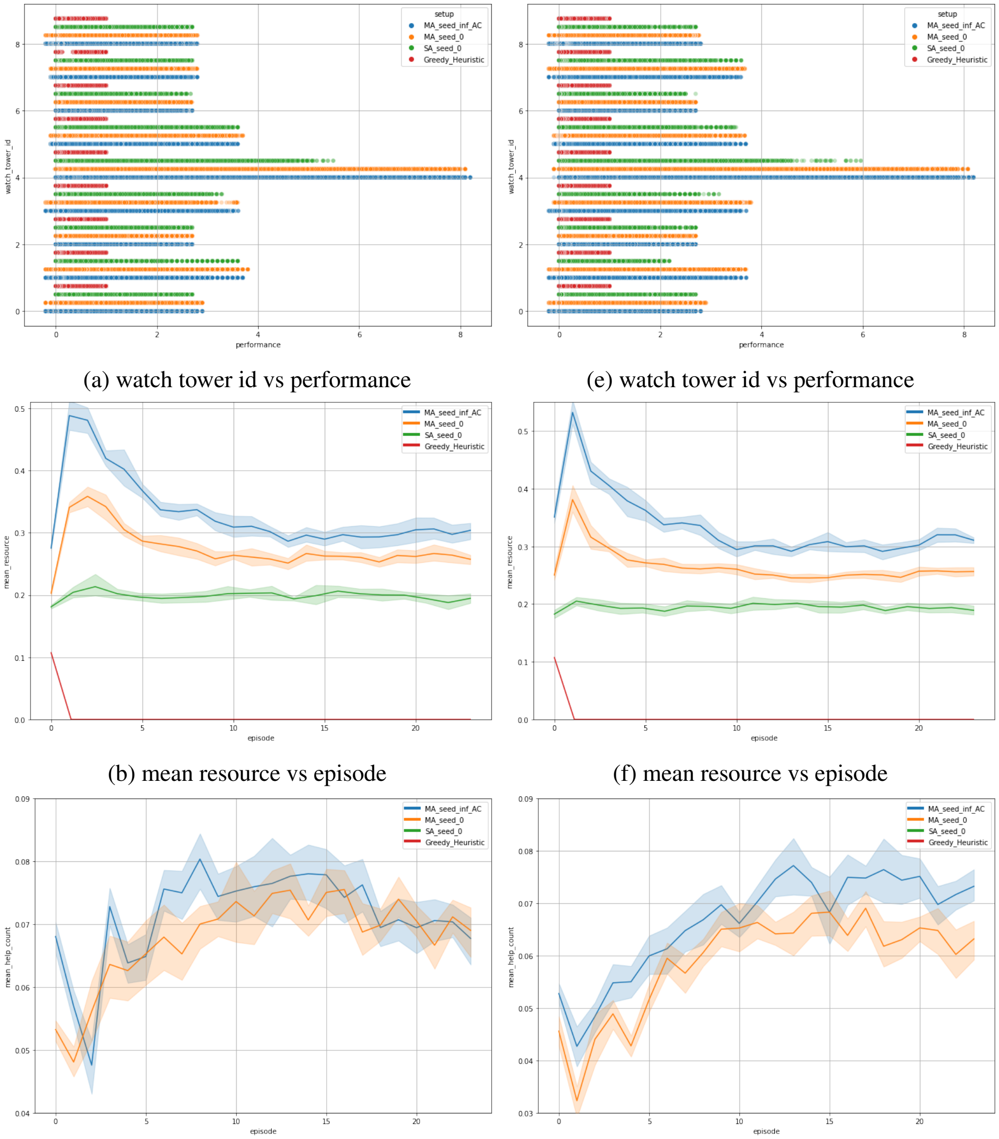
(d) mean resource vs mean collective performance

(i) mean resource vs mean collective performance

(e) Seed = 0

(j) Seed = inf, Auto-Curriculum

Figure 11



(a) watch tower id vs performance

(e) watch tower id vs performance

(b) mean resource vs episode

(f) mean resource vs episode

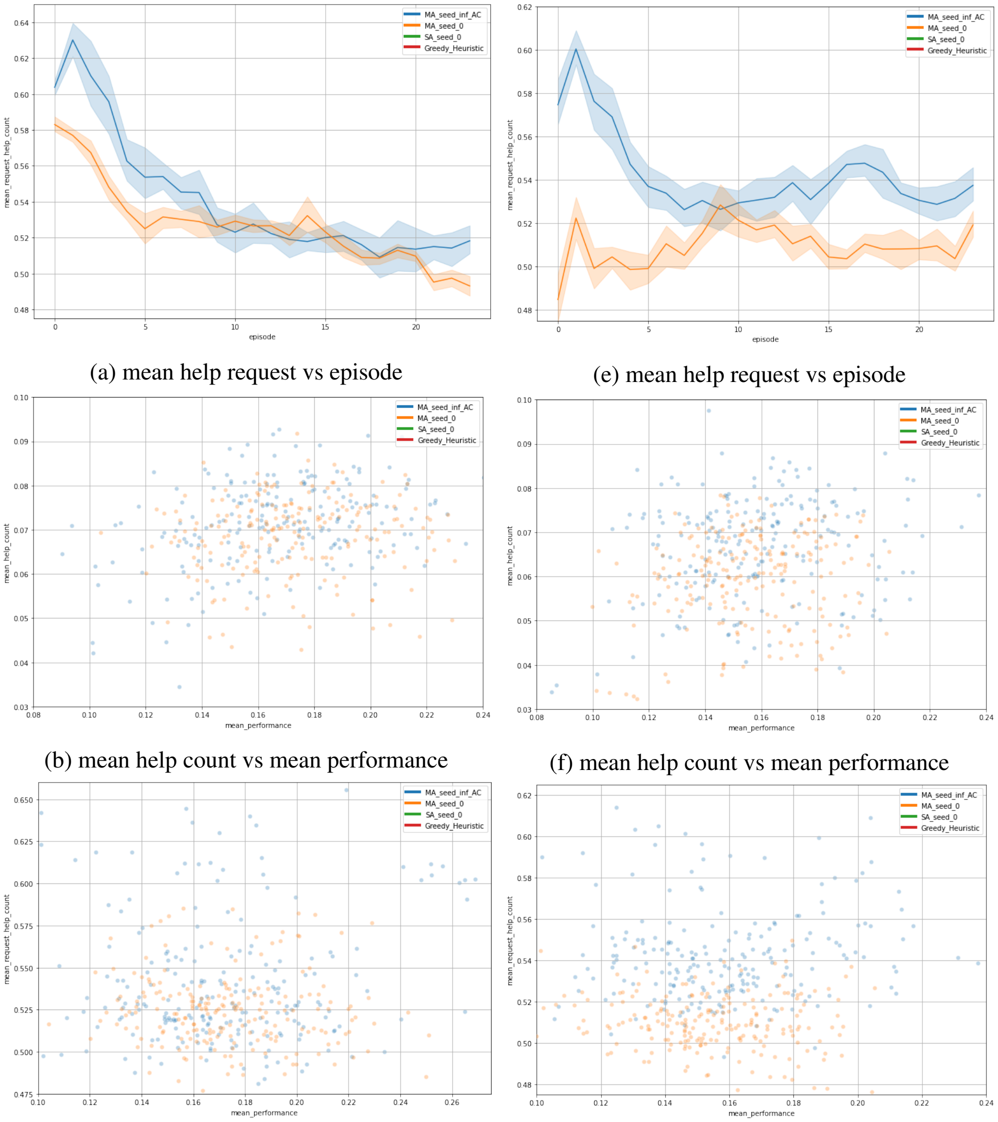
(c) mean help count vs episode

(g) mean help count vs episode

(d) Seed = 0

(h) Seed = inf, Auto-Curriculum

Figure 12



(a) mean help request vs episode

(e) mean help request vs episode

(b) mean help count vs mean performance

(f) mean help count vs mean performance

(c) mean request help count vs mean performance

(g) mean request help count vs mean performance

(d) Seed = 0

(h) Seed = inf, Auto-Curriculum

Figure 13

F Environment Scenario Samples: Difficulty vs. Seed Matrix

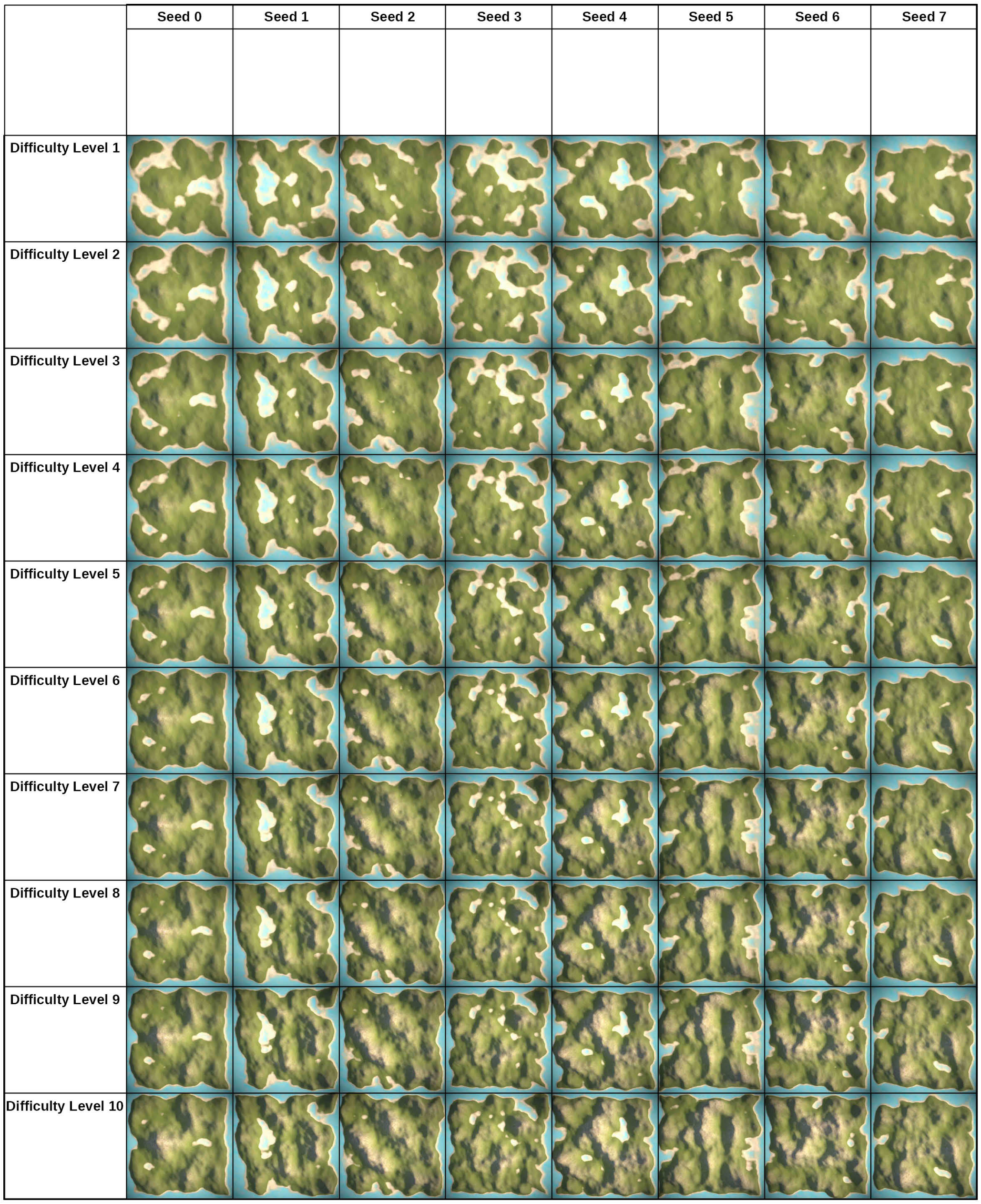


Figure 14

G Multi-Agent Communication

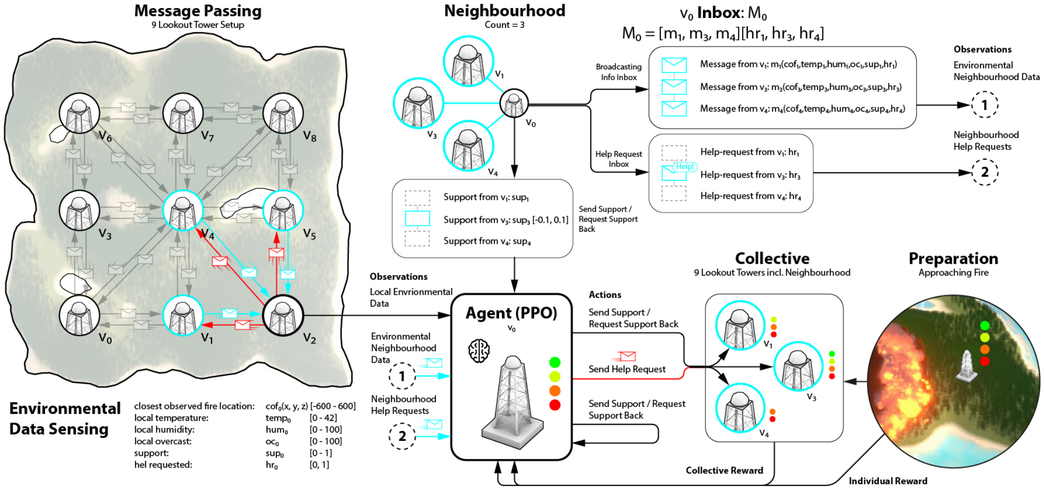

Message Passing
9 Lookout Tower Setup
V6
V7
V8
V3
V4
V5
V0
V1
V2
Environmental Data Sensing
closest observed fire location: cof0(x, y, z) [-600 - 600]
local temperature: temp0 [0 - 42]
local humidity: hum0 [0 - 100]
local overcast: oc0 [0 - 100]
support: sup0 [0 - 1]
hel requested: hr0 [0, 1]
Neighbourhood
Count = 3
v1
v3
v4
v0
v0 Inbox: M0
M0 = [m1, m3, m4][hr1, hr3, hr4]
Broadcasting Info Inbox
Message from v1: m1(cof1,temp1,hum1,oc1,sup1,hr1)
Message from v3: m3(cof3,temp3,hum3,oc3,sup3,hr3)
Message from v4: m4(cof4,temp4,hum4,oc4,sup4,hr4)
Observations
Environmental Neighbourhood Data
1
Help Request Inbox
Help!
Help-request from v1: hr1
Help-request from v3: hr3
Help-request from v4: hr4
Neighbourhood Help Requests
2
Support from v1: sup1
Support from v3: sup3 [-0.1, 0.1]
Support from v4: sup4
Send Support / Request Support Back
Observations
Local Envrionmental Data
Environmental Neighbourhood Data
1
Neighbourhood Help Requests
2
Agent (PPO)
v0
Actions
Send Support / Request Support Back
Send Help Request
Send Support / Request Support Back
Collective
9 Lookout Towers incl. Neighbourhood
v1
v3
v4
Preparation
Approaching Fire
Collective Reward
Individual Reward

Figure 15

H Egoistic Reward Function Diagram

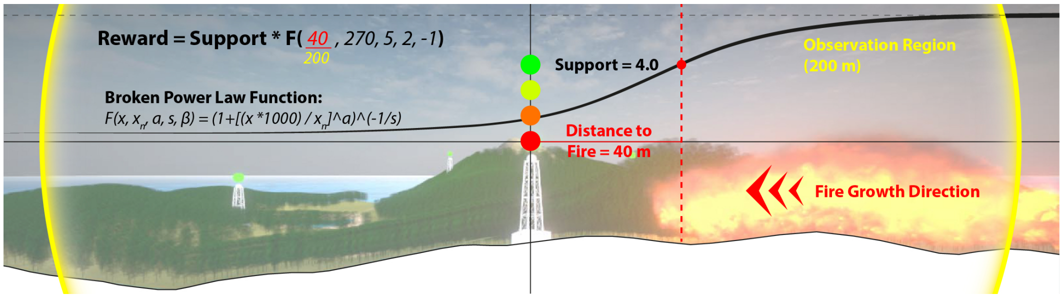
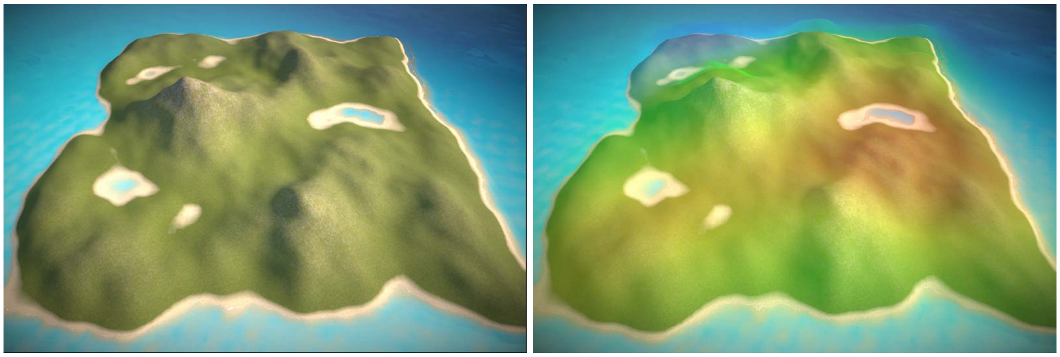

Reward = Support * F(40/200 , 270, 5, 2, -1)
Broken Power Law Function:
F(x, x_n, a, s, β) = (1+[(x*1000)/x_n]^a)^(-1/s)
Support = 4.0
Distance to Fire = 40 m
Observation Region (200 m)
Fire Growth Direction

Figure 16

I Static and Dynamic Environment Features



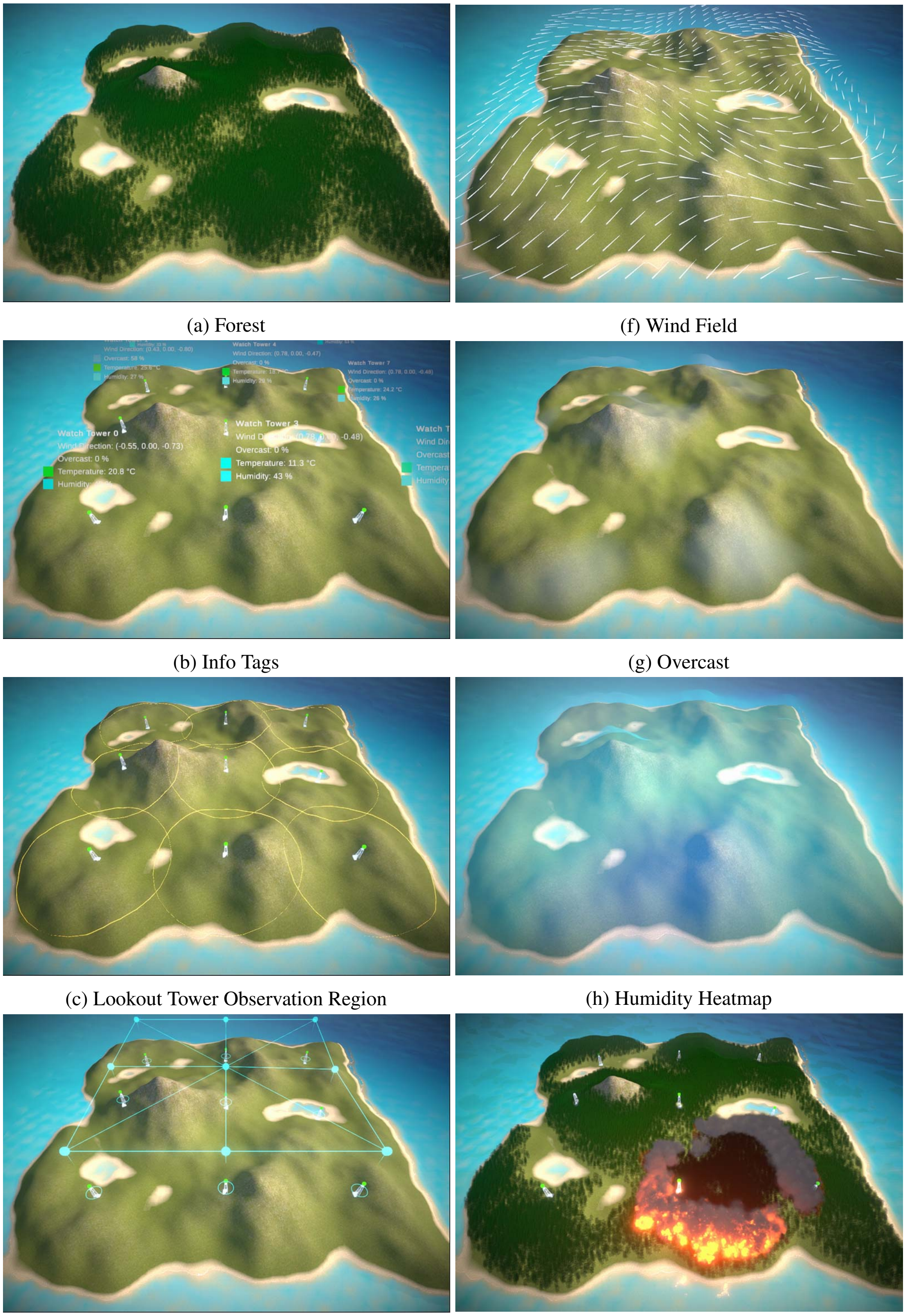
(a) Terrain

(c) Temperature Heatmap

(b) Static Features

(d) Dynamic Features

Figure 17



(a) Forest

(b) Info Tags

(c) Lookout Tower Observation Region

(d) Neighbourhood Network

(e) Static Features

(f) Wind Field

(g) Overcast

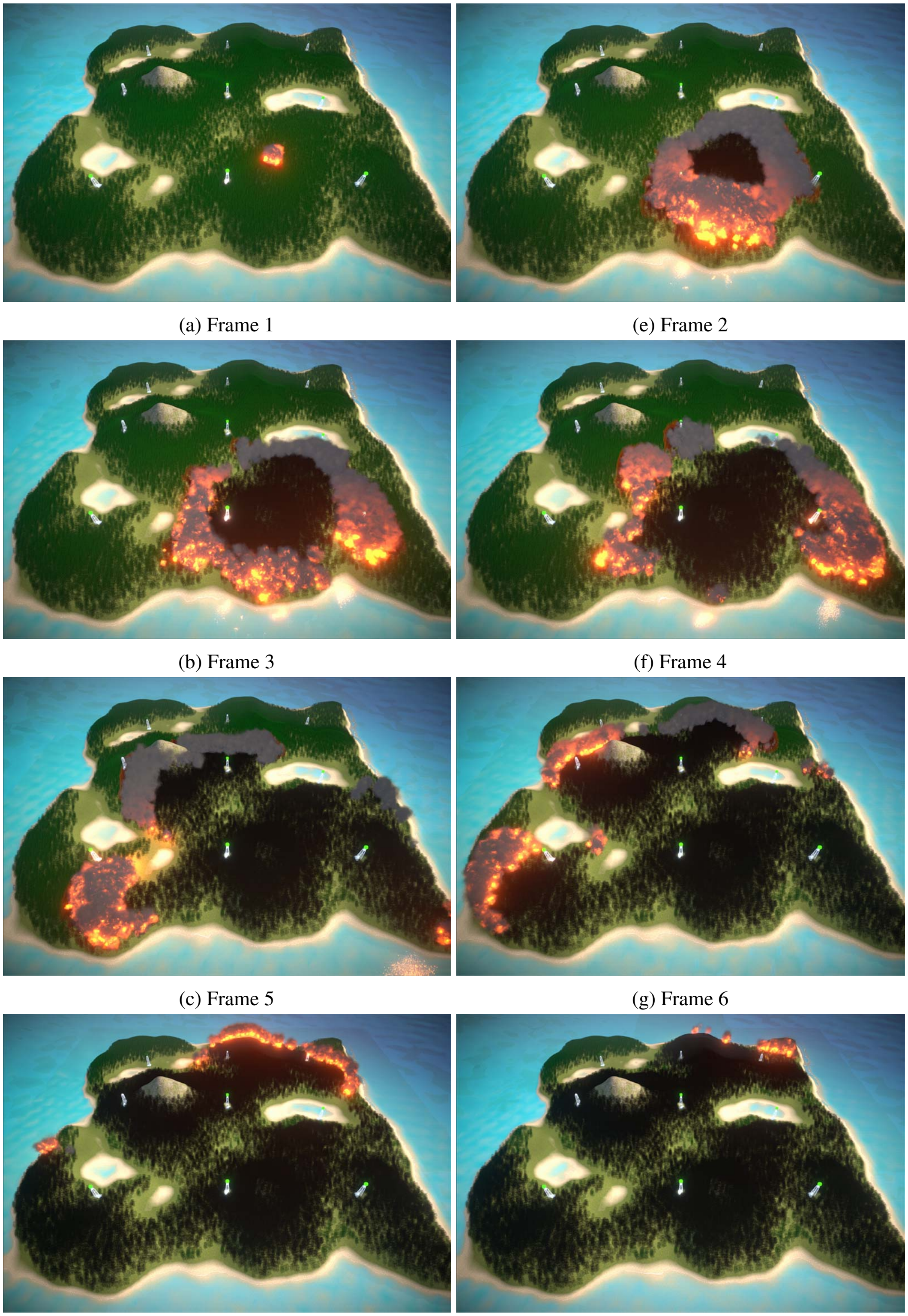
(h) Humidity Heatmap

(i) Wild Fire

(j) Dynamic Features

Figure 18

J WILD FIRE GROWTH BEHAVIOUR FRAMES



(a) Frame 1

(e) Frame 2

(b) Frame 3

(f) Frame 4

(c) Frame 5

(g) Frame 6

(d) Frame 7

(h) Frame 8

Figure 19

K Training School Environment Screenshots

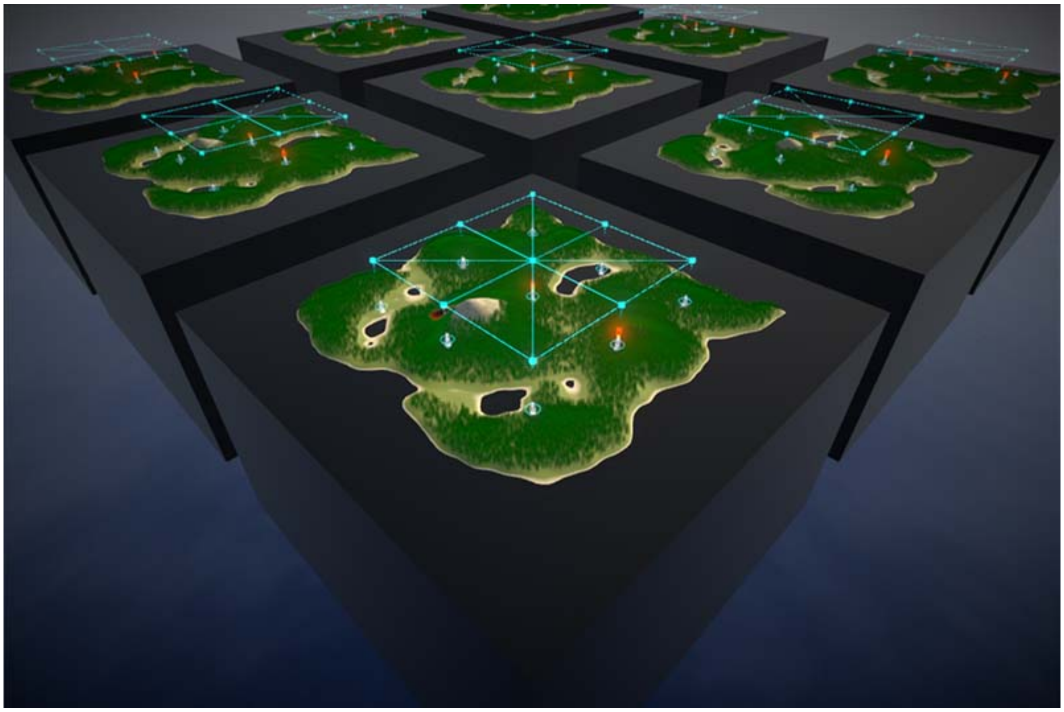


Figure 20: Multi-Agent Training School

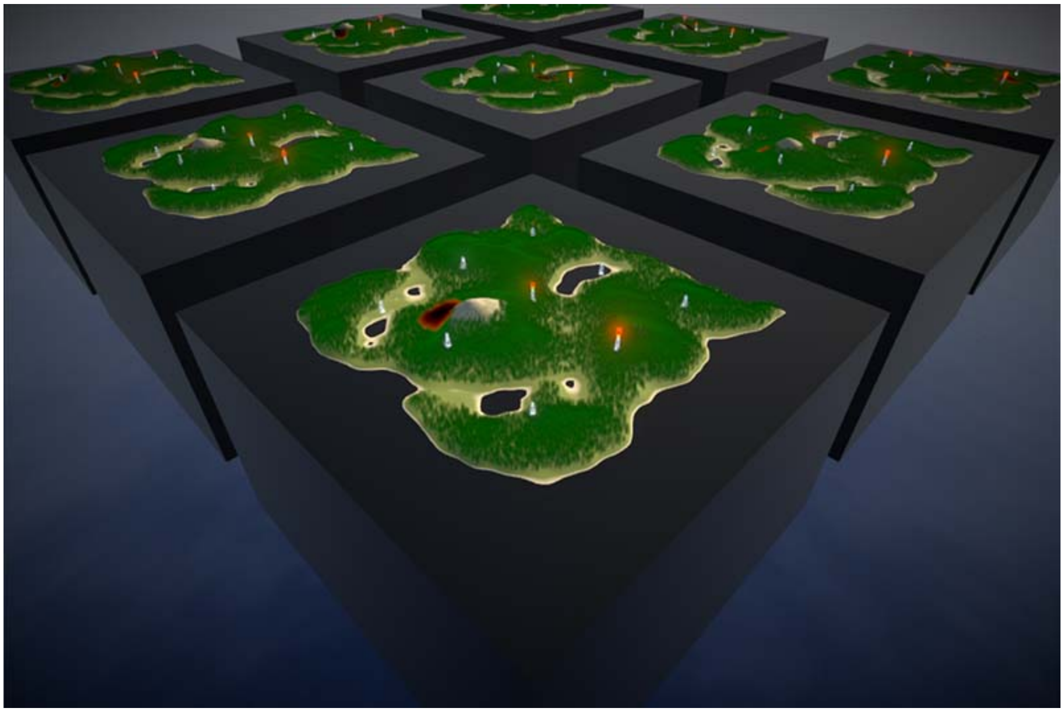


Figure 21: Single-Agent Training School