Benchmarking Robustness and Generalization in Multi-Agent Systems: A Case Study on Neural MMO

ABSTRACT

We present the results of the second Neural MMO challenge, hosted at IJCAI 2022, which received 1600+ submissions. This competition targets robustness and generalization in multi-agent systems: participants train teams of agents to complete a multi-task objective against opponents not seen during training. The competition combines relatively complex environment design with large numbers of agents in the environment. The top submissions demonstrate strong success on this task using mostly standard reinforcement learning (RL) methods combined with domain-specific engineering. We summarize the competition design and results and suggest that, as an academic community, competitions may be a powerful approach to solving hard problems and establishing a solid benchmark for algorithms. We open-source our benchmark including the environment wrapper, baselines, a visualization tool, and selected policies for further research.

KEYWORDS

Multi-agent Reinforcement Learning, Benchmark, Competition

ACM Reference Format:

. . Benchmarking Robustness and Generalization in Multi-Agent Systems: A Case Study on Neural MMO. In , , , IFAAMAS, 9 pages.

1 INTRODUCTION

Real-world applications of reinforcement learning (RL) require robust algorithms [3] that can adapt to dynamic environments. While substantially studied in single-agent RL [2, 13], this subject has been less explored in multi-agent systems. This is of particular importance to multi-agent RL (MARL) algorithms because learned policies must adapt to changes in other agents' behaviors in addition to changes in the environment. Here we suggest three difficulties for establishing benchmarks in multi-agent systems that should resonate with MARL researchers:

(1) **Lack of environments:** while there are many single-agent environments of varying complexities that are standard, efficient, and simple to use, few multi-agent environments satisfy all three of these properties.
(2) **Lack of infrastructure:** most RL libraries and interfaces are intended for single-agent systems, but multi-agent training requires scalability, flexibility, and other additional features. For example, an accurate skill-rating system is needed for multi-agent evaluation as performance is relative to other agents.
(3) **Lack of domain-specific optimization:** minor implementation details and domain-specific tricks like feature engineering often highly influence the final performance of RL algorithms [6]. Although these techniques are not the focus of academic research, without them, it is hard to identify the roots of progress and benchmark algorithms fairly.

This paper summarizes the IJCAI 2022 Neural MMO challenge and offers a solution to these three problems. Neural MMO is a good environment to start with because it supports large-scale populations, is computationally efficient and is actively maintained. In addition to the environment, we built a large-scale parallel evaluation tool and a TrueSkill[8] rating system on the AICrowd platform as the infrastructure. The competition among participants provides an inherent incentive for domain-specific optimization, which is often overlooked in academic research. We hope that our methodology can serve as a stepping stone towards establishing more general benchmarks and promoting future research in Neural MMO and other multi-agent systems. Our main contributions are:

(1) **Orchestration:** we detail the structure of our competition, including the environment, resources, the design of tracks, and the evaluation system. While RL competitions are gaining popularity, few resources exist on how to design a good competition. We believe this will be useful to guide future RL competitions.
(2) **Insights:** we analyze the emergent behaviors and strategies over the 1600+ submissions received and provide insights about the dynamics of the unique multi-agent system consisting of different participants. For example, we find an interesting arms race between rule-based methods and learning-based methods.
(3) **Policy Pool:** we release a pool of 20 submitted policies to promote future research on Neural MMO. The policy pool is diverse, containing both rule-based and RL-based implementations of aggressive and conservative strategies. This will be useful in evaluating policy robustness against a variety of opponents.

2 RELATED WORKS

2.1 Environments and Benchmarks

In recent years, wrapping existing games as environments has been a popular approach to increase complexity and promote novel algorithms. MineRL[17] uses Minecraft to highlight hard problems such as hierarchical task structure and sparse rewards. The NetHack Learning Environment [11] provides a rich and challenging environment focused on the problems of exploration and skill acquisition while allowing fast simulation. ProcGen [2] uses 16 procedurally-generated gym environments and is designed to benchmark both sample efficiency and generalization in reinforcement learning.

Among multi-agent environments, the most commonly used is the StarCraft Multi-agent Challenge (SMAC) [15] from the game StarCraft2. SMAC is mainly intended to investigate algorithms for multi-agent cooperation. Google Research Football (gfootball) [9] uses a physics-based 3D football environment in multi-player and multi-agent scenarios, proposed to benchmark algorithms on the sparse reward and multi-agent cooperation. Neural MMO [18] proposes an open-ended Massively Multiplayer Online (MMO) environment with up to 1024 agents to study robustness and teamwork

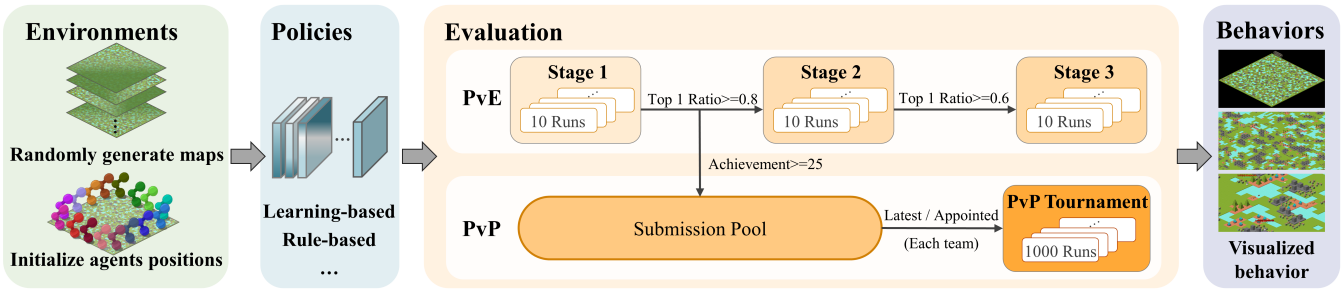

Figure 1: Evaluation structure of the competition. Submitted policies are evaluated in two stages. In the PvE track, there are multiple qualifying rounds against increasingly difficult built-in opponents, and participants receive feedback within minutes of submitting. The PvP track features weekly tournaments to determine the relative skill of all qualified submissions.

in a massive-agent environment. Agar.io [20] and the similar Go-Bigger [1] uses the popular online multi-player game Agar.io[1] for multi-agent cooperation and competition. We use Neural MMO as our competition environment because it supports large-scale population simulation with up to 16 teams in one environment.

2.2 RL Competitions

Several works [5] attempt to establish solid benchmarks for deep RL algorithms. A key issue in doing so is that performance highly depends on minute implementation details [6], which are often not the focus of academic research. One alternative is an open competition that provides a natural incentive for domain-specific optimization: winning. This format has become popular in recent years, and existing competitions can be roughly categorized into three classes: PvE, 1v1, and FFA.

PvE (Player vs Environment): these competitions evaluate agents against preset (usually randomized) environments as specific algorithmic benchmarks. For example, the NetHack challenge [7] concentrates on sparse reward and exploration while the MineRL competition [12] concentrates on sample efficiency. However, in PvE settings, agents are evaluated against given environments or fixed bots instead of other learning agents. This evaluation paradigm limits the ability to benchmark robustness and generalizability to new opponents.

1v1 or one team vs another: these competitions evaluate agents against other participants' policies in a two-agent or two-team mode, such as Google Research Football [9] and Lux-AI [4]. This requires agents to adapt to different kinds of opponents instead of specializing in a fixed environment.

FFA (Free-for-All): this setting places many independent agents or many teams in the same shared environment. Compared to the 2-team mode, FFA competitions can create a vast space of cyclic, non-transitive strategies and counter-strategies because of the combinatorial complexity and the dynamic relationships among agents. To our knowledge, the first Neural MMO challenge in 2021 [2] is the first competition that supports this FFA mode. In the competition, 16 participants' policies are evaluated together on the same map to benchmark their robustness and generalization. Our competition also follows this setting.

[1] https://agar.io/
[2] https://www.aicrowd.com/challenges/the-neural-mmo-challenge

To achieve accurate evaluation and get more participants involved, we set up two tracks: PvE and PvP. In the PvE track, submitted policies are confronted with different levels of preset AIs, which can be seen as a fixed environment. This PvE setting reduces the uncertainty of the evaluation process and helps participants identify potential improvements. In the PvP track, we adopt the FFA setting as it can better benchmark the policy's robustness and also provides a persistent incentive for participants to improve their policies. Our competition is the first RL competition with this dual-track system, which was well received by our participants.

3 COMPETITION ORCHESTRATION

3.1 Environment

3.1.1 Introduction to Neural MMO. Neural MMO is an open-source research platform that simulates populations of agents in procedurally generated virtual worlds. It is inspired by classic massively multiagent online role-playing games (MMORPGs or MMOs for short) as settings where lots of players using entirely different strategies interact in interesting ways. Unlike other game genres typically used in research, MMOs simulate persistent worlds that support rich player interactions and a wider variety of progression strategies. We refer the reader to the original publication [19] for full information on Neural MMO and its objectives. Our environment is adapted from version 1.5 of Neural MMO.

3.1.2 Competition Configuration . The competition configuration of Neural MMO places 128 agents in procedurally generated maps. Each map is 128x128 tiles. Scripted non-playable characters (NPCs) are spawned across the map. Agents must collect resources, *Food* and *Water* to survive and can attack each other and NPCs using three combat styles with strategic tradeoffs: *Melee*, *Mage*, and *Range*. The competition focuses on robustness to new maps and new opponents and the team design introduces cooperation and specialization to different roles on top of this.

3.1.3 Environment Wrapper. To make the environment work with different agents, i.e., rule-based and RL agents, we wrap Neural MMO with two major changes. First, agents are spawned uniformly at the edges of the map. We randomize both the map seed and the initial position of each team across episodes to ensure a fair evaluation. Second, the observations of 8 agents in a team are grouped and made available to a single policy when they make decisions. A

(a) Overall: shows overall team position and resource distribution.

(b) Close-up: shows important local details such as individual fights.

(c) Details: shows the current numeric properties of the chosen agent.

Figure 2: Web Viewer: a light visualization tool to show episode replays. (a) (b) (c) are three levels of view that can be altered during the playback. Users can rewind, pause, and change the playback speed on this web page. This allows participants to better understand the game and thus debug.

more strict setting in multi-agent cooperation might require that each agent compute its actions independently from teammates. We loosen that limit as in OpenAI Five [14] and AlphaStar[21] in favor of enabling higher overall policy quality.

3.2 Resources

For the participants' convenience, we have created a number of resources:

- **Starter Kit**[3]: a project containing all required segments to make a successful submission. With this guidance, new participants can make their first submission within 15 minutes.
- **Baseline**[4]: an RL baseline implementation in a single file based on *TorchBeast* [10]. This provides RL researchers with a fundamental baseline to start with.
- **Env Docs**[5]: Documents and tutorials to help participants to get familiar with Neural MMO.
- **Web Viewer**[6]: A light web replay viewer for our challenge, which allows participants with visual straightforward feedback for their policy development.

3.2.1 Web Viewer. The web viewer is a light visualization tool to show the replays of the episodes, allowing our participants to review their policy's performance. For RL researchers, an accessible viewer is crucial to analyze learned strategies and improve their policies. The web viewer UI contains three levels of view:

(1) An overall view, as shown in Fig.2a, demonstrating the whole team's trajectories and the resource distribution of the global map;
(2) A close-up view, as shown in Fig. 2b, which reveals local details such as individual fights, including the attack target and attack style of each agent;
(3) A view of numeric details, as shown in Fig. 2c, which shows the current numeric properties of the chosen agent, such as its skill levels, current health, and collected resources.

The right side of the interface shows the achievement scores of each team, allowing participants to interpret the varying abilities of each team to complete the four subtasks.

3.3 Competition Structure

The competition consists of two tracks: the PvE track and the PvP track. For clarity, PvE refers to one participant's policy vs. 15 built-in policies provided by the organizers. The PvE track serves as a fixed reference to help participants develop their policies. The PvE track contains 3 stages with different built-in policies and increasing difficulties. In the main PvP track, 16 participants' policies are thrown into shared environments. This can better test a policy's robustness and generalization to opponents not seen during training.

3.4 PvE (vs. fixed baselines)

The main PvP track enables players to test the robustness and generalization of their agents against a variety of foes. However, there is a high degree of uncertainty: the quality of opponents changes over time, and the only measure of policy performance is relative to that of all other submissions. We have thus set up an additional PvE track of 3 stages with 3 main purposes: (1) To help participants identify their agents' current performance against a fixed set of opponents of increasing quality; (2) To further incentivize participation by providing three reasonably achievable milestones; (3) To help participants understand the environment as guidance from easy to hard.

3.4.1 PvE Stage 1 (vs. Rule-Based Scripted Baselines). PvE stage 1 is a start-up stage arranged to help our participants familiarize themselves with the environment. We use three rule-based AIs named *Combat*, *Forage*, and *Random*. The *Combat* policy is hostile and will attack nearby agents. The *Forage* policy focuses only on collecting resources and will attempt to flee from combat. The *Random* policy moves randomly and is intended as a basic sanity check. Considering that all of these policies are open-source, this stage is intended to be relatively easy to beat.

[3] https://gitlab.aicrowd.com/neural-mmo/ijcai2022-nmmo-starter-kit
[4] https://gitlab.aicrowd.com/neural-mmo/ijcai2022-nmmo-baselines
[5] https://neuralmmo.github.io/build/html/rst/landing.html
[6] https://ijcai2022-viewer.nmmo.org/

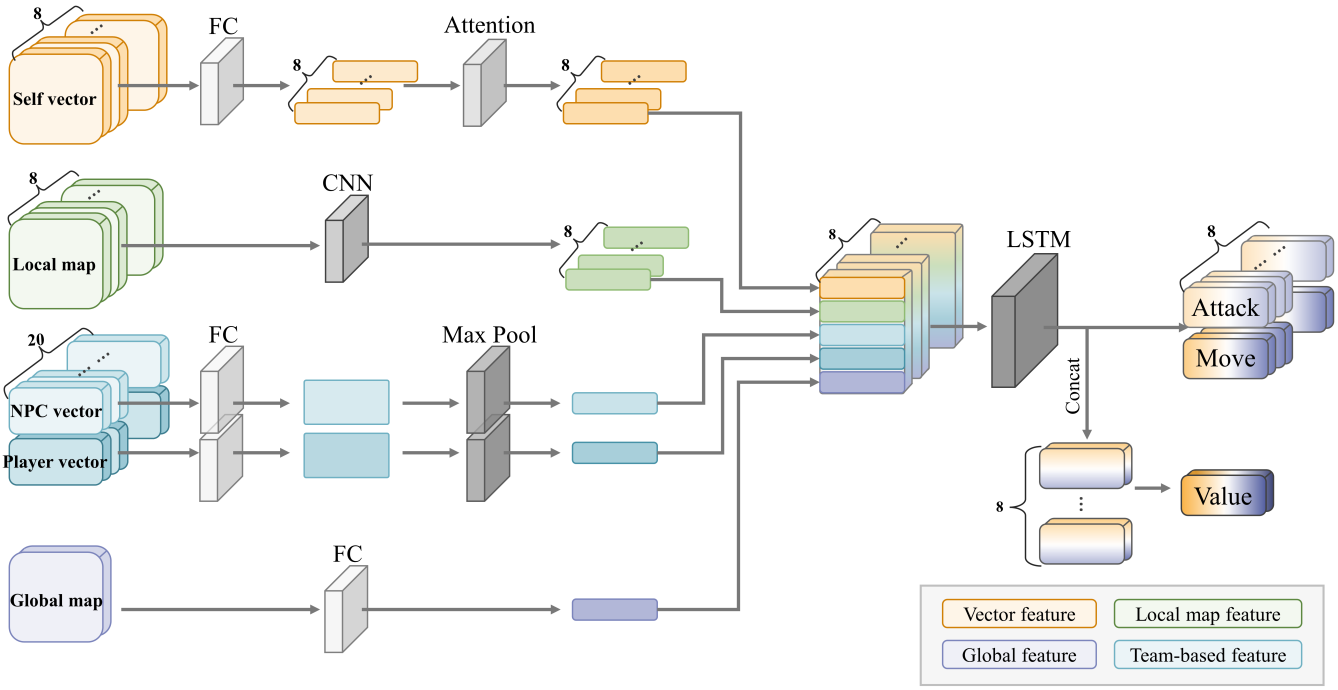

Figure 3: Model architecture for the PvE stage 3 baseline. It includes a core LSTM, a domain-specific observation encoder with sub-networks for flat, map, and set data, and a domain-specific decoder for fixed and variable-length actions.

Table 1: Tasks defined in this competition to measure the performance of policies. Teams earn 0, 4, 10, or 21 points per task, depending on the hardest difficulty of that task completed by at least one agent in the team. *Achievement* is defined as the sum of this score over all four tasks.

Task	Easy(4 points)	Medium(10 points)	Hard(21 points)
Travel the Lands	Explore 32 Meters	Explore 64 Meters	Explore 127 Meters
Forage for Resources	Attain Skill Lvl 20	Attain Skill Lvl 35	Attain Skill Lvl 50
Secure an Advantage	Acquire Lvl 1 Equipment	Acquire Lvl 10 Equipment	Acquire Lvl 20 Equipment
Eliminate the Competition	Defeat 1 Player	Defeat 3 Players	Defeat 6 Players

3.4.2 PvE Stage 2 (vs. RL Baselines). In PvE stage 2, we trained agents with PPO [16] using well-designed features extracted from raw observations. It is worth noting that we applied a *decentralized training* method, which means each agent in our team can only get its own observations and act individually. This provides a medium-level reference for participants, which is much harder than that in Stage 1 but only takes RL agents about one week to conquer.

3.4.3 PvE Stage 3 (vs. Team-Based RL Baselines). The PvE stage 3 AI has the highest performance over all the 4 sub-tasks and can prevail over the PvE stage 2 baseline. The key distinction here is that we adopt a team-based method to process the whole team's observation and compute actions jointly. To be more specific, we devised a *centralized training* strategy in which one network would concurrently process all 8 agents' observations and output the actions of eight agents. A team-based achievement is used as a reward. The advantage of this approach is that information is shared explicitly among teammates. Details of our modeling solution are as follows.

Feature Design. At the current scale, specialized feature extraction yields a large performance increase over processing raw observations. We divide, featurize, and process observations from all 8 agents on the team as follows:

- **Member features**: Each team member's self-information such as their IDs (to better correlate to the index of the following local map feature), the initial positions (to measure the derivation of the current position), etc.
- **Enemy & NPC features**: The observed entities' key features such as HP, level, and types are embedded here. This information helps agents learn how to behave in the presence of potential adversaries. We aggregate all 8 agents' observed entities together and use an additional vector of each entity to identify which agent is observing this entity to encourage the team-based policy to attack cooperatively;

- **Local map features**: Spatial information, such as the resource distribution on the observed local map, is embedded here to help agents learn to pathfind;
- **Global features:** Key global information such as time elapsed or the number of still-living teammates.

Policy Architecture and Training: The policy has three subnetworks: an observation encoder, the main long short-term memory (LSTM) network, and the action decoder. This architecture is shown in Fig.3. The input network contains fully-connected (FC) layers and max-pooling to process all scalar features and global information. Convolutional networks are used to process spatial information and attention modules are used to process position-invariant entity data. The main LSTM network processes the aggregate output of all of these encoders. The action output layers are normal FC layers. The policy is trained in a self-play setup against 15 teams controlled by the same policy. We employ valid action masks to accelerate exploration.

Reward Design: The competition scores teams based on their combat, foraging, and exploration. This mechanism is described in Section 4 below; the relevant aspect here is that we use this function directly in order to compute rewards.

3.5 PvP (vs. other participants)

Participants must pass the qualifying PvE Stage 1 in order to compete in PvP. Unlike in the PvE stages where opponents are fixed, the PvP opponent pool is dynamically sampled from the latest qualifying submissions from other participants. This includes reinforcement learned, scripted, and hybrid submissions.

We saw a large number of strategies emerge throughout the PvP stages, increasing our confidence in the platform as a proving ground for multi-agent reinforcement learning research, especially in testing the robustness of algorithms to new maps and opponents. We've noticed that some participants can rank high on PvE stage 1 or stage 2 but cannot maintain their advantage on the PvP stage, which indicates overfitting to the training domain.

4 EVALUATION SYSTEM

Each participant's policy will control a team of 8 agents and will be evaluated in a free-for-all against 15 other teams on 128x128 maps. After 1024 environment steps, the team with the highest *Achievement* wins.

4.1 Metrics

4.1.1 Multi-Task Metrics Definition. To evaluate the generalization of the policy, we design a suite of 4 tasks, as shown in Table 1. Each task has 3 difficulty levels: 4 points for easy, 10 points for normal, and 21 points for hard. Points were only awarded for the highest tier task completed in each category. The team with the most points at the end of a game (1024 steps) wins. This is a multi-objective task intended to be completed as a team: to achieve the maximum score for a task, only one agent on the team needs to complete it. This means it is reasonable for different agents on the team to employ different strategies. Such design encourages cooperation as a team and specialization of individual players.

4.1.2 TrueSkill in PvP. For the PvP track evaluation procedure, we randomly select 16 submissions from all qualified submissions to start a PvP match. In the final evaluation, each submission will participate in approximately 1000 matches. The mean achievement score is not a good evaluation metric due to the variability of opponents. For example, model A gets a high score against weaker opponents and a low score against stronger opponents, while model B gets an above-average score against all levels of opponents. In this case, the mean achievement scores of the two models may be close, but it is obvious that model B is more robust. To more accurately measure the relative strength of the models, we use TrueSkill [8] to compute scores for each submission.

4.1.3 Top 1 Ratio in PvE. In the PvE track, the participant's model will play 10 matches against our built-in AI. With 16 teams per match, the variance of the mean achievement score for 10 matches is high. To evaluate the robustness of the policy, we use Top1Ratio as the evaluation metric. The Top1Ratio is the ratio of games won (i.e. highest score among all teams) over 10 matches. A Top1Ratio close to 1.0 indicates that the model is significantly stronger than the 15 built-in teams.

4.2 Implementation

We developed the distributed evaluation system shown in Fig. 1 to quickly process submissions at scale. It can roll out hundreds of matches in parallel using k8s clusters and can return results within 10 minutes of submission.

We use the same distributed evaluation system for both the PvE and PvP tracks. The PvE track contains three levels; the main difference between them is the strength of the built-in AIs. Participants can enter the next level by reaching the specified Top1Ratio in the previous level. Reaching 25 points in PvE stage 1 qualifies a submission for the PvP track against other user submissions. Between PvE and PvP evaluation, we can accurately measure the strength of all models relative both to each other and to fixed baselines. The PvP evaluation is run once per week while PvE evaluation proceeds immediately upon submission. This ensures fast feedback to inform development at all times and more extensive feedback weekly.

5 SUMMARY AND ANALYSIS

5.1 Summary of the competition

The competition received over 40k views, 537 individual signups, 110 team signups, and 1679 submissions. This makes it one of the largest RL competitions to date, outpacing all of the MineRL competitions thus far and Nethack – despite having a significantly higher barrier to entry due to the complexity of the task, lack of a single-agent track, lack of offline data, and complex observation and action representation. Of these participants, 48 teams were

[8] https://www.aicrowd.com/challenges/ijcai-2022-the-neural-mmo-challenge
[9] https://www.aicrowd.com/challenges/neurips-2022-minerl-basalt-competition
[10] https://www.aicrowd.com/challenges/neurips-2021-minerl-diamond-competition
[11] https://www.aicrowd.com/challenges/neurips-2019-minerl-competition
[12] https://www.aicrowd.com/challenges/neurips-2021-the-nethack-challenge
[13] https://www.aicrowd.com/challenges/neurips-2020-procgen-competition
[14] https://www.aicrowd.com/challenges/flatland-3
[15] https://www.aicrowd.com/challenges/flatland
[16] https://www.aicrowd.com/challenges/unity-obstacle-tower-challenge
[17] https://www.aicrowd.com/challenges/neurips-2019-learn-to-move-walk-around
[18] https://www.aicrowd.com/challenges/neurips-2021-aws-deepracer-ai-driving-olympics-challenge/
[19] https://www.aicrowd.com/challenges/learn-to-race-autonomous-racing-virtual-challenge

Table 2: Comparison of major RL competitions on the AICrowd platform, the primary venue for these events. Our competition has the most unique submitters and the highest sign-up-to-submission conversion rate.

Competition	Views	Users	Submitted At Least Once	True Entry Rate
IJCAI 2022: Neural MMO Competition[8]	40.3k	540	**111**	**20.50%**
NeurIPS 2021: MineRL BASALT Competition[9]	38.8k	353	17	4.0%
NeurIPS 2021: MineRL Diamond Competition[10]	35.8k	511	65	12.7%
NeurIPS 2019: MineRL Competition[11]	69.3k	1124	41	3.6%
NeurIPS 2021 - The NetHack Challenge[12]	49.1k	584	46	7.9%
NeurIPS 2020 Procgen Competition[13]	52.8k	711	85	12.0%
Flatland 3[14]	19.2k	328	24	7.3%
Flatland[15]	72k	1090	65	6.0%
Unity Obstacle Tower Challenge[16]	74k	637	95	14.9%
NeurIPS 2019: Learn to Move - Walk Around[17]	37.9k	364	71	19.5%
NeurIPS 2021 AWS DeepRacer AI Driving Olympics Challenge[18]	20.4k	337	41	12.2%
Learn-to-Race: Autonomous Racing Virtual Challenge[19]	24.1k	476	53	11.1%

able to pass our first-round qualifier. 20 teams were able to win at least some games versus better policies that we trained for round 2, with 16 qualifying for round 3. We trained much stronger baselines for these rounds, but 7 teams were still able to win at least some games, and 6 were convincingly better than our best baseline. The best policies fully accomplished the task of the competition. Table 2 compares the metrics of major RL competitions and demonstrates the scope of our contest.

5.2 Analysis of the Competition Design

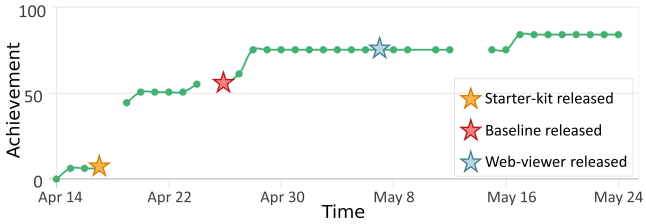

Figure 4: Maximum achievement in PvE stage 1 through time, measured over all participants. The release of the baseline corresponds with a large jump in submission quality.

Fig.4 shows the increases of max achievement over time. We can find that the three sudden rises are due to the starter kit release, the official baseline release, and the web viewer release: these tools were either useful or at least motivational to participants.

Fig.5 shows performance against different stages of the PvE track. The baseline quality increases across rounds, so submission performance declines as expected from stage to stage. Interestingly, the final PvE stage results are similar to those of the final PvP track. This suggests that this track was effective in allowing participants to quickly evaluate their submissions. This is useful because the PvP stage is more computationally expensive, so we can only run it once per week.

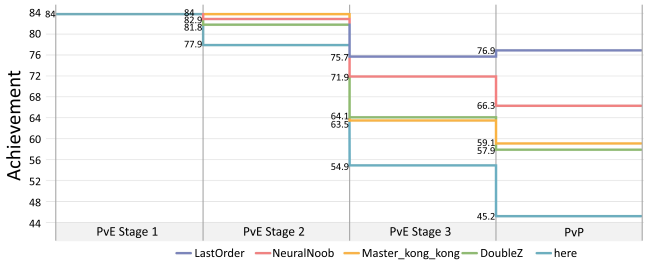

Figure 5: Top five participant scores in each round. The increasing difficulty of later PvE rounds corresponds with a decline in achievement score. Performance in stage 3 is comparable to performance in the last PvP stage.

5.3 Analysis of 1600+ Policies

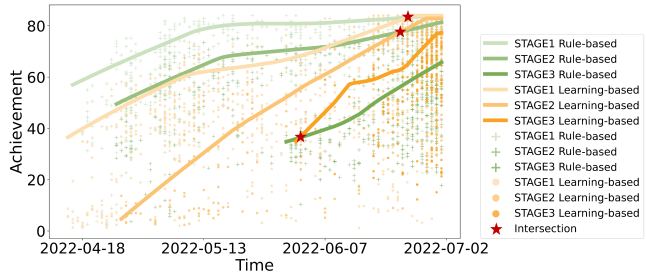

Figure 6: The effectiveness of Rule-Based and Learning-Based methods across three PvE stages. The six lines represent the peak performance of the approach at each stage. The red star marks the point at which the highest performance of the two approaches overlaps.

We gathered over 1600 submissions and categorized them as rule-based methods (behavior tree, planning-based methods, heuristic methods, etc.) or learning-based methods (reinforcement learning)

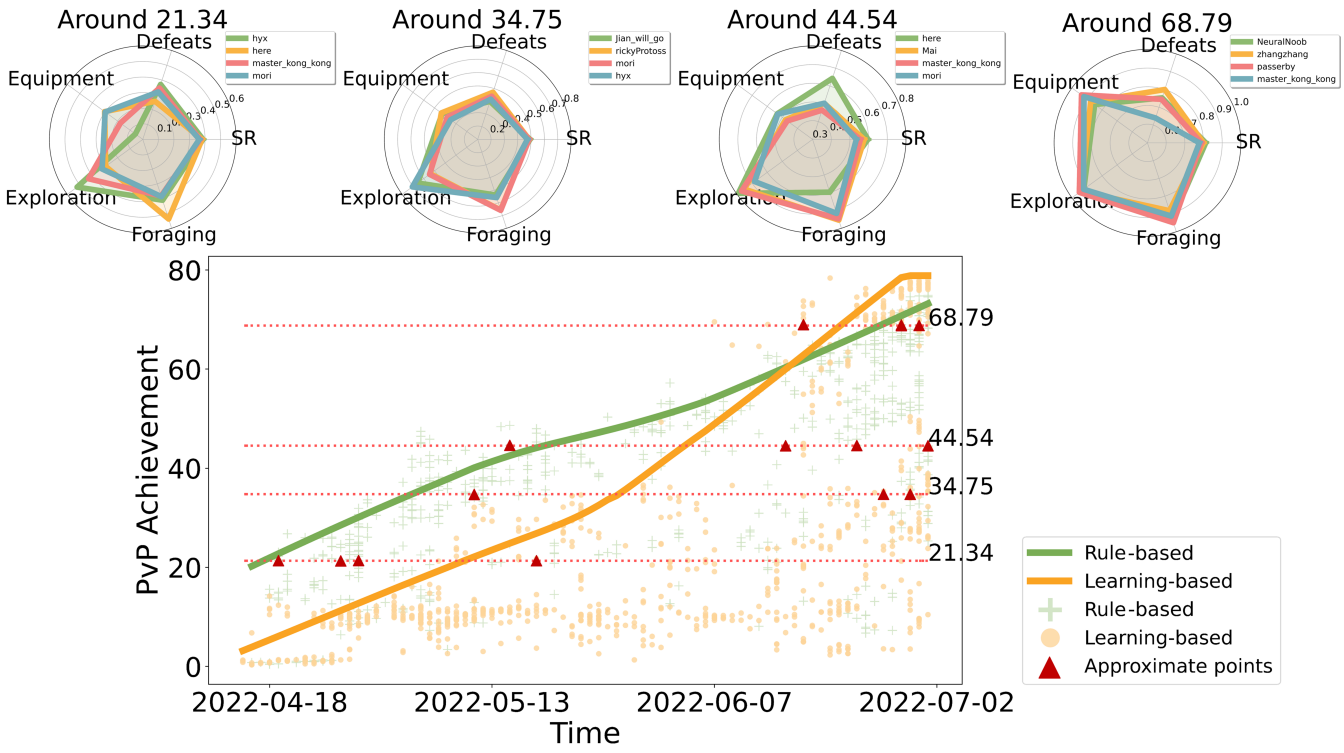

Figure 7: We compared the performance of players with the same achievement on each of the four subtasks. Even if the final achievements are identical, participants will employ different methods to accomplish the job, demonstrating that Neural MMO can accommodate a variety of tactics.

based on the algorithms employed by the participants. **We additionally release presentations from the top 5 teams about their approach (camera ready for anonymity).**

Fig.6 illustrates the best achievements over time for both classes of methodologies in the three stages of PvE and PvP track. We make several observations. (1) Rule-based or learning-based methods both achieve satisfactory performance. (2) The performance curves of rule-based and learning methods cross earlier in later stages. This suggests that rule-based methods are quick to get working but do not scale as well against complex opponents. (3) The green lines of rule-based methods in Stage 1 and Stage 2 almost converge and still climb in Stage 3. The orange lines of learning-based methods are all climbing. Thus, there is still room for further research even on this version of Neural MMO, without even considering some of the more recent additions to the environment.

5.3.1 Robustness and Generalization. As seen in the Fig.5, the performance of the participants' models varies at different stages. As an example, plotted the exploration pattern of the winning policy *passersby*, against a weaker opponent (PvE stage 2) and against a stronger opponent (PvE stage 3). This player's model explores significantly more against inferior opponents than against stronger ones, which is shown in Fig.10. This further demonstrates that this environment may facilitate the study of model robustness and generalization by introducing diverse adversaries.

5.3.2 Diversity of Policies. We find that policies that achieve the same score may employ different strategies, as indicated by differing performance on the four subtasks and the overall trajectories of different agents. The models of different players can have varied strengths and weaknesses on the sub-tasks as shown in Fig.7. Using the four players with final achievements close to 68.79 as an example, *zhangzhang*'s model achievement on *Defeat* is high, but their *Equipment* is inadequate, indicating that their agents' primary tactic is to kill other players' agents. The superior performance of *kongkong*'s agents in both *Foraging* and *Equipment* suggests that their strategy is more adept at utilizing map resources.

For the trajectories shown in Fig.9, we choose the paths of the top five ranking submissions and observe that various teams have distinct navigation preferences. The team *here*, for instance, will explore in a straight line to maximize their exploration score, whereas the team *DoubleZ* will go deeper into the heart of the map from the beginning because there are higher-level NPCs there, allowing them to quickly upgrade their equipment. *Master_kong_kong*'s team will do the most comprehensive exploration, allowing them to become familiar with the entire area more quickly. Similarly, we count the frequency of visits to each tile by the agents under different strategies, and we can find that there will be differences among models as shown in Fig.8.

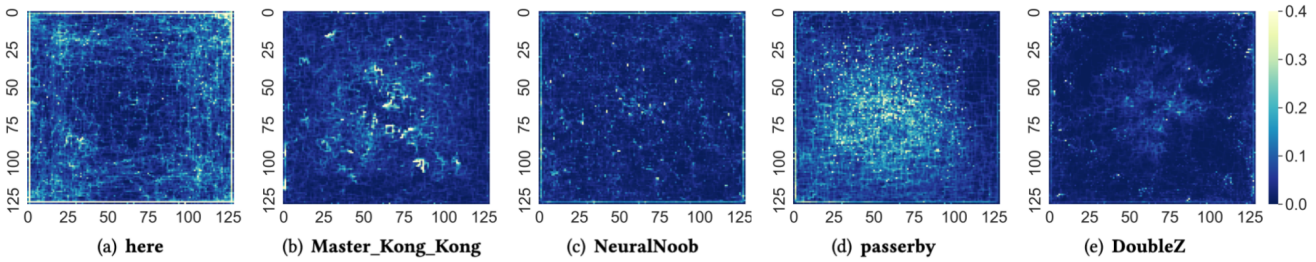

Figure 8: Visitation frequency of various policies, computed by summing per-tile exploration counts over over 50 episodes. Different policies demonstrate distinct exploration preferences. For example, the *here* policy primarily explores around the edges of the map while *passerby* spends more time in the center of the map.

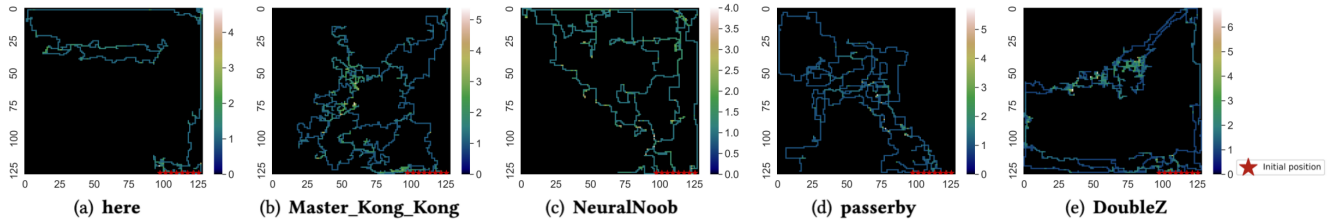

Figure 9: Movement path of five teams. Different policies employ different pathing strategies, with some choosing to disperse at the start and converge at the center while others explore as a team.

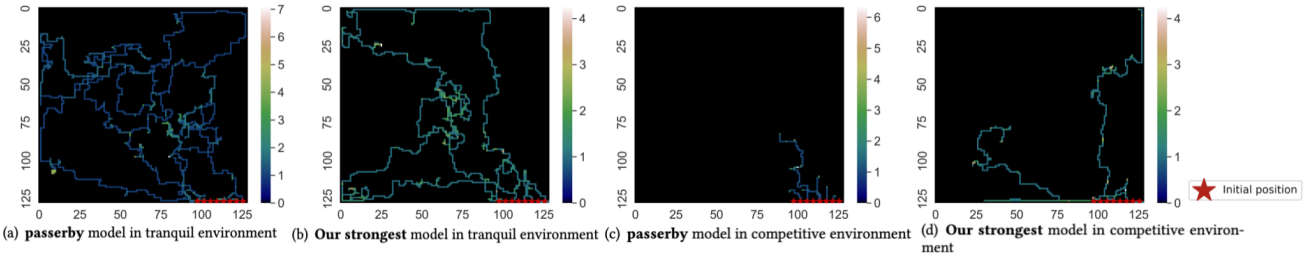

Figure 10: The figure depicts the movement trajectory of the participant model against weak (PvE stage 1) and strong (PvE stage 3) opponents. We have compared our strongest model in the same context. Passerby employs different pathing strategies against different opponents, demonstrating that our competition may be utilized to evaluate policy robustness.

6 CONCLUSION

To benchmark the robustness and generalization of MARL algorithms, we hosted a multi-agent artificial intelligence challenge and received 1600+ policy submissions. The top five submissions all surpassed the best existing baselines while employing strategies ranging from rule-based to full RL. However, the performance curve till the end of the competition indicates that policies have not yet reached the performance upper bound in the environment and that there is still considerable potential for RL algorithms in further research.

From an algorithmic perspective, the results of this competition and the analysis of the top-ranking solutions demonstrate that the conceptually simple methods effective in large-scale industry research can also work on complex but academic-scale tasks. We suggest that a gap in tooling and infrastructure, rather than purely algorithms, is the main short-term bottleneck preventing reinforcement learning from working on complex, multi-agent environments. We argue that the simplest way to realize this result in other environments is to run competitions and open-source the tools built by organizers and participants. By aggregating these implementations across multiple domains, we may begin to see the commonalities and build more general-purpose tools. We hope that our work will inspire others to adopt the competition model of research and open-source their tooling as we have.

REFERENCES

[1] Anonymous. 2023. GoBigger: A Scalable Platform for Cooperative-Competitive Multi-Agent Interactive Simulation. In *Submitted to The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=NnOZT_CR26Z under review.

[2] Karl Cobbe, Christopher Hesse, Jacob Hilton, and John Schulman. 2020. Leveraging Procedural Generation to Benchmark Reinforcement Learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event (Proceedings of Machine Learning Research, Vol. 119)*. PMLR, 2048–2056. http://proceedings.mlr.press/v119/cobbe20a.html

[3] Karl Cobbe, Oleg Klimov, Chris Hesse, Taehoon Kim, and John Schulman. 2019. Quantifying Generalization in Reinforcement Learning. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 1282–1289.

[4] Bovard Doerschuk-Tiberi and Stone Tao. 2021. *Lux AI Challenge Season 1.* https://github.com/Lux-AI-Challenge/Lux-Design-2021

[5] Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. 2016. Benchmarking Deep Reinforcement Learning for Continuous Control. In *Proceedings of The 33rd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 48)*, Maria Florina Balcan and Kilian Q. Weinberger (Eds.). PMLR, New York, New York, USA, 1329–1338.

[6] Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. 2020. Implementation Matters in Deep RL: A Case Study on PPO and TRPO. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

[7] Eric Hambro, Sharada Mohanty, Dmitrii Babaev, Minwoo Byeon, Dipam Chakraborty, Edward Grefenstette, Minqi Jiang, Jo Daejin, Anssi Kanervisto, Jongmin Kim, et al. 2022. Insights from the NeurIPS 2021 NetHack challenge. In *NeurIPS 2021 Competitions and Demonstrations Track*. PMLR, 41–52.

[8] Ralf Herbrich, Tom Minka, and Thore Graepel. 2006. TrueSkill™: a Bayesian skill rating system. *Advances in neural information processing systems* 19 (2006).

[9] Karol Kurach, Anton Raichuk, Piotr Stanczyk, Michal Zajac, Olivier Bachem, Lasse Espeholt, Carlos Riquelme, Damien Vincent, Marcin Michalski, Olivier Bousquet, and Sylvain Gelly. 2020. Google Research Football: A Novel Reinforcement Learning Environment. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, 4501–4510.

[10] Heinrich Küttler, Nantas Nardelli, Thibaut Lavril, Marco Selvatici, Viswanath Sivakumar, Tim Rocktäschel, and Edward Grefenstette. 2019. TorchBeast: A PyTorch Platform for Distributed RL. *CoRR* abs/1910.03552 (2019). arXiv:1910.03552 http://arxiv.org/abs/1910.03552

[11] Heinrich Küttler, Nantas Nardelli, Alexander Miller, Roberta Raileanu, Marco Selvatici, Edward Grefenstette, and Tim Rocktäschel. 2020. The NetHack Learning Environment. *Advances in Neural Information Processing Systems* 33 (2020), 7671–7684.

[12] Stephanie Milani, Nicholay Topin, Brandon Houghton, William H Guss, Sharada P Mohanty, Keisuke Nakata, Oriol Vinyals, and Noboru Sean Kuno. 2020. Retrospective analysis of the 2019 MineRL competition on sample efficient reinforcement learning. In *NeurIPS 2019 Competition and Demonstration Track*. PMLR, 203–214.

[13] Sharada P. Mohanty, Jyotish Poonganam, Adrien Gaidon, Andrey Kolobov, Blake Wulfe, Dipam Chakraborty, Grazvydas Semetulskis, João Schapke, Jonas Kubilius, Jurgis Pasukonis, Linas Klimas, Matthew J. Hausknecht, Patrick MacAlpine, Quang Nhat Tran, Thomas Tumiel, Xiaocheng Tang, Xinwei Chen, Christopher Hesse, Jacob Hilton, William Hebgen Guss, Sahika Genc, John Schulman, and Karl Cobbe. 2020. Measuring Sample Efficiency and Generalization in Reinforcement Learning Benchmarks: NeurIPS 2020 Procgen Benchmark. In *NeurIPS 2020 Competition and Demonstration Track, 6-12 December 2020, Virtual Event / Vancouver, BC, Canada (Proceedings of Machine Learning Research, Vol. 133)*, Hugo Jair Escalante and Katja Hofmann (Eds.). PMLR, 361–395.

[14] OpenAI, :, Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, Rafal Józefowicz, Scott Gray, Catherine Olsson, Jakub Pachocki, Michael Petrov, Henrique P. d. O. Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, Jonas Schneider, Szymon Sidor, Ilya Sutskever, Jie Tang, Filip Wolski, and Susan Zhang. 2019. Dota 2 with Large Scale Deep Reinforcement Learning. https://doi.org/10.48550/ARXIV.1912.06680

[15] Mikayel Samvelyan, Tabish Rashid, Christian Schroeder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philiph H. S. Torr, Jakob Foerster, and Shimon Whiteson. 2019. The StarCraft Multi-Agent Challenge. *CoRR* abs/1902.04043 (2019).

[16] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[17] Rohin Shah, Steven H. Wang, Cody Wild, Stephanie Milani, Anssi Kanervisto, Vinicius G. Goecks, Nicholas R. Waytowich, David Watkins-Valls, Bharat Prakash, Edmund Mills, Divyansh Garg, Alexander Fries, Alexandra Souly, Jun Shern Chan, Daniel del Castillo, and Tom Lieberum. 2021. Retrospective on the 2021 MineRL BASALT Competition on Learning from Human Feedback. In *NeurIPS 2021 Competitions and Demonstrations Track, 6-14 December 2021, Online (Proceedings of Machine Learning Research, Vol. 176)*, Douwe Kiela, Marco Ciccone, and Barbara Caputo (Eds.). PMLR, 259–272. https://proceedings.mlr.press/v176/shah22a.html

[18] Joseph Suarez, Yilun Du, Phillip Isola, and Igor Mordatch. 2019. Neural MMO: A massively multiagent game environment for training and evaluating intelligent agents. *arXiv preprint arXiv:1903.00784* (2019).

[19] Joseph Suarez, Yilun Du, Clare Zhu, Igor Mordatch, and Phillip Isola. 2021. The Neural MMO Platform for Massively Multiagent Research. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, Joaquin Vanschoren and Sai-Kit Yeung (Eds.).

[20] Zhenggang Tang, Chao Yu, Boyuan Chen, Huazhe Xu, Xiaolong Wang, Fei Fang, Simon Shaolei Du, Yu Wang, and Yi Wu. 2021. Discovering Diverse Multi-Agent Strategic Behavior via Reward Randomization. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. https://openreview.net/forum?id=lvRTC669EY_

[21] 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 Sasha Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom Le Paine, Çaglar Gülçehre, Ziyu Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy P. Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* 575, 7782 (2019), 350–354. https://doi.org/10.1038/s41586-019-1724-z