

# The NeurIPS 2022 Neural MMO Challenge: A Massively Multiagent Competition with Specialization and Trade

**Enhong Liu\***

*Parametrix.AI*

**Joseph Suarez\***

*Massachusetts Institute of Technology*

MUDOULIU@CHAOCANSHU.AI

JSUAREZ@MIT.EDU

**Chenhui You**

**Bo Wu**

**Bingcheng Chen**

**Jun Hu**

**Jiaxin Chen<sup>†</sup>**

**Xiaolong Zhu**

*Parametrix.AI*

KIRSTYYOU@CHAOCANSHU.AI

BOWU@CHAOCANSHU.AI

ARBINGCHEN@CHAOCANSHU.AI

JUNHU@CHAOCANSHU.AI

JIAXINCHEN@CHAOCANSHU.AI

XIAOLONGZHU@CHAOCANSHU.AI

**Clare Zhu**

*Stanford University*

**Julian Togelius**

*New York University*

**Sharada Mohanty**

*AICrowd*

CLAREZHU@CS.STANFORD.EDU

JULIAN.TOGELIUS@NYU.EDU

MOHANTY@AICROWD.COM

**Weijun Hong**

*NetEase Games AI Lab*

**Rui Du**

*Bilibili*

**Yibing Zhang**

*Chengdu Goldwin Electronics Technology Co., Ltd*

HONGWEIJUN@CORP.NETEASE.COM

DURUI@BILIBILI.COM

554011619@QQ.COM

**Qinwen Wang**

**Xinhang Li**

**Zheng Yuan**

**Xiang Li**

**Yuejia Huang**

*Advanced Network Technology Laboratory, Beijing University of Posts and Telecommunications*

WANGQINWEN@BUPT.EDU.CN

LIXINHANG@BUPT.EDU.CN

YUANZHENG@BUPT.EDU.CN

LIXIANGLGC@BUPT.EDU.CN

HUANGYUEJIA@BUPT.EDU.CN

**Kun Zhang**

*Independent Researcher*

MORI42@QQ.COM

**Hanhui Yang**

*The Hong Kong Chinese University*

HHYANG@MATH.CUHK.EDU.HK

**Shiqi Tang**

*The Hong Kong City University*

SQTANG2-C@MY.CITYU.EDU.HK

**Phillip Isola**

*Massachusetts Institute of Technology*

PHILLIPI@MIT.EDU

**Editors:** Marco Ciccone, Gustavo Stolovitzky, Jacob Albrecht

## Abstract

In this paper, we present the results of the NeurIPS-2022 Neural MMO Challenge, which attracted 500 participants and received over 1,600 submissions. Like the previous IJCAI-2022 Neural MMO Challenge, it involved agents from 16 populations surviving in procedurally generated worlds by collecting resources and defeating opponents. This year’s competition runs on the latest v1.6 Neural MMO, which introduces new equipment, combat, trading, and a better scoring system. These elements combine to pose additional robustness and generalization challenges not present in previous competitions. This paper summarizes the design and results of the challenge, explores the potential of this environment as a benchmark for learning methods, and presents some practical reinforcement learning training approaches for complex tasks with sparse rewards. Additionally, we have open-sourced<sup>1</sup> our baselines, including environment wrappers, benchmarks, and visualization tools for future research.

## 1. Introduction

The ecosystems and populations of Earth feature mixed cooperation and competition. Inspired by this, the Neural MMO environment was designed to accommodate a large number of agent populations with limited resources, necessitating socially-aware planning for survival. We propose the Neural MMO Challenge as a means to explore the emergence of many-agent intelligence and promote research on cognitively sophisticated reinforcement learning environments.

We have held two Neural MMO challenges to date. The previous challenge was designed to study multi-task RL and provide a benchmark for robustness and generalization in multi-agent systems [Suarez and Isola \(2022\)](#). Although it attracted many players and yielded valuable policies, the exact task definition and unbalanced attack mechanisms limited the emergence of interesting behaviors. Therefore, we require more complex environments and a sparse scoring system to make the environment more engaging.

In the NeurIPS-2022 Neural MMO Challenge, we updated the game mechanics and scoring method of the Neural MMO competition. The specific adjustments are as follows:

1. In order to encourage the emergence of more interesting strategies, we do not set an explicit task for agents. Instead, we demand that agents complete a final, sparse goal: survival for as long as possible.
2. We use the updated v1.6 release of Neural MMO, which introduces combat, professions, equipment, a battle-royale inspired fog mechanic, and a trading mechanism.
3. We have made dozens of fixes to the environment based on the previous competition to ensure that it is fair and strategically interesting.

Other multiagent reinforcement learning (MARL) environments include StarCraft 2 [Oriol Vinyals \(2019\)](#) (open-source) and Dota2 [Christopher Berner \(2019\)](#) (not publicly available), as well as a number of smaller and more niche projects. The main issue is that

---

<sup>\*</sup> Equal Contribution

<sup>†</sup> Corresponding Author

1. <https://github.com/NeuralMMO/NeurIPS2022NMMO-Submission-Pool>

no other environment couples large population capacity with high per-agent complexity. The competition build of Neural MMO features 128 agents spread across 16 teams where each individual agent must make complex decisions about foraging, combat, specialization, cooperation, and trade. The mechanics introduced by the latest version of the environment support more diversified and complicated policies compared to previous versions. Our contributions based on the NeurIPS-2022 Neural MMO competition are as follows:

1. We provide a specific, controlled setting of Neural MMO suitable for benchmarking a wide range of learning methods.
2. We establish an imitation learning (IL) track and associated dataset of agent trajectories. Most competitions in this area are limited to reinforcement learning and rule-based (scripted) approaches. This competition shows that IL can also achieve satisfactory results.
3. We summarize our experience in designing the competition tracks and resources for participants. We have previously held a similar competition at IJCAI, but much of our previous experience needs updating. We believe the structures we present in this paper will be useful for other competitions.

## 2. Related Work

The real world is massively multiagent, and research on MARL is essential for solving large-scale and cognitively realistic problems. To aid in the study of multi-agent behavior, Jiang et al. proposed a basic MARL environment library with several simple grid-based environments [Jiang and Amato \(2021\)](#). The particle world project [Ryan Lowe \(2017\)](#) allows users to customize MARL environments using their APIs. However, the rules and action space in these environments are limited.

To address this, researchers have created more realistic environments such as Fever Basketball [Jia and Chunxu Ren \(2020\)](#), Google Research Football [Karol Kurach \(2019\)](#), Pommerman [Cinjon Resnick \(2018\)](#), and StarCraft II. The latter allows observation of the control ability of a MARL algorithm for multi-agent units and has led to the development of well-known algorithms like Qmix [Rashid et al. \(2018\)](#) and COMA [Foerster et al. \(2017\)](#). While these environments enable RL algorithms to control multiple agents with different action spaces, the number of agents is still far less than the number in natural populations.

Neural MMO, an open-source complex MARL environment, is unique in that it can accommodate a vast number of populations while providing limited resources for survival, forcing populations to learn to cooperate and compete. With the introduction of a battle-royale inspired fog mechanic, equipment mechanism, and trading mechanism, researchers can simulate social behaviors, evaluate the robustness and generalization of RL algorithms, and explore diverse multi-agent research topics.

## 3. Competition Overview

Neural MMO is an environment where agents must fight, forage, barter, and more in order to survive against multiple teams of intelligent adversaries. As shown in Fig. 1, agents

must make complex decisions at each game tick. The large decision space and long-term dependency of behaviors bring diversity to agents' strategies.

The environment has the following characteristics:

- 1. Massively multiagent:** The Neural MMO environment can accommodate 1000+ agents, a capability that few environments possess.
- 2. Interesting mechanics:** The battle-royale inspired fog mechanic, combat system, equipment system, and trading mechanism are introduced in Neural MMO. These fresh game mechanics make the environment appealing to researchers and worth exploring.
- 3. Flexible settings:** The generation ratio of various terrains in the environment can be adjusted at any time, which enables better testing of the stability and robustness of RL algorithms. Additionally, game mechanics can be controlled flexibly to accommodate different research topics.
- 4. Sparse scoring method:** The environment uses a sparse scoring rule for ranking. In other words, agents must complete long-term tasks with unclear rewards, which encourages the emergence of interesting behaviors and diverse strategies.

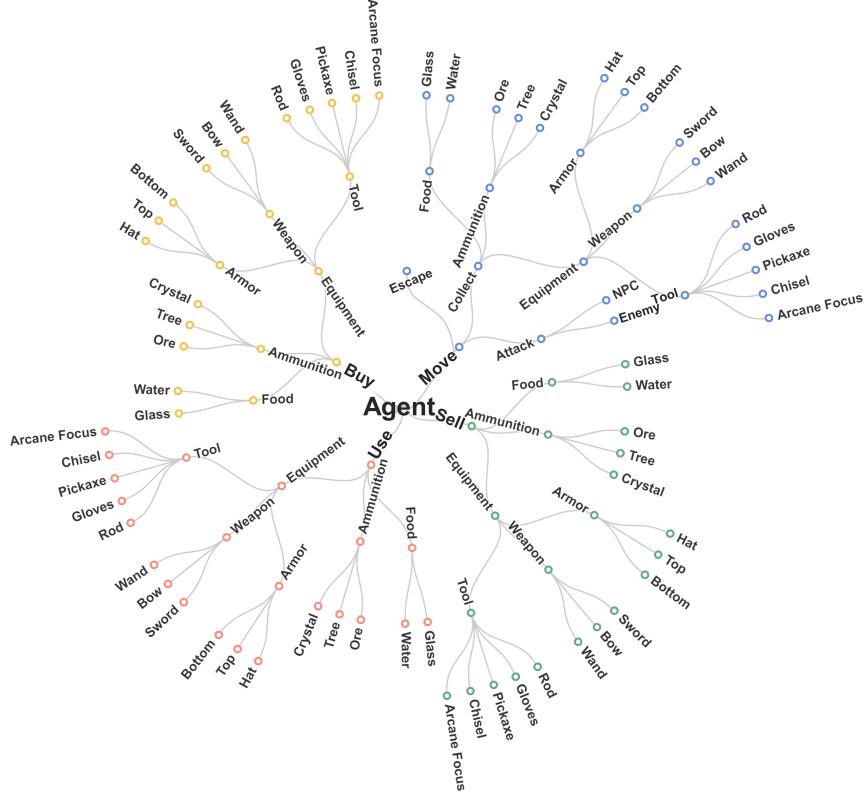


Figure 1: The decisions that agents face at every game tick of the environment

5. **Support for various algorithms:** The environment supports arbitrary agent controllers, not just RL algorithms.

At each step of the game, a map with dimensions of  $128 \times 128$  is randomly generated, with 128 agents from 16 teams spawning around the edges of the map. Each agent begins with initialized health, food, and water levels set to 100. Teams must navigate the environment, gathering resources, equipping weapons, hunt NPCs, and engage in combat. The game includes the following features:

1. **Terrain:** There are 16 different types of terrain on the map, each offering different benefits to agents who pass through them. For example, passing through foliage replenishes food, while passing through ore generates melee ammo.
2. **NPCs:** Three types of NPCs are evenly distributed from the edge to the center of the map. NPCs play a critical role in the game, as agents primarily obtain gold and items by defeating them. Generally, the higher-level NPCs an agent defeats, the better the items they can obtain.
3. **Blue Circle:** From the 240th game tick, a battle-royale inspired fog or "blue circle" begins to shrink. For every 16 ticks, the circle shrinks one tile. Agents within the circle take damage proportional to their distance from the safe zone.
4. **Skills:** Agents can train eight skills. A higher skill level allows agents to use better equipment, which in turn allows them to inflict more damage on enemies, take less damage, or restore more HP/Food/Water.
5. **Combat:** Agents have access to melee, range, and mage attacks, which follow a rock-paper-scissors dominance relationship: melee beats range, range beats mage, and mage beats melee. Dominance is calculated using the attacker's chosen attack skill and the defender's main combat skill. The dominant style causes 1.5 times damage.

### 3.1. Observation Space

To make decisions, each agent can observe various aspects of the game, including its own equipment information, the global information store, some enemy information, all teammate information, and all terrain information within its field of view.

### 3.2. Action Space

Table 1 shows the available actions that the agent can take. These include deciding where to move, who to attack, how to attack, what to use, what to buy, and what interaction information to expose. Note that the agent cannot move to a tile with other agents or certain special terrains, which are outlined in Table 2. Using an item will have different effects based on its properties, which are detailed in Table 3. When selling equipment, the agent can select any item in its backpack, specify the selling price, and place the item on the global market. If the sale is successful, the agent will receive the proceeds from the sale.

Table 1: Introduction of the action space of an agent in the Neural MMO.

Action	Option	Dimension
Move	Up,Down,Left,Right	3
Attack Target	Enemy in sight	225
Attack Style	Melee,Ranged,Mage	3
Use	Item in package	12
Sell	Item in package	12
Buy	Item in global market	170
Communicate	Discrete numerical values	128

Table 2: Different terrains and their corresponding property.

Element	Usage	Passable
Lava	Lethal upon contact	False
Water	Harvest Water	False
Grass	Can walk on	True
Scrub	Forest degrades to Scrub after being harvested	True
Forest	Harvest Forest for producing food	True
Stone	Impassible barriers	False
Slag	Ore degrades to Slag after being harvested	True
Ore	Harvest ore to produce melee ammunition	True
Stump	Tree degrades to Stump after being harvested	True
Tree	Harvest Tree to produce range ammunition	True
Fragment	Crystal degrades to Fragment after being harvested	True
Crystal	Harvest crystal to produce mage ammunition	True
Weeds	Herb degrades to weeds after being harvested	True
Herb	Harvest Herb to produce Ration	True
Ocean	Fish degrades to ocean after being harvested	False
Fish	Harvest Fish to produce Poultice	False

### 3.3. Technical Details

In PvE stage 2, participants face RL-based agents that we trained. Figure 2 shows the structure of the model we designed. We adopt a team-based modeling approach, which integrates the information of the entire team to obtain a joint policy. For table-type features such as self-features, enemy features, env features, and market features, we use an MLP for encoding. We apply a CNN to process the image-type features, i.e., map features. To improve data efficiency, we introduce an attention component into the model. For instance, in the buy action, buying items is related to the agent and the items existing in the market. Therefore, we perform attention to self-embedding and market embedding and put the attention embedding into the MLP to obtain the final action.

We take a curriculum learning approach to train the model and obtain three different styles of built-in AI, which we refer to as "reckless", "ruthless", and "coward". In the first

Table 3: Different items and their corresponding property.

Category	Usage	Items	Level
Ammunition	Increase damage	Shaving/Scrap/Shard	1-10
Weapon	Increase damage	Sword/Bow/Wand	1-10
Armor	Increase defense	Hat/Top/Bottom	1-10
Consumable	Restore Food/Water/HP	Ration/Poultice	1-10
Tool	Increase the level of products after gathering resources	Rod/Gloves/Pickaxe Chisel/Arcane Focus	1-10
Gold	Buy items from the global market		N/A

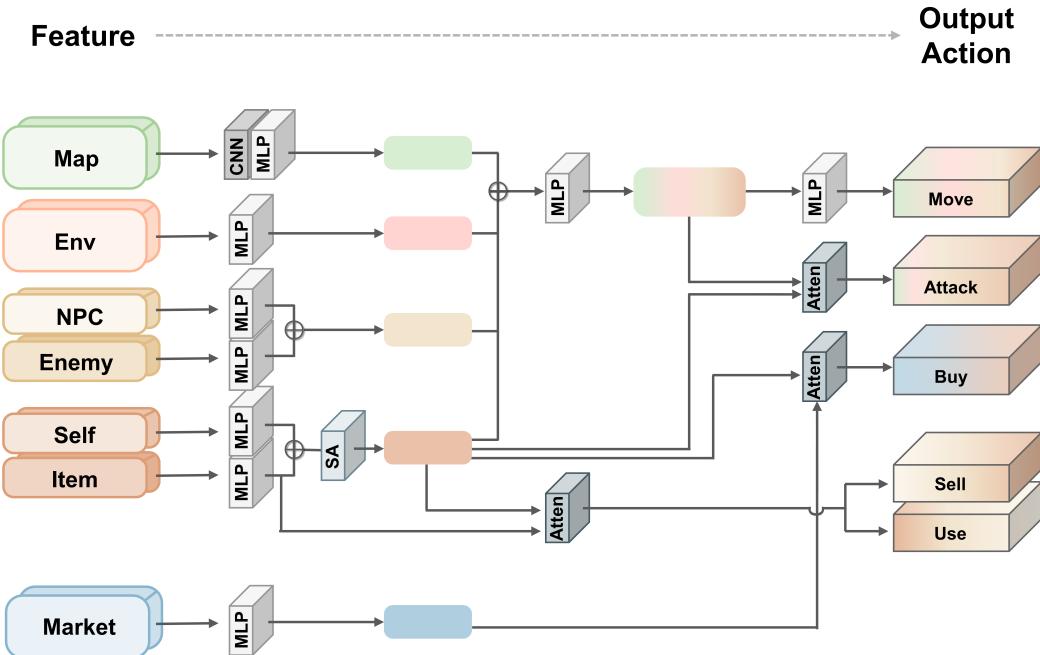


Figure 2: Model architecture for the PvE stage 2 built-in AI. Subnetworks include MLP, CNN, self-attention (SA), and attention. Baseline included with code release.

training stage, we set general rewards for agents to help them learn basic skills such as attacking, gathering, and escaping the blue circle. This produces the "reckless" baseline. These agents do not cooperate well, so we continue training with an additional reward to produce the "ruthless" baseline. In addition, we also set some penalties based on the checkpoint of the first training stage and obtain the "coward" baseline. Compared to the previous two built-in AIs, the "coward" watches and reacts to situations and never strikes more capable rivals first.

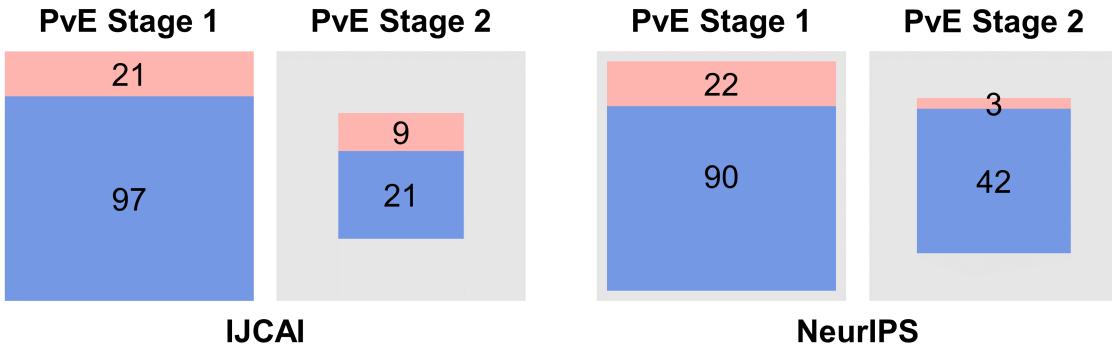


Figure 3: Performance of participants in the IJCAI and NeurIPS Neural MMO challenges. In the IJCAI competitions, 21 participants achieved 1.0 top 1 ratio in PvE stage 1 and 9 participants achieved 1.0 top 1 ratio in PvE stage 2. In the NeurIPS Neural MMO challenge, 22 participants achieved 1.0 top 1 ratio in PvE stage 1 and only 3 participants achieved 1.0 top 1 ration in PvE stage 2

## 4. Competition Summary

### 4.0.1. IMPROVEMENT

The NeurIPS Neural MMO competition attracted over 20,000 views, 475 registrations, 134 team registrations, and 1651 submissions. Compared to the previous competition, this year’s competition was more challenging due to the introduction of more complex game mechanics. As depicted in Fig. 3, only a few participants achieved ideal results. In the PvE stage 1 of the IJCAI and NeurIPS challenges, many players were able to successfully overcome the built-in AIs. However, in the PvE stage 2, only 9 participants outperformed the built-in AIs in the IJCAI challenge, and in this challenge, only the top 3 participants were able to outmatch the built-in AIs.

It is worth noting that the change in the scoring mechanism in this competition resulted in a better representation of the strengths and weaknesses of participants’ policies. As shown in Fig. 4, in the previous competition, the top 5 participants’ scores were very close, and it was difficult to distinguish their strategies based on the achievement indicators. In contrast, in this competition, although the top 5 participants’ results were similar, we were able to distinguish obvious differences among their policies based on evaluation indicators.

To summarize, the game mechanics of this competition were more interesting than those of the previous competition, and the flexible scoring system showcased the potential of the Neural MMO environment for MARL research.

### 4.0.2. TOOLS AND SERVICE

Figure 5 shows the submission performance curve of the NeurIPS-2022 Neural MMO PvE stage 1. Unlike the previous competition, where participants achieved ideal results without a baseline, this time, the number of submissions was infrequent, and none of them achieved good results before we provided the baseline. This indicates that as competition difficulty

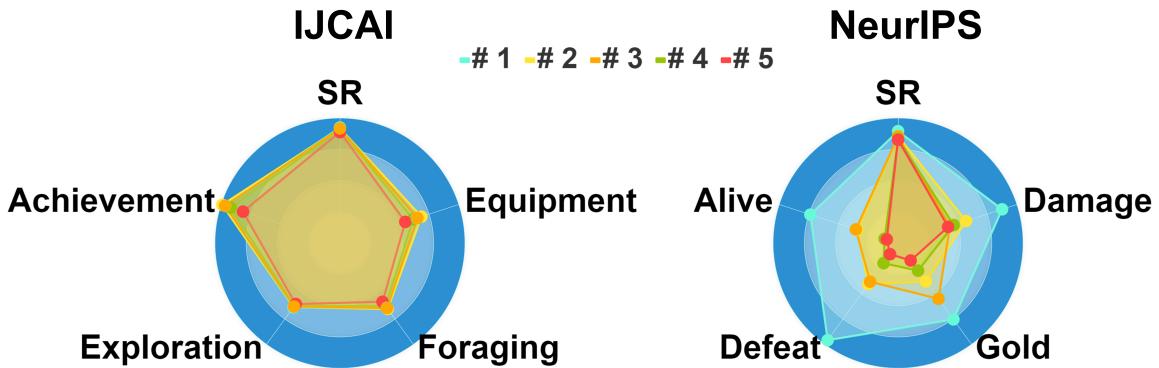


Figure 4: An overview of top 5 submissions’ capabilities in the past two competitions.

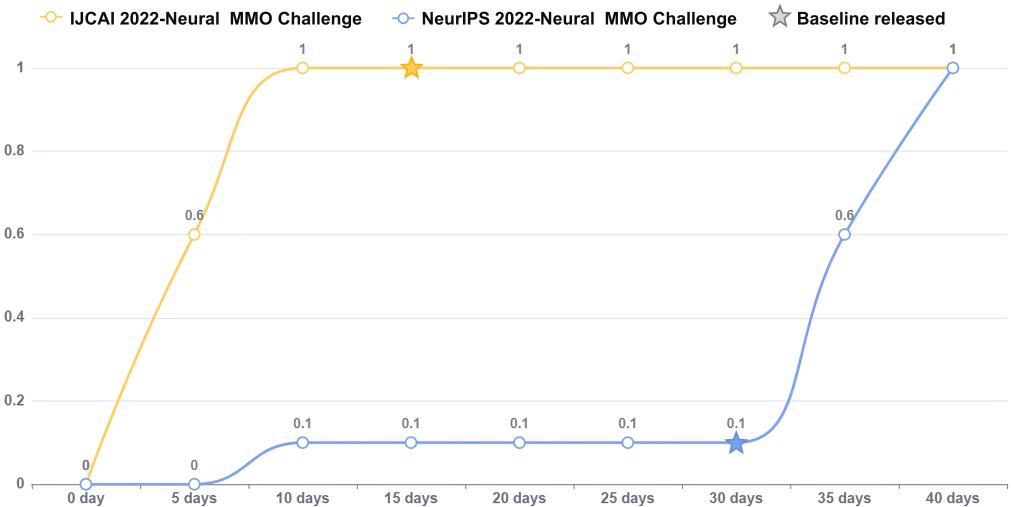


Figure 5: Submission performance over time in PvE Stage 1 of the past two competitions.

increases, participants require additional guidance. Therefore, we launched our RL baseline, which resulted in an increase in the frequency of submissions and breakthroughs in submissions by participants.

Apart from providing a baseline, we also offered a starter kit, QA, and live broadcasts to help participants better understand the competition’s progress and details, thereby improving their policies.

In summary, our tools and services enable participants to concentrate on implementing their ideas and refining their policies rather than spending time building from scratch.

## 5. Conclusion

We organized the NeurIPS 2022 Neural MMO challenge, which attracted over 500 participants from 35 countries and regions. The competition featured two tracks, PvE and PvP, and we observed that the same policy may perform differently on both tracks. In PvE, the competition is designed with a difficulty ladder. Most participants achieve a top1 ratio of 1 in the first phase, but only three players can achieve a top1 ratio of 1 in the second phase, and the score difference between participants is significant. This highlights the potential of this environment as a benchmark for algorithm robustness and generalization.

The environment introduces various mechanics, such as trading, skill differentiation, and many-agent cooperation, making it more interesting and suitable for MARL research. Analyzing the item distribution in the environment, we found some human-like trading behaviors, which surprised us.

Before the baseline was provided to participants, the number of submissions was minimal, and the majority of participants received unsatisfactory scores. This phenomenon reflects the increased difficulty of the environment. However, after the baseline was released, we observed an uptick in the number and quality of submissions. This highlights the importance of tools and services surrounding the environment for RL research, rather than relying solely on algorithmic innovation. We believe that with the advent of generic and flexible tools, there is room for substantial improvement to policy robustness and generalization on Neural MMO.

## References

- Brooke Chan Vicki Cheung Przemyslaw Debiak Christy Dennison David Farhi Quirin Fischer Shariq Hashme Christopher Hesse et al. Christopher Berner, Greg Brockman. Dota 2 with large scale deep reinforcement learning. *ArXiv*, abs/1912.06680, 2019.
- David R Ha Denny Britz Jakob N. Foerster Julian Togelius Kyunghyun Cho Joan Bruna Cinjon Resnick, Wes Eldridge. Pommerman: A multi-agent playground. *ArXiv*, abs/1809.07124, 2018.
- Jakob N. Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. *CoRR*, abs/1705.08926, 2017. URL <http://arxiv.org/abs/1705.08926>.
- Hangtian Jia and Yingfeng Chen Tangjie Lv Changjie Fan Hongyao Tang Jianye Hao Chunxu Ren, Yujing Hu. Mastering basketball with deep reinforcement learning: An integrated curriculum training approach. In *Adaptive Agents and Multi-Agent Systems*, 2020.
- Shu Bin Jiang and Chris Amato. Multi-agent reinforcement learning with directed exploration and selective memory reuse. *Proceedings of the 36th Annual ACM Symposium on Applied Computing*, 2021.
- Piotr Stanczyk Michal Zajac Olivier Bachem Lasse Espeholt Carlos Riquelme Damien Vincent Marcin Michalski Olivier Bousquet Sylvain Gelly Karol Kurach, Anton Raichuk.

Google research football: A novel reinforcement learning environment. *ArXiv*, abs/1907.11180, 2019.

Wojciech M. Czarnecki Michaël Mathieu Andrew Dudzik Junyoung Chung David H. Choi Richard Powell Timo Ewalds Petko Georgiev Junhyuk Oh Dan Horgan Manuel Kroiss et al. Oriol Vinyals, Igor Babuschkin. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, pages 1–5, 2019.

Tabish Rashid, Mikayel Samvelyan, Christian Schröder de Witt, Gregory Farquhar, Jakob N. Foerster, and Shimon Whiteson. QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning. *CoRR*, abs/1803.11485, 2018. URL <http://arxiv.org/abs/1803.11485>.

Aviv Tamar Jean Harb P. Abbeel Igor Mordatch Ryan Lowe, Yi Wu. Multi-agent actor-critic for mixed cooperative-competitive environments. *ArXiv*, abs/1706.02275, 2017.

Joseph Suarez and Phillip Isola. Specialization and exchange in neural mmo. In *ICLR Workshop on Agent Learning in Open-Endedness*, 2022.

## Appendix A. Environment Setting for NeurIPS Neural MMO Challenge

### A.1. Environment Wrapper

To make the environment more extensible and participant-friendly, we have incorporated some new functions in addition to the original environment. These changes include:

1. Random assignment of team numbers and spawns, which allows teams start at different initial positions on the map after multiple evaluations. This is helpful for addressing asymmetry in evaluation of participants’ algorithms.
2. Aggregation of observations of single agents at the team level, allowing each agent to have the information of other teammates. Our approach is different from OpenAI Five and AlphaStar as it does not strictly conform to the MARL definition within an individual team.

### A.2. Tracks

As shown in Fig. 6, the competition is divided into two tracks: PvE and PvP. In the PvE track, participants’ policies are evaluated against 15 built-in policies (i.e. copies of policies that we trained), while in the PvP track, they are evaluated against policies submitted by 15 other participants. From a competitor’s perspective, the PvE track is designed to help players quickly familiarize themselves with the environment against a fixed set of opponents, while the PvP track is intended to increase the competitiveness of the game. The structure of the different tracks is as follows:

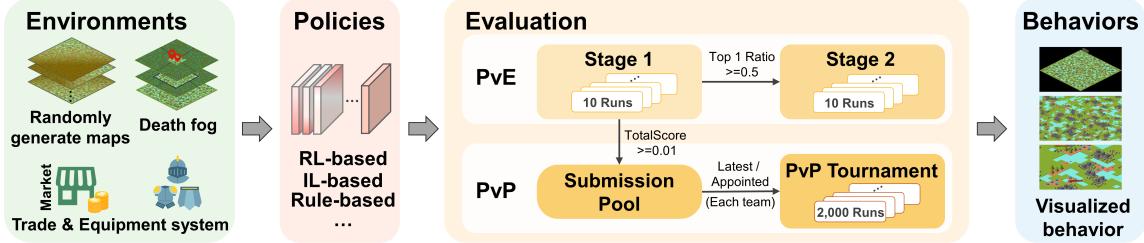


Figure 6: An overview of the track structure and evaluation process of the competition.

#### A.2.1. PvE STAGE 1 VS SCRIPTED AI

In this stage, participants’ policy will be thrown into a game with 15 scripted built-in AI teams and evaluated for 10 rounds. The result of each submission will be presented as a Top 1 Ratio (see section A.3). Each built-in AI team (Mixture Team and Combat Team) in stage 1 consists of 8 agents with 8 respective skills. These policies are open-sourced so that an evaluation environment is accessible during training. Each member of the mixture team specializes in a specific skill. The combat team is hostile and will attack every agent close to them. The aim of this track is to help new participants get familiar with the challenge.

#### A.2.2. PvE STAGE 2 VS RL-BASED AI

To qualify for Stage 2 in the PvE track, a participant must achieve a 0.4 Top 1 Ratio in Stage 1. In Stage 2, the rules are the same as in Stage 1, except that the built-in AIs are designed by the organizers and trained using a deep reinforcement learning framework developed by Parametrix.AI. This stage is intended to be more challenging than Stage 1, as the built-in AIs are much stronger, and defeating them is a significant achievement. There are three types of built-in AI teams in Stage 2:

1. Reckless: These bots are skilled in combat and also in using, selling, and buying items.
2. Ruthless: An evolution of the Reckless policy, Ruthless bots are both hostile and cooperative. They will fight enemies independently but also recognize the importance of team cooperation.
3. Coward: These bots are risk-adverse and avoid combat with high-level NPCs. They move quickly to the edges of the map to ensure a survival advantage and snipe incoming opponents for points.

#### A.2.3. PvP

In this stage, each participant’s policy is pitted against the policies sampled from the pool of qualifying submissions. Given the number of submissions from different teams employing different approaches, this procedure is an effective assessment of robustness and generalization. To ensure an accurate assessment, each participant will play at least 1000 matches.

### A.3. Metrics

#### A.3.1. SCORE

The participants' policy score is determined by the number of other agents (excluding NPCs) they defeat (i.e., inflict the final blow) and how long at least one team member remains alive. Each agent defeated earns 0.5 points for the participants' policy, and the final defeat score is the sum of the per-agent defeat scores across the team. The team's survival time is determined by when the last agent dies. The survival scores in each tournament range from 10 to 0. Specifically, the survival scores in each tournament from high to low are 10, 6, 5, 4, 3, 2, 1, 1, 0, 0, 0, 0, 0, 0, 0, and will be assigned to the team surviving the longest, the second longest, the third longest, and so on respectively. In the case of tied survival times, the team with more survivors ranks higher. If the number of survivors is equal, the average level of the agents is taken into account. If all factors are still equal, the tied teams will split the ranking scores. For example, if a policy lands the last hit on 5 agents and survives the longest, they receive  $5 * 0.5 + 10 = 12.5$  points.

#### A.3.2. TOP 1 RATIO IN PvE

The PvE track's fixed opponents offer a consistent performance metric for both scripted and reinforcement learned submissions. A Top 1 Ratio of 1.0 over 10 games (i.e. 10 wins in a row) indicates that the submission is significantly better than the built-in AI. This makes Top 1 Ratio a reasonable first screen of the performance of competitors.

#### A.3.3. TRUESKILL IN PvP

Top 1 Ratio is a poor indicator of performance in PvP evaluations against unknown opponents. We instead use a TrueSkill scoring mechanism that takes into account the ranking of all policies in a match, not just the winner. In other words, even if a strategy loses first place in a competition, its ranking under TrueSkill can still improve.

### A.4. Resources

#### A.4.1. STARTER KIT

This project includes environment installation instructions, submission demos, and FAQs for common problems to ensure that participants have a positive experience. Everything necessary for participants to submit their work is included.

#### A.4.2. BASELINE

To accommodate the limited device resources of most participants, we have developed a baseline using torchbeast that is available to all participants. This baseline enables participants to achieve a 0.5 Top 1 ratio on a PC with a GPU within one day.

#### A.4.3. ENVIRONMENT DOCUMENTATION

The documentation provided by the Neural MMO website is intended for a broader audience of researchers using the base Neural MMO platform for a variety of research agendas. This makes it difficult for competitors to get the information they need in a short time. To

help participants get started quickly, we have published a competition-specific document containing important information about the environment.

#### A.4.4. WEB VIEWER

We have created a browser-based video replay tool that allows participants to view their policy performance in terms of movement, collection, fighting, and trading, enabling them to make better policy adjustments. The web viewer includes different perspectives, controllable playback speed, and detailed information for display, as shown in Fig. 7.

#### A.4.5. EVALUATION SYSTEM

We have designed a distributed evaluation system based on k8s to handle large-scale submissions and fully evaluate the performance of each player’s policy. During the PvE phase, each player’s strategy will compete against built-in AI on a randomly generated map. In the PvP phase, all players in PvE stage 2 will compete against each other. For the PvP track, we provide both daily and weekly evaluations. Daily evaluations run only 100 matches against other participants. This functions as a more frequent feedback signal to help participants improve their policies. We also run a more extensive 1000 match evaluation once per week. As the performance of a policy is always relative opponents, this system helps to provide an accurate evaluation of player strategies.

## Appendix B. Policy Analysis

### B.0.1. DIFFERENT TYPES ALGORITHMS

We collected over 1600 policies and sorted participants into three categories based on the algorithms they used: rule-based, RL-based, and IL-based. Fig. 8 shows a comparison between these three types of algorithms. As depicted, most participants chose the RL algorithm for the challenge, and some of them achieved a top 16 ranking. However, a few participants submitted scripted policies, and one of the IL submissions achieved 5th place.

Overall, RL took participants longer to get up and running but delivered higher performance by the end of the competition. The theoretical upper bound of IL-based learning is close to that of RL but is limited by the quality of the data (which, in this case, is generated by an RL policy because human data is unavailable). Rule-based methods allow participants to control the actions of agents directly. We noticed that scripted approaches were quick to develop and effective at the start of the competition, but their performance fell off compared to RL by the end, and they were overly adapted to the opponents of the first two PvE stages.

### B.0.2. DIFFERENT TRACKS

The main challenges of this competition are robustness and generalization to new opponents in a complex environment. In the NeurIPS-2022 Neural MMO challenge, we set up two different tracks that can fully test the robustness and generalization of participants’ algorithms. As shown in Figure 9, the top 3 participants in PvE stage 1 all demonstrate good robustness and generalization. Realikun consistently wins in all tracks, while the rankings of the other two players remain steady. However, for most other participants, their rankings

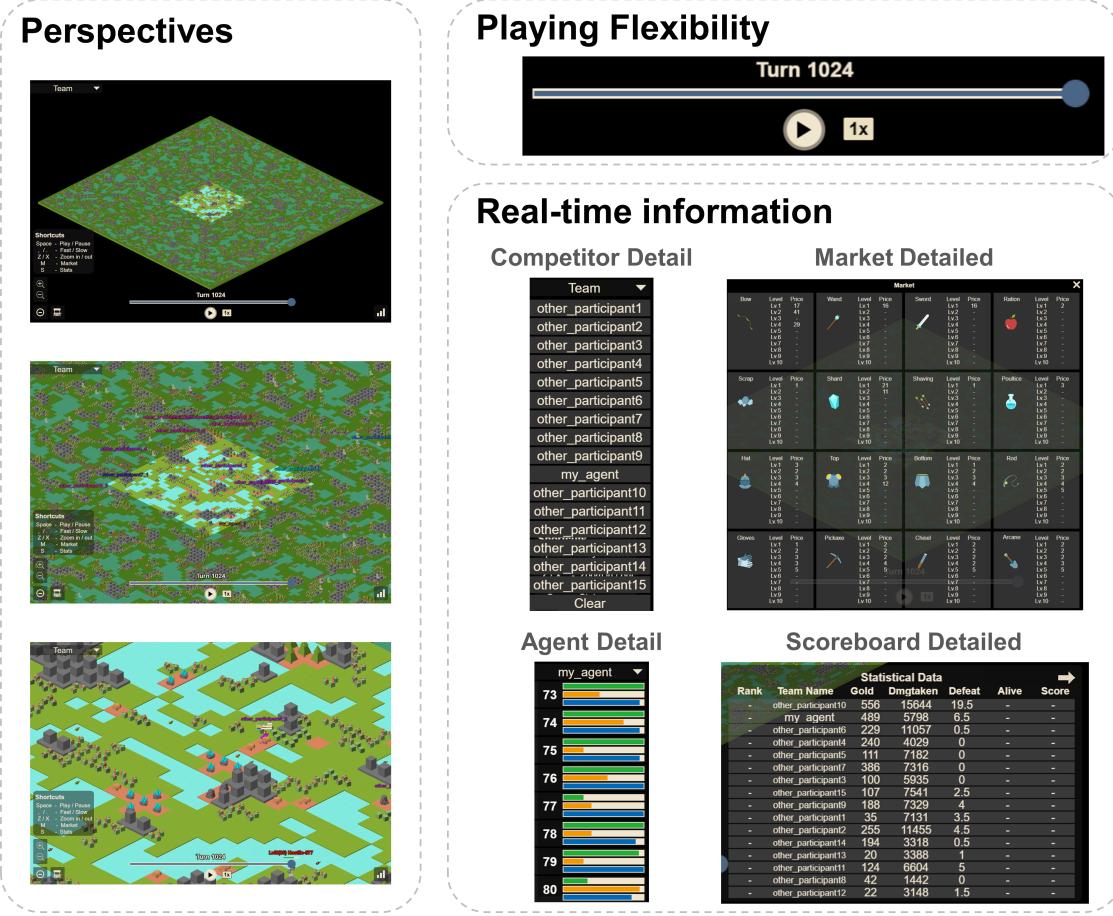


Figure 7: An overview of the web viewer. Left: various zoom levels. Top Right: draggable timeline. Bottom Right: Detailed statistics about the agent and market states.

change continually. For example, the Vanilla algorithm is weak on generalization, performing well in PvE stage 1 but poorly in PvE stage 2 and PvP stage. These observations demonstrate that this environment can facilitate the evaluation of policy robustness and generalization by introducing different opponents. They also illustrate that when a policy is robust enough, it is able to outperform all other policies in the environment.

#### B.0.3. DIFFERENT COMBAT PREFERENCES

After analyzing Fig. 4, we observed that even strategies with very close total scores behaved differently. We focus on the analysis of the strategies used by the top three players. As depicted in the figure, it is evident that realikun aimed to obtain higher points by defeating more participants. Typically, most participants believe that the maximum survival score is 10 points and that achieving this score will create a significant lead over their opponents. As a result, most participants tend to be relatively conservative and avoid fighting aggres-

sively to attain this score. However, if most participants are conservative, they may be at a disadvantage when fighting becomes inevitable later in the game. This allows a more

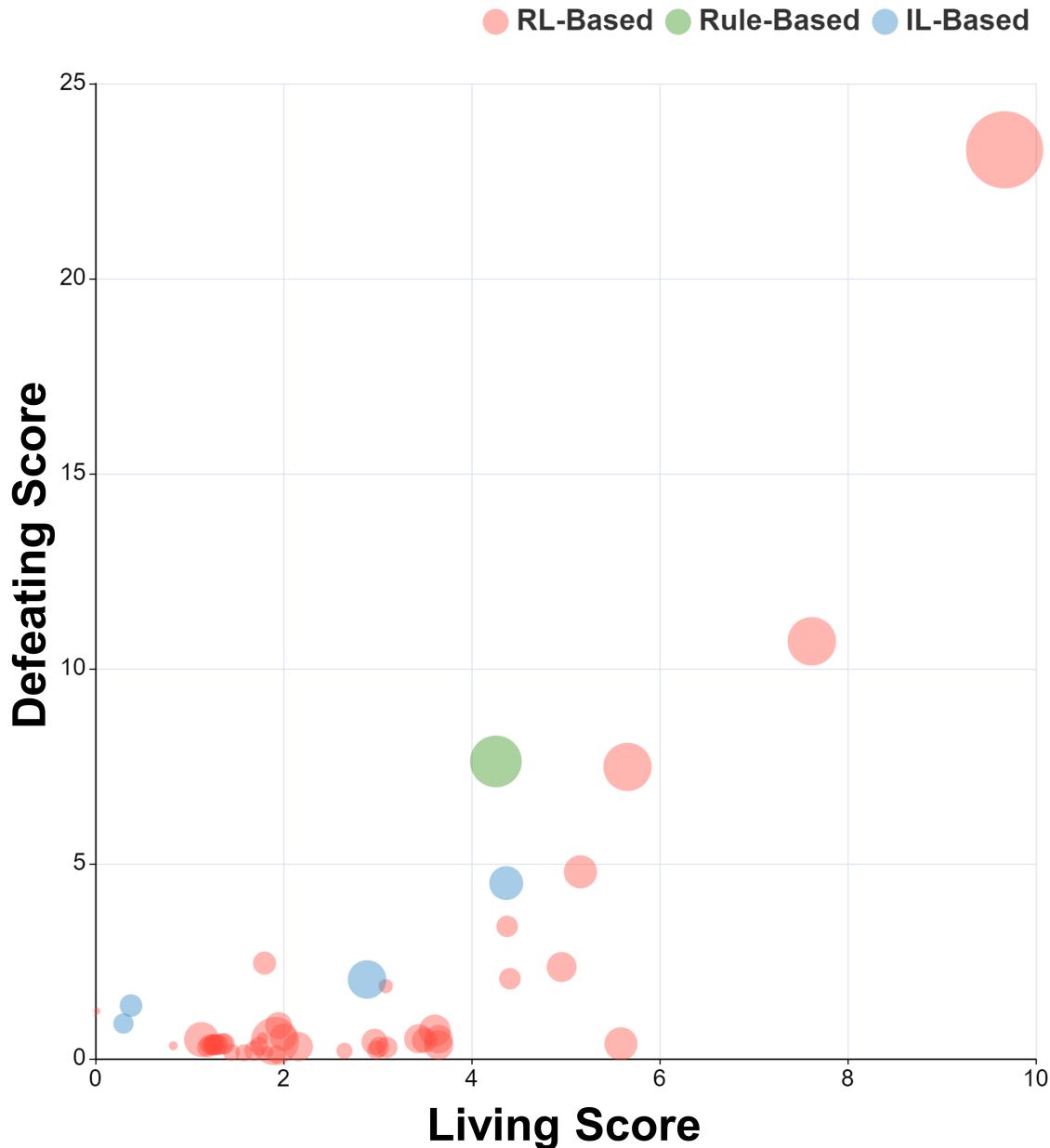


Figure 8: Performance of different type algorithms in the NeurIPS-2022 Neural MMO challenge PvP stage. Every point in the chart denotes a participant. The size of the point represents the total number of gold obtained by that participant.

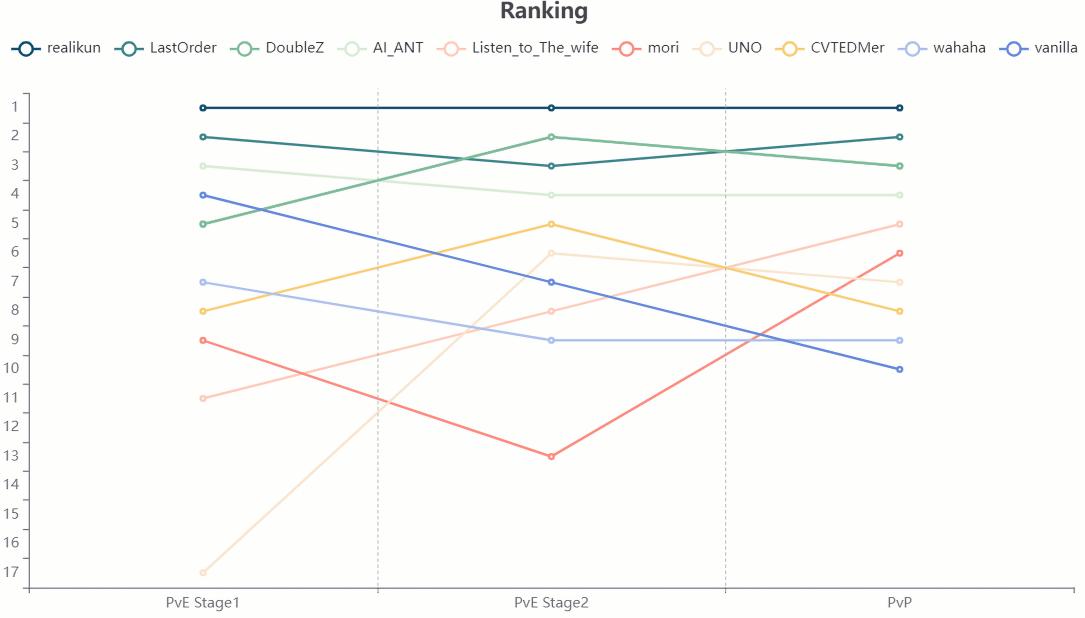


Figure 9: The ranking of participants in different tracks.

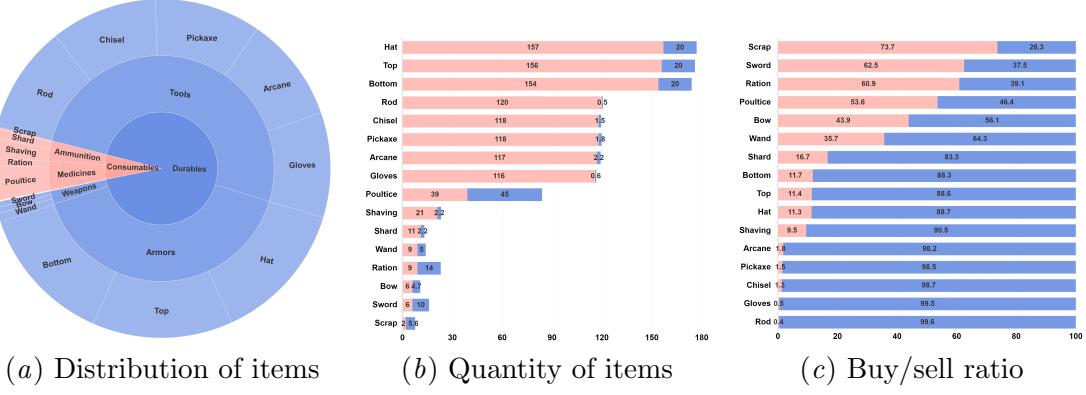


Figure 10: Details of the distribution of items in the NMMO environment.

aggressive policy to become substantially stronger than most participants, ensuring that it can defeat all other agents and survive to the end. Evidently, realikun is based on this idea and successfully won first place in all tracks of the competition. In contrast to realikun, doublez avoids unnecessary fighting (rarely fighting powerful NPCs in the environment), while passerby's agents are more aggressive. Passerby has more combat points than doublez, but it takes more damage and has lower survival rate, as shown in the figure.

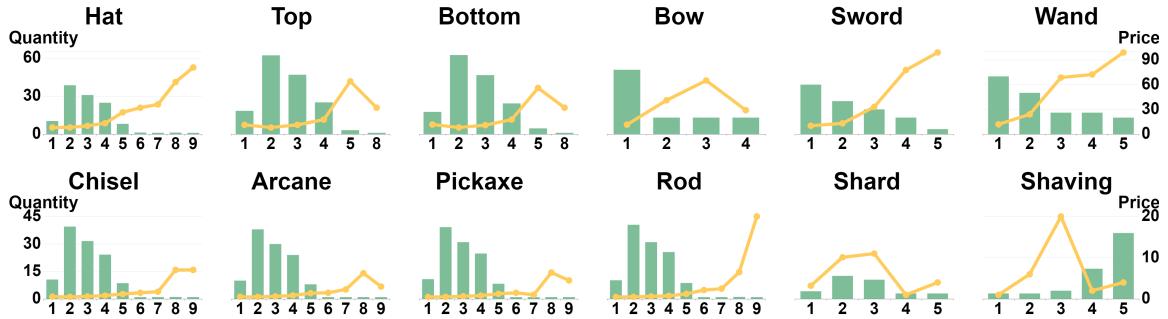


Figure 11: Quantity available and price of each item at various levels. Acquiring higher-level items is generally more difficult.

#### B.0.4. TRADE OVERVIEW

Figure 10 (a) shows the distribution of items in the environment, which has numerous tools and few consumables. Additionally, Figure 10 (b) depicts the amount of items, revealing that the total number of items of the same type is comparable, indicating the environment’s setting is reasonable. Based on this, we can deduce that different items hold different importance for agents, as shown in Fig. 10 (c). Agents prioritize owning weapons and restoration potions rather than selling them to the global market. However, since agents’ inventory is constrained, they need to trade their some supplies in the global market to make space. As seen in Fig. 11, we observe some interesting trading strategies: (1) For universally preferred items (such as weapons), agents’ offers usually increase with the level (difficulty to obtain) of the weapon. (2) Similarly, the value of such as tools also increases with their level, but if agents realize the expected price is too high to sell top-level supplies, they may reduce the price for trading.