The Third Neural MMO Challenge: Learning to Specialize in Massively Multiagent Open Worlds

Joseph Suarez*, **Sharada Mohanty**§, **Jiaxin Chen**†, **Hanmo Chen**†⋆,
Hengman Zhu†, Chenghui You†, Bo Wu†, Xiaolong Zhu†, Jyotish Poonganam§
Clare Zhu£, Xiu Li⋆, Julian Togelius‡, Phillip Isola*
*MIT, †Parametrix.AI, §AICrowd,
⋆Tsinghua Shenzhen International Graduate School, Tsinghua University,
£Stanford University, ‡New York University
jsuarez@mit.edu

Abstract

Neural MMO [SDZ+21] is an open-source environment for agent-based intelligence research featuring large maps with large populations, long time horizons, and open-ended multi-task objectives. We propose a benchmark on this platform wherein participants train and submit agents to accomplish loosely specified goals – both as individuals and as part of a team. The submitted agents are evaluated against thousands of other such user submitted agents. Participants get started with a publicly available code base for Neural MMO, scripted and learned baseline models, and training/evaluation/visualization packages. Our objective is to foster the design and implementation of algorithms and methods for adapting modern agent-based learning methods (particularly reinforcement learning) to a more general setting not limited to few agents, narrowly defined tasks, or short time horizons. Neural MMO provides a convenient setting for exploring these ideas without the computational inefficiency typically associated with larger environments.

Keywords

environment, reinforcement-learning, multi-agent, open-endedness, open-world

1 Competition description

1.1 Background and impact

The real world is a massively multiagent environment, and the ability to learn and reason within it is a hallmark of human intelligence. It seems inconceivable that artificial agents without this capability could operate intelligently within the real world outside of narrowly predefined tasks. Training directly within the real world has proven unwieldy because it often requires specialised hardware, is difficult to reproduce, and is generally expensive. The vast majority of tasks considered in modern reinforcement learning research are simulated environments. However, most of these are limited to a single or a few agents, short time horizons, and narrowly defined tasks.

In some sense, the lack of progress on more general environments can be attributed to a lack availability of such environments. `Neural-MMO` provides such a platform, and the purpose of the proposed benchmark is to spur research upon it. This would produce new methods for training many-agent policies, learning over long horizons, and navigating multi-modal and loosely specified tasks. The ultimate goal is, of course, to integrate associated findings into real-world agents – `Neural-MMO` is simply a computationally efficient setting for developing such methods.

Preprint. Under review.

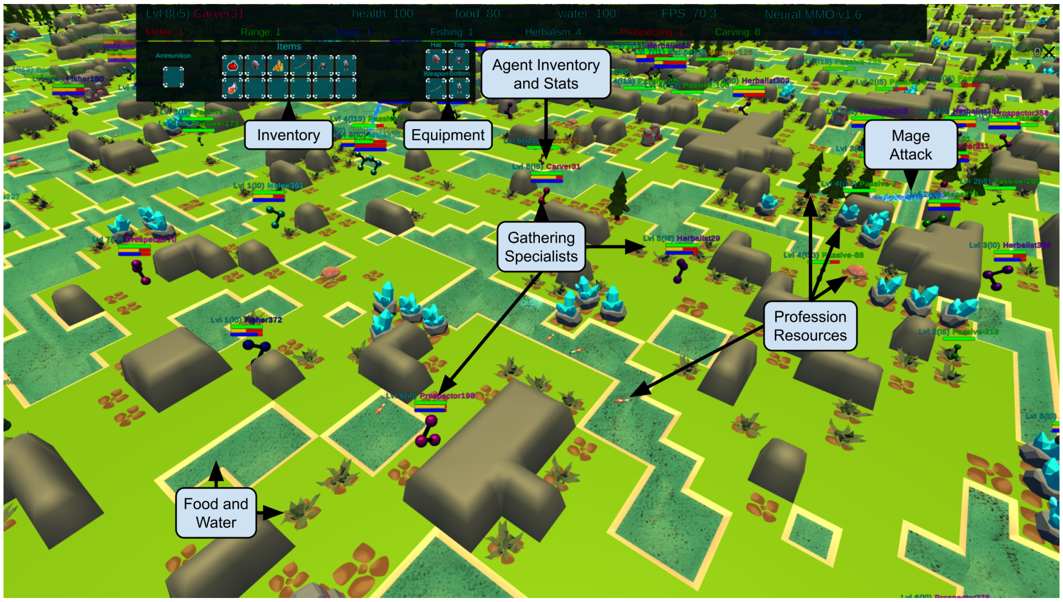

Figure 1: Annotated screenshot of a Neural MMO environment with profession and exchange systems. Around a hundred specialized agents are engaging in combat, gathering profession-specific resources, and exchanging goods on a global exchange.

Initial baselines on the `Neural-MMO` environment have revealed that simple single-agent reinforcement learning methods work surprisingly well on this new domain. This makes the competition accessible to the entire reinforcement learning community – specialization to multi-agent learning is not required. That said, the later rounds of the competition will explore team-based play within the `Neural-MMO` environment. We expect specialised cooperation-based algorithms to be useful here. However, many multiagent reinforcement learning algorithms make strong assumptions about the task structure that may not prove true in the more general `Neural-MMO` setting – it is possible that simpler methods may prevail. Understanding the scalability of various modern reinforcement learning methods to increasingly general environments is the other explicit goal of the competition.

1.2 Novelty

This will be the third in a series of competitions using the Neural MMO platform. The first competition was a smaller pilot not associated with a conference. The second is a larger run of the same competition at IJCAI this year and will be live at time of submission. The proposed NeurIPS 2022 competition would follow the IJCAI competition

The proposed competition would use a newer version of the Neural MMO platform [1] than the IJCAI competition. These releases implement substantially different environments. For comparison, agents in the IJCAI competition are mainly limited to basic foraging and combat tasks. As described in detail below, the new version includes a large item and equipment system, the ability to trade on a global exchange, and the ability to specialize. Learning to specialize would be the main task of the proposed competition, as compared to cooperation in the IJCAI version.

2 Data (Environment)

`Neural-MMO` is a game environment capable of hosting large number of agents in a interactive map.

Maps are procedurally generated with seven different types of resource. Two of these resources, food and water, are required by all agents to survive. The other five resources are gathered using the

[1]Full information available here: https://openreview.net/forum?id=H89qN7DbUWc. Environment description duplicated for convenience of the readers

associated profession. Figure 1 shows a map with generated terrain and resources. Any agent can harvest any resource, but agents trained in the corresponding profession will obtain better items. Of these five resources, two are used to make consumable items that restore food and water or health. The other three are used as ammunition in the three available combat styles. Combat is used to fight both other agents and scripted non-player characters (NPCs) which drop gold, equipment, weapons, and tools upon defeat. Equipment is useful to all agents as a means of defense. Each weapon type corresponds to a specific combat skill. Each tool type corresponds to a specific gathering skill. Agents can trade consumables, ammunition, weapons, tools, and equipment on a global market using gold. Figures 2 and 3 show the relationships between professions and items. Below, we describe each of these systems. All of the constants used below are configurable, but we provide the defaults to give a sense of scale.

The environment state is represented by a grid of tiles. Each tile has a particular assigned material with various properties, but it also maintains a set of references to all occupying entities. When agents observe their local environment, they are handed a crop of all visible game tiles, including all visible properties of the tile material and all visible properties of occupying agents. All parameters in the following subsystems are configurable; we provide only sane defaults obtained via multiple iterations of balancing.

2.0.1 Tiles

We adopt a tile-based game state, which is common among MMOs. This design choice is computationally efficient for neural agents and can be made natural for human players via animation smoothing. When there is no need to render the game client, as in during training or test time statistical tests, the environment can be run with no limit on server refresh rate. Game tiles are as follows:

- **Grass:** Passable tile with no special properties
- **Forest:** Passable tile containing food. Upon moving into a food tile, the agent gains food and the tile decays into a scrub.
- **Scrub:** Passable tile that has a 2.5 percent probability to regenerate into a forest tile on each subsequent tick
- **Stone:** Impassible tile with no special properties
- **Water:** Passable tile containing water. Upon moving adjacent to a water tile, the agent gains water.
- **Lava:** Passable tile that kills the agent upon contact

2.0.2 Agents

Input: On each game tick, agents observe a 15x15 square crop of surrounding game tiles, all occupying agents, their own inventory, and a global market. We extract the following observable properties:

Per-tile properties:

- **Material**: an index corresponding to the tile type
- **Row**: the row of the tile
- **Column**: the column of the tile

Per-agent properties:

- **Lifetime**: Number of game ticks alive thus far
- **Health**: Agents die at 0 health (hp)
- **Food**: Agents begin taking damage at 0 food or water
- **Water**: Agents begin taking damage at 0 food or water
- **Position**: Row and column of the agent
- **Damage**: Most recent amount of damage taken
- **Same Color**: Whether the agent is the same color (and thereby is in the same population) as the observer

- **Levels**: Levels of all agent skills
- **Item Level**: Total level of equipment held
- **Gold**: Amount of money held
- **Freeze**: Whether the agent is frozen in place as a result of having been hit by a mage attack

Per-item properties (separate for inventory and market):

- **Index**: Unique identifier for this item type
- **Level**: Overall quality of the item
- **Equipment Stats**: Bonuses to agent abilities if the item is equipment
- **Recovery Stats**: Amount of food/water/health restored if the item is consumable

Output: Agents submit one value for each of the following action types each time step: movement, attack, buy, sell, use item. The server ignores any actions that are not possible or permissible to fulfil, such as attacking an agent that is already dead or attempting to move into stone. *Pass* corresponds to no movement.

Movement:	**North**	**South**	**East**	**West**	**Pass**
Attack:	**Melee**	**Range**	**Mage**		
Buy:	**Item on market**				
Sell:	**Item in inventory**	**Price**			
Use:	**Item in inventory**				

2.1 Environment Mechanics

Survival: Agents start with 100 food, water, and health. They lose 5 food and water per time step and begin losing health if either hits 0. Agents die at 0 health. Health slowly regenerates if agents have over 50 food and water. Agents restore water by moving adjacent to a water tile and restore food by stepping on a food tile. Food tiles decay once harvested and replenish slowly over time.

Combat: There are 3 combat profession: melee, range, and mage. These use rock-paper-scissors dominance: mage beats melee, melee beats range, and range beats mage. In this context, attacking a specialized agent with their weakness inflicts 1.5x damage.

Gathering: There are 5 gathering professions: fishing, herbalism, prospecting, carving, and alchemy. The former two produce consumable items which restore food, water, and health. These act as supplies when exploring resource-sparse areas or as immediate healing in a pinch. The latter three professions produce ammunition that strengthens the attacks of corresponding combat styles.

Items: There are 17 types of items: 2 consumables, 3 armor pieces, 3 munitions, 3 weapons, 5 tools, and 1 currency (gold). Except for gold, each category contains items from levels 1 through 10, for a total of 161 unique items. Higher level consumables restore more food, water, and health. Higher level armor provides better defensive bonuses in combat. Higher level munitions and weapons increase damage in the respective combat style. Higher level tools enable gathering higher level items using the respective profession. Gold is inherently valuable as the only currency of exchange.

Progression: Each profession begins at level 1 and can be raised up to level 10. Experience is awarded for using the skill. For example, fighting with melee will raise the melee skill. Higher level agents can use higher level items. This places a strong emphasis on exchange with other agents as a means for acquiring relevant items.

Equipment: Agents can equip a helmet, chestplate, platelegs, held item, and ammunition. The former three armor pieces reduce damage taken in combat. Equipping weapons and ammunition increases damage with the associated combat profession. Note that munitions are consumed upon use, so combat agents must constantly replenish their supply. Wielding a tool confers a significant bonus to defense, enabling gathering agents to flee from poorly equipped aggressors. Equipping armor requires that agents have at least one skill of the same level of the armor piece. Equipping a held item or ammunition requires at least the same level in the associated skill. Higher-level tools enable agents to harvest higher-level resources. Higher-level ammunition and weapons enable agents to inflict more damage in combat.

Figure 2: Achievement system with point thresholds for the first competition. We are still finalizing achievements for the latest version of the environment. They will likely revolve around obtaining various items and reaching milestones in professions.

Exchange: Agents can buy and sell items on a global market using gold. Gold can be obtained from defeating scripted non-players and is inherently valuable because it is the sole currency of exchange. To sell an item, agents specify an item in their inventory and a price. To buy an item, agents purchase one of the current market offers.

2.1.1 Metrics (Achievement System)

`Neural-MMO` features an achievement system inspired by modern games. Agents may complete tasks of different difficulty (easy, normal, and hard) in a number of different categories, each awarding a different number of points. Figure 2 illustrates the achievement system. with point thresholds for the first competition on the platform.

By design, it is possible to complete easy tasks without much deviation in overall strategy. Medium tasks require significant planning and time investment. Hard tasks require commitment to a premeditated strategy.

Relative difficulty across tasks was calibrated using the scripted and pre-trained baseline models. We evaluated these baselines across several trials and adjusted task thresholds accordingly. The baselines solve easy tasks roughly 50% of the time, normal tasks 10% of the time, and hard tasks less than 1% of the time. We believe this to be a reasonable initial calibration, and we will refine these thresholds throughout the competition based on participant feedback.

Points are awarded per team: a task is considered completed if it is achieved by *any* agent on the team. This incentivizes specialization and division of labor – another important property of real-world intelligence that has been relatively unexplored in reinforcement learning.

2.1.2 Metrics (Tournaments)

Apart from the public leaderboard of the round, we will also maintain a Tournament Leaderboard for each of the rounds. Every week we will run a Tournament using the best submission from the top-N teams on the public leaderboard. The Tournament will be composed of multiple "Games" using `Neural-MMO` where user submitted policies will be the "players". The Game specific rankings will be determined by the in-game achievement points (as described in Section 2.1.1), which will be used to update the TrueSkill ratings for the individual players (user submitted policies). The updated TrueSkill ratings are then used to match groups of "players" optimally for the subsequent games. These games are iteratively conducted until the TrueSkill Ratings converge. The final TrueSkill ratings will be then used to update the weekly Tournament Leaderboard. The winners of the competition will be determined by the final standings on this Tournament Leaderboard.

Honorary mentions may be awarded for approaches that stand out in other ways, such as using a particularly small model, a simple training scheme, or lack of reward shaping.

2.2 Baselines, code, and material provided

`Neural-MMO` includes two scripted baselines and a pre-trained recurrent network with associated training code. Evaluation results for all three models are available on the project homepage above (this is currently the previous version of the environment – we are in the process of updating all materials).

- **Baselines:**
  `https://github.com/NeuralMMO/baselines`
- **Source Code:**
  `https://github.com/NeuralMMO/environment`
- **API:**
  `https://neuralmmo.github.io`

Participants will have access to the full source code of `Neural-MMO`: environment, map generator, evaluations, renderer, and visualizations. The baselines above include two strong scripted models and one pretrained recurrent model with all associated source and training code. We have also provided API documentation and guides for the various components of the environment.

We expect participants with access to a single GPU to be able to take part comfortably. We are also working on a sponsorship for cloud credits. Note that even our baseline was trained using a single commodity GPU for less than a day. We believe this efficiency is currently unique to the `Neural-MMO` platform: there are no other comparably complex many-agent environment, and many of the best few-agent environments require industry-scale hardware (e.g. DoTA, Starcraft, Go, DeepMind Capture the Flag, OpenAI Hide and Seek).

3 Organizational aspects

3.1 Protocol

3.1.1 Platform

The evaluation of the submissions will be managed by AIcrowd. The platform offers state-of-the art technology and management specifically for running competitions like the one proposed here. The technology underlying AIcrowd has been tried and tested in over 50 competitions, including many at NeurIPS (NeurIPS 2020 Procgen Challenge, NeurIPS 2020 Flatland Challenge, The Learning to Run Challenge in 2017, 2018, 2019, the AI for Prosthetics Challenge in 2018, and the Adversarial Vision Challenge in 2018, Disentanglement Challenge in 2019, MineRL Competition in 2019/2020, REAL Robots Challenge in 2019). The AIcrowd platform offers to the full range of services for a competition, from user management to discussion forums and leaderboards. The evaluation infrastructure can be flexibly adapted to any execution requirements, and allows for full traceability and reproducibility of the submissions.

3.1.2 Submission Protocol

Throughout the competition, participants can work on their code bases as private git repositories on `https://gitlab.aicrowd.com`. The requirements of the AIcrowd evaluators require participants to package their intended software runtime in their repositories, to ensure that the evaluators can automatically build relevant Docker images from their repositories, and orchestrate them as needed by the evaluators of the particular round. This approach also ensures that all the user submitted code that is successfully evaluated in context of this competition is both versioned across time, and also completely reproducible.

Submission Mechanism. Participants can pack their evaluation codes and the model into an archive file through the AICrowd UI page. On the successful evaluation of the submission, the scores and any relevant artifacts(generated media, etc) are added automatically to the leader-board.

3.2 Rules

1. Do not make any effort to intentionally overwhelm our evaluation servers.
2. Participants may mark up to three submissions for inclusion in the tournament at any given time.

3. Participants may adjust their algorithm, training scheme, and reward in response to aggregate observations of other agents, but do not hard-code exclusive targeting of a particular submission or subset of submissions (harassment) or similar nonaggression (teaming).
4. Reproducible training code is required for prize eligibility. We will contact the winners individually if they qualify for a prize.
5. Attempting to circumvent any of the above will result in disqualification.

We will be monitoring the submission portal for suspicious activity including duplicate or similar models or weights from different users. Contact us if you believe your submission is being individually targeted by another competitor or if you see signs of teaming. We will investigate all accusations thoroughly and disqualify any participants for whom we find conclusive evidence of cheating.

3.3 Challenge Structure

3.3.1 Evaluation Stages

We will evaluate your agent in two stages.

Stage 1: Verses Scripted Bots We will evaluate your agent against scripted baselines of a variety of skill levels. Your objective is to earn more achievement points (see Evaluation Metrics) than your opponents. We will estimate your agent's relative skill or match-making rank (MMR) based on several evaluations on different maps. We generate these maps using the same algorithm and parameters as provided in the starter kit, but we will use another random seed to produce maps outside of the direct training data.

Stage 2: Verses Other Participants We will evaluate your agents against models submitted by other participants as well as our baselines. When there are only a few submissions at the start of the competition, we will take a uniform sample of agents for each tournament. Once we have enough submissions from the participants, we will run tournaments by sampling agents of similar estimated skill levels. Your objective is still to earn more achievement points than your opponents.

4 Participants

The number of teams is expected to be about 50 and the number of participants is expected to be about 100, with teams consisting of up to 4 participants. The estimation is based on the previous Neural MMO competition organized by AIcrowd[2] and the NetHack Competition [3].

5 Prizes

We plan to provide cash prizes for winners. The money will be sponsored by Parametrix.AI and the total cash pool is 20,000 US Dollars. The prizes are designed as follows.

- No.1: $ 5000
- No.2: $ 3000
- No.3: $ 2000
- No.4-No.10: $ 1000 $\times$ 6
- Community Prize: $ 1000 $\times$ 4

Additionally, we plan to set up a Community Prize for participants who substantially contribute to the Neural MMO community and help foster the challenge.

[2] https://www.aicrowd.com/challenges/the-neural-mmo-challenge
[3] https://www.aicrowd.com/challenges/neurips-2021-the-nethack-challenge

6 Source Code

The Neural MMO environment is open sourced[4] with MIT license. Also we will give a starter kit and a baseline for the starters. We encourage the participants to open-source their code after the competition but it is not forced.

7 Timeline in Milestones

Timeline of the competition:

- 29 July: Send out call for participation.
- 29 July - 5 August: Test period, to see if the baselines work for participants.
- 10 August: Competition starts.
- 11 November: Submission Deadline.
- 18 November: Winner Announcement.

Although this is later than the suggested timeline, we would still be able to present the results of the competition at (virtual) NeurIPS. Winners would be formally declared later in December after we have verified the integrity of top submissions. We can also push the whole competition earlier if desired – this proposal simply aims to minimize overlap with the competition tracks of other conferences, most notably IJCAI.

8 Post-Proceedings / Publications

We are interested in preparing one paper of our challenge including the most interesting agents / algorithms / techniques developed by participants.

9 Organizer Backgrounds and Roles

- Joseph Suarez: Creator of Neural MMO, organizer in the first Neural MMO Challenge and co-organizer in this challenge
- Sharada Mohanty: AICrowd founder, organizer in the first Neural MMO Challenge and co-organizer in this challenge
- Jiaxin Chen: Senior Research Scientist at Parametrix.AI, co-organizer of this challenge
- Hanmo Chen: Winner of first Neural MMO Challenge, in the AI master program of Tsinghua Shenzhen International Graduate School, research intern at Paramatrix.AI
- Xiaolong Zhu: Senior Director at Parametrix.AI, competition advertising and university relations
- Xiu Li, Professor at Tsinghua Shenzhen International Graduate School, advising on challenge organization
- Clare Zhu: Data scientist involved in early versions of Neural MMO, advising on accessibility to non-RL methods/backgrounds
- Julian Togelius: Associate Professor, NYU, expert in procedural generation with experience on Neural MMO
- Phillip Isola: Assistant Professor, MIT, Advisor of Neural MMO for 3+ years

References

[SDZ+21] Joseph Suarez, Yilun Du, Clare Zhu, Igor Mordatch, and Phillip Isola. The neural mmo platform for massively multiagent research. In J. Vanschoren and S. Yeung, editors, Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks, volume 1, 2021.

[4] `https://github.com/NeuralMMO/environment`