Dumoba: Reinforcement Learning in a MOBA-Like Environment

Lance Yi

Abstract

This project introduces **Dumoba**, which is a simplified Multiplayer Online Battle Arena (MOBA) game-like environment developed to study how multi-agent reinforcement learning can handle different situations under a MOBA like environment. Inspired by projects such as OpenAI Five for *Dota 2*, due to time and cost limit, Dumoba serves as a simplified but strategically meaningful domain where agents must navigate, cooperate, and defeat an opposing team while protecting their own base. Using Proximal Policy Optimization (PPO), we trained agents in asymmetric 3v5 scenarios, fighting against scripted opponents and evaluated their performance under multiple difficulty settings. Despite relying on sparse terminal rewards and minimal state encoding, PPO agents learned structured behaviors well. A full 5v5 self-play system with more refined rules and policies is under development. Dumoba provides a flexible, controllable ,and accessible platform for potential future multi-agent RL research in adversarial domains.

1. Introduction

Team-based instant reaction strategy games such as *StarCraft II*, *Dota 2* ,and *League of Legends* are considered the most complex and dynamic among all game domains for artificial intelligence. These games require agents to reason over long time horizons, adapt to partial observability, coordinate with teammates, and combat intelligent adversaries. Landmark achievements such as **OpenAI Five**, which defeated a former *Dota 2* world champion team OG, demonstrate the power of deep reinforcement learning (RL) when trained at scale.

However, high-fidelity MOBA environments require immense computational resources, which makes them impractical and alomst impossible to achieve or reproduce for small or individual academic projects. To better bridge this gap, we developed **Dumoba**-a simplified game built on Python. It captures the core ideas of lane pushing, hero combat, and base destruction, but eliminates unnecessary graphical and mechanical complexity. This design allows rapid iteration on RL algorithms, short training time taken, controlled experimentation, and transparent inspection of agent behavior.

The primary goals of this project were to:

1. **Build a simplified but strategically relevant MOBA environment.**
2. **Train PPO agents to coordinate in asymmetric combat scenarios.**
3. **Evaluate performance under multiple difficulty modes.**
4. **Analyze the effectiveness of sparse reward training.**
5. **Begin the development of a 5v5 PPO self-play system.**

We demonstrate that PPO agents are capable of learning meaningful strategies even with minimal reward shaping, although certain limitations restrict emergence of advanced tactics. This report describes the design of the current Dumoba environment, the PPO training framework, experimental results, limitations, and avenues for extension.

2. Related Work

Reinforcement learning in complex MOBA games has advanced significantly in recent years. Key contributions include:

OpenAI Five with Dota 2 (Berner et al. 2019).

The StarCraft Multi-Agent Challenge for multi-agent RL (Samvelyan et al. 2019).

And Proximal Policy Optimization (PPO) used widely in similar projects(Schulman et al. 2017).

Unlike most large-scale projects, Dumoba is intentionally lightweight and friendly for individual level training. Its purpose is not to compete with the most advanced RL MOBA agents, but to provide a controlled environment where algorithmic behavior can be studied in isolation, without the confounding variables of full game engines or massive action spaces.

3. Environment Design

3.1 Overview

Dumoba models the essential mechanics of a MOBA:

- Two opposing teams: **blue** (PPO controlled learning agents) and **red** (scripted agents).
- Grid-based movement and navigation.
- Hero and base health(HP) and damage in combat.
- Team victory conditions.

The map is a 2D grid with only one lane or a big playground connecting the two bases. While there are no items, spells, or complex hero abilities, the environment preserves the need for positioning, timing, and complex cross-hero-coordinations.

3.2 Observation Model

Each agent receives a combination of global and localized information. Observations include:

- Hero positions and HP values.
- Base positions and remaining base HP.
- Grid occupancy (enemy units, allied units, base locations).
- Environmental features such as obstacles.

To simplify training, all features are flattened into a single state vector for use with a multilayer perceptron (MLP). Although this sacrifices the potential and more realistic spatial structure, it provides a consistent fixed-size input which is more ideal for PPO.

3.3 Action Space

Each hero controls:

- **Movement** in four simple directions.
- **Attack** to an enemy or base.

Actions are discrete and compact, allowing efficient exploration and reducing the complexity of credit assignment. Agents choose actions independently, but training encourages cooperative behavior through a shared reward policy.

3.4 Opponent Policies

The red team uses **rule-based scripted logic**, designed to be predictable but still competitive:

- Move toward the nearest visible enemy or opposing base.
- Attack when within range.
- Maintain simple formation to avoid blocking teammates.

Multiple difficulty modes adjust:

- Red hero HP
- Red hero damage
- 1 time respawn mechanics

This enables scalable evaluation of policy robustness.

---

4. PPO Training

4.1 Algorithm

Proximal Policy Optimization (PPO) was selected because:

- It is stable for continuous and discrete control.
- It handles environment stochasticity well.
- It was the fundation of OpenAI Five's project, aligning our work with established literature.
- It performs reliably in multi-agent self-play settings when agents have a shared policy.

PPO acts as a policy-gradient method with clipped updates to prevent instability.

4.2 Neural Network Architecture

The agent's policy and value networks use a feed-forward multilayer perceptron (MLP). Although convolutional layers could exploit spatial structure, flatten map proved effective and reduced the computational costs.

Hidden layers use ReLU activations, which is chosen for:

- Efficient gradient propagation
- Suitability for dense numerical input
- Consistency with PPO literature

4.3 Reward Structure

The training rewards are intentionally sparse, mirroring real MOBA endgame objectives:

- +1 for winning
- -1 for losing
- -0.01 per timestep to encourage faster victories

No direct rewards were implemented for:

- Dealing damage
- Hero kills
- Taking map control

These reward extensions are proposed for future work and expected to allow a more aggresive and realistic tactical behavior.

5. Experiments and Results

5.1 Evaluation Procedure

The final trained PPO model was evaluated under three difficulty modes:

- **Easy:** Red heroes have symmetric HP and damage.
- **Medium:** Red heroes are slightly buffed.
- **Hard:** Buffed Red heroes can respawn once.

Each mode has 100 episodes for training.

5.2 Results

Difficulty	PPO Win Rate	Interpretation
Easy	99%	PPO agents reliably defeat scripted bots.
Medium	84%	Reduced performance due to stronger opponents.
Hard	0%	Respawns make the game impossible to win.

5.3 Behavioral Analysis

Across matches, PPO agents learned to:

- Group together before engaging enemies.
- Prioritize attacking isolated enemy heroes and trade HP wiser.
- Push toward the enemy base when advantageous.
- Avoid unnecessary idle behavior due to time-penalty.

However, as the difficulty incraeses, especially after the introduction of respawn system, the PPO can barely handle the instable evironment of respawned enemies.

5.4 Self-Play Progress

A **5v5 self-play training system** is implemented and currently under active training. Initial expectations are:

- More diverse strategies than scripted-opponent training.
- Early signs of lane splitting and temporary retreating behaviors.
- Longer training cycles required due to increased complexity.

Due to time constraints, complete results could not be included in this report, but the system forms the core of planned future work.

---

6. Limitations

Despite promising results, several limitations remain:

1. **Sparse rewards limit rich tactical development.**
   Without shaping for damage, kills, or positioning, agents sometimes adopt overly direct strategies.
2. **Flattened observation removes spatial structure.**
   Spatial relationships are encoded but not exploited as effectively as with convolutional or attention-based models.
3. **No communication model.**
   All coordination emerges implicitly; explicit message-passing could improve strategies.
4. **Scripted opponents lack strategic diversity.**
   They are too predictable, which reduced the dynamic challenges PPO must overcome.

5. **Training computational constraints.**

   Full-scale self-play requires more time than available.

---

7. Future Work

This project is a foundation for further research. Planned extensions besides triditional MOBA rules like hero class, hero abilities, and fog of wars include:

1. **Reward for Damage and Kills**

   This is designed for more aggresive tactical decisions.

2. **Base Damage Reward Scaling**

   Encourages structured lane pushing and objectives beyond hero elimination.

3. **Self-Play 5v5 Training**

   Already implemented and currently running; expected to produce more sophisticated behaviors than scripted training.

4. **Convolutional or Attention-Based Networks**

   Better exploit spatial relationships.

5. **Dynamic Maps and Terrain Features**

   Tests generalization ability across map configurations.

---

8. Conclusion

Dumoba, a custom-built MOBA like environment, is built in this project as a experimental platform for multi-agent reinforcement learning.

Through PPO training, agents learned coordinated strategies, achieving strong performance under easy and medium game modes. Early efforts toward self-play demonstrate the potential for emergent complexity beyond scripted behaviors. While resource and time limitations restricted the full implementation of advanced mechanics, the project establishes both the environment and the RL framework needed for future experimentation.

Dumoba will continue to evolve as a sandbox for studying coordination, adversarial learning, and robust policy development in team-based settings. And hopefully one day, we can develop a fully functional platform for well-trained agents play against human.

References

Berner, Christopher, Greg Brockman, Brooke Chan, Vicki Cheung, Przemyslaw Debiak, Christy Dennison, David Farhi, et al. 2019. "Dota 2 with Large Scale Deep Reinforcement Learning." *CoRR* abs/1912.06680. http://arxiv.org/abs/1912.06680.

Samvelyan, Mikayel, Tabish Rashid, Christian Schröder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philip H. S. Torr, Jakob N. Foerster, and Shimon Whiteson. 2019. "The StarCraft Multi-Agent Challenge." *CoRR* abs/1902.04043. http://arxiv.org/abs/1902.04043.

Schulman, John, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. "Proximal Policy Optimization Algorithms." *CoRR* abs/1707.06347. http://arxiv.org/abs/1707.06347.