MULT-STAGE PHASIC POLICY GRADIENT FOR HOK 1V1

Yixing Chen, Songlin Jiang, Qihao Luo
ACM Honor Class
Shanghai Jiaotong University
Shanghai, China
{polaris_dane,clorf6,sjtu18815199753}@sjtu.edu.cn

ABSTRACT

We introduce a reinforcement learning framework designed to train robust agents for 1v1 MOBA game environments, integrating an enhanced variant of the Phasic Policy Gradient (PPG) algorithm with a structured multi-phase reward curriculum. The framework decouples policy and value function optimization to enhance training stability and sample efficiency, while a progressive reward curriculum transitions the agent from mastering core mechanisms (e.g., skill execution and positioning) to advanced strategic decision-making (e.g., resource control and adaptive engagements). Experimental evaluations demonstrate superior performance over state-of-the-art PPO-based methods, attaining win rates exceeding 90% against competitive baselines and exhibiting sophisticated in-game tactics such as adaptive target prioritization and skill combos. Systematic ablation analyses further validate the synergistic impact of algorithmic innovations (e.g., phased gradient updates) and multi-stage reward engineering.

1 INTRODUCTION

The rapid advancement of deep reinforcement learning (DRL) has propelled artificial intelligence (AI) to achieve superhuman performance in complex competitive environments, from board games like Go to real-time strategy (RTS) games such as StarCraft. Among these challenges, the 1v1 mode of *Honor of Kings* (HoK), a popular Multiplayer Online Battle Arena (MOBA) game, stands out as a benchmark for testing AI's decision-making capabilities in high-dimensional, partially observable, and dynamic scenarios. In this mode, the state and action spaces are astronomically large (approximately 10^{600} and 10^{18000}, respectively), far exceeding the complexity of traditional games like Go (Ye et al., 2020b). Unlike 5v5 team mode, 1v1 emphasizes individual tactical skills, including target selection, skill combos, and adaptive strategies against diverse opponents, making it an ideal testbed for exploring hierarchical action spaces and generalization in competitive reinforcement learning.

Despite all the progress, existing methods face challenges in scalability, sample efficiency, and robustness. For instance, early approaches relied on rule-based AI or Monte Carlo tree search (MCTS), which struggled to handle the game's real-time dynamics. Recent work, such as Tencent's ”Juewu” AI, demonstrated the potential of DRL frameworks by defeating top human players with a 99.8% win rate. However, these systems often depend on massive computational resources (e.g., 600,000 CPU cores and 1,064 GPUs) (Ye et al., 2020b) and lack standardized benchmarks for a fair comparison. Moreover, the absence of high-quality offline datasets for MOBA 1v1 scenarios limits research reproducibility.

This paper addresses these issues by proposing a multi-stage DRL training framework, which integrates time-varying reward, value normalization, and Phasic Policy Gradient (PPG). Our work builds on the *HoK 1v1* environment, which provides standardized APIs for agents to control heroes and receive combat-related information. By combining various training methods, we aim to obtain a competitive AI agent with limited data and computational resources.

2 Related Works

2.1 MOBA Games as AI Testbeds

MOBA games have emerged as critical platforms for AI research due to their strategic depth and real-time complexity. Early studies focused on rule-based systems or simplified environments, but the release of Honor of Kings Arena (Wei et al., 2022) marked a shift toward scalable, open-source frameworks. This environment supports 20 heroes with unique skills and provides 400 task combinations, enabling studies on cross-hero generalization. Similarly, Hokoff (Qu et al., 2023) introduced offline RL benchmarks for both single- and multi-agent scenarios, filling a gap in reproducible research.

2.2 Enhancements to PPO-Based Policy Optimization Frameworks

Proximal Policy Optimization (PPO) has become a cornerstone in reinforcement learning due to its balance of stability and computational efficiency. However, challenges persist in high-dimensional action spaces and long-horizon tasks, where simultaneous optimization of policy and value networks often leads to gradient interference and instability. Recent advancements address these limitations through refined mechanisms:

- **Dual-Clipping**: Building on the original clipping mechanism, dual-clip PPO (Ye et al., 2020b) introduces a secondary constraint to bound policy updates in large-batch or off-policy scenarios, preventing overshooting during parameter updates.
- **Value Normalization**: By rescaling value (advantage, reward, etc.) estimates within mini-batches, this method stabilizes gradient directions and mitigates skewed updates caused by outlier trajectories (Yu et al., 2022).
- **Phasic Optimization**: Phasic Policy Gradient (PPG) (Cobbe et al., 2021) decouples policy and value updates into distinct phases, resolving representation conflicts through prioritized replay buffers and staggered training cycles. These innovations collectively enhance sample efficiency and convergence robustness in complex decision-making environments.

2.3 Reward Shaping and Curriculum Learning

Reward shaping has long been used to guide learning in sparse- or delayed-reward environments by providing more frequent and informative feedback.In complex games such as MOBA, where the overall objective may require hundreds or thousands of decisions, shaping rewards can help agents learn intermediate goals like resource gathering, positioning, or damage dealing. Complementary to reward shaping, curriculum learning (Ye et al., 2020a) introduces structured progression through tasks of increasing difficulty, allowing agents to build up skills incrementally.These strategies have proven effective in environments with hierarchical objectives or multiple game stages, such as real-time strategy and multiplayer games.

3 Methods

3.1 Phasic Policy Gradient (PPG)

Traditional PPO architectures often share base network layers between policy and value functions, creating potential conflicts in gradient updates, especially from value estimation errors. To address this training instability and enhance data efficiency, we implement Phasic Policy Gradient (PPG) as an alternative to Proximal Policy Optimization (PPO). PPG explicitly separates policy optimization from value function training through a dual-phase framework, mitigating gradient interference between these components.

During the policy phase, we adopt the dual-clip PPO method to update the policy. To prevent value-related gradients from affecting the shared representation, the input to the value head is explicitly detached from the computational graph. This ensures that only the policy loss and entropy loss contributes to updates of the shared encoder. The policy loss is defined as:

$$\mathcal{L}_{\text{PPO}} = -\mathbb{E}_t \left[\max\left(\min\left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right), c\hat{A}_t \right) \right] \tag{1}$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$ is the probability ratio, $\hat{A}_t$ is the estimated advantage, and $c > 1$ is a bounding constant.

After a fixed number of policy phase updates, we switch to the auxiliary phase, where the model is updated using data cached during the policy phase. Two loss components are optimized: (1) a mean squared error (MSE) loss between the predicted value and a stored target value to improve value function accuracy:

$$\mathcal{L}_{\text{value}} = \frac{1}{2} \left\| V_\theta(s_t) - V_{\text{target}} \right\|^2 \tag{2}$$

and (2) a Kullback-Leibler (KL) divergence loss between the current policy output and a stored copy of the old policy distribution, encouraging consistency in action probabilities:

$$\mathcal{L}_{\text{KL}} = \text{KL} \left[\pi_{\theta_{\text{old}}}(\cdot \mid s_t) \parallel \pi_\theta(\cdot \mid s_t) \right] \tag{3}$$

The total auxiliary loss is defined as:

$$\mathcal{L}_{\text{aux}} = \mathcal{L}_{\text{value}} + \beta \cdot \mathcal{L}_{\text{KL}} \tag{4}$$

where β is a weight coefficient for the distillation loss.

3.2 Advantage Normalization

Among the reinforcement learning algorithms based on the advantage function, the calculation of policy gradient relies on the advantage $A(s, a)$, i.e. the degree to which an action is ”good” or ”bad” compared to the average action at a given state. However, raw advantage values often exhibit high variance, potentially destabilizing policy updates.

Advantage Normalization is a simple and effective trick that stabilizes the training process. In each batch, the normalized advantage $\hat{A}$ is computed as:

$$\hat{A} = \frac{A - \mu_A}{\sigma_A + \epsilon} \tag{5}$$

where μ_A, σ_A is separately the mean and standard deviation of the advantages in the current batch, and ϵ is a small constant (e.g, 1×10^{-8}) to avoid division by zero.

The normalization does not alter the direction of policy gradient, but helps mitigate the impact of outlier values and stabilizes training. In our experiments,

3.3 Reward Shaping

We propose a multi-stage and time-varying reward mechanism, in which the reward weights are dynamically adjusted over the course of training. The specific time schedules and corresponding weight configurations are detailed in Table 1. This design reflects the evolving strategic priorities during a single game. For instance, as the game progresses, killing an enemy hero or dying leads to increasingly significant consequences due to longer revive times. Likewise, the ultimate skill hit rate becomes more crucial, as successful ultimate skill usage often provides a decisive advantage in subsequent engagements.

Table 1: Reward Design

Reward	Weight	Time Scaling Factor	Meaning
hp_point	5.0	1.001	the health point of the hero
tower_hp_point	10.0	1.002	the health point of the turret
money	0.007	0.9998	money gained
exp	0.007	0.9998	experience points gained
ep_rate	0.75	1.0	the rate of mana
death	-1.2	1.0015	being killed
kill	-0.6	0.995	killing an enemy hero
last_hit	0.6	1.001	striking the last hit to an enemy unit
forward	0.05	1.001	moving forward
ult_hit	5.0	1.0015	hitting an enemy hero with ultimate skill

To enhance training stability and promote the development of complex behaviors, we further introduce a curriculum learning framework. This framework gradually increases the difficulty and specificity of the learning objectives across three stages:

Stage 1: Focuses on dense, continuous rewards that encourage fundamental gaming mechanics. Rewards for accumulating gold and experience guide the agent to farm efficiently by killing enemy minions.

Stage 2: Emphasizes intermediate objectives such as dealing damage to enemy heroes and towers, with increased reward weights assigned to these actions.

Stage 3: Shifts to sparse, high-level rewards that are more closely correlated with winning the game. For example, rewards are given for increasing the agent's skill hit rate and decreasing that of the enemy, promoting more strategic and effective combat behavior.

The curriculum progression is initially determined via manual evaluation, where human experts analyze agent rollouts against baseline models to assess whether the desired behaviors have emerged. Once proficiency is observed at the current stage, training advances to the next phase.

An overview of the complete training curriculum is illustrated in Figure 1.

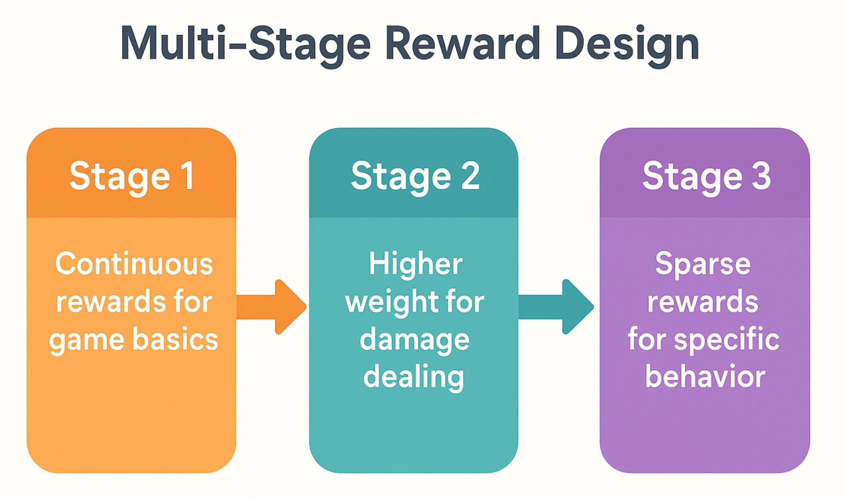
Multi-Stage Reward Design
Stage 1
Continuous rewards for game basics
Stage 2
Higher weight for damage dealing
Stage 3
Sparse rewards for specific behavior

Figure 1: Multi-stage Curriculum

4 EXPERIMENTS

In this section, we present two sets of experiments to evaluate the effectiveness of our proposed approach. First, we compare our best model against three baseline methods in a MOBA 1v1 environment, focusing on Win Rate and K/D (Kills over Deaths) as performance metrics. Subsequently, we conduct an ablation study to measure how key components——PPG + Advantage Normalization and Multi-stage Reward—contribute to final performance.

4.1 EXPERIMENTAL SETUP

4.1.1 BASELINES

All three baseline models were from Tencent's closed-source codebase and trained under the same environment configuration, differing primarily in the total training time and the number of interaction steps:

- Baseline-1: Trained for 3 hours, totaling 10,000 environment steps.
- Baseline-2: Trained for 5 hours, totaling 20,000 environment steps.
- Baseline-3: Trained for 8 hours, totaling 30,000 environment steps.

Although the detailed architectures and hyperparameters of these baselines remain unavailable, we include them here as comparative references to evaluate the performance gains achieved by our proposed approach.

4.1.2 METRICS

We adopt two metrics to evaluate agent performance in the HoK 1v1 environment:

- **Win Rate (%)**:
$$\text{Win Rate} = \frac{\text{Number of Wins}}{\text{Total Games Played}} \times 100\%$$
- **K/D**:
$$\text{K/D} = \frac{\sum \text{Kill Num}}{\sum \text{Death Num}}$$
Here, the total kills and deaths are summed over all matches against an opponent model.

For each pairwise comparison between our model and a baseline, we conducted a total of 50 matches.

4.2 EVALUATION

We evaluate our proposed method in two main ways: first, by comparing its performance against three strong baselines, and second, by performing an ablation study to validate the contribution of each algorithmic component.

4.2.1 COMPARISON WITH BASELINES

Table 2: Compared with Baseline Models

Model	Win Rate (%)	K/D
Baseline-1	90	3.62
Baseline-2	68	2.16
Baseline-3	12	0.75

Table 2 presents the results of our strongest model against the baselines. Our model achieves a 90% win rate against Baseline-1 and around 70% win rate against Baseline-2, confirming its robust combat capabilities. However, we only have a 10% win rate against Baseline-3.

Overall, the results indicate that our approach is well suited for the multifaceted challenges in 1v1 MOBA scenarios. The win rate improvements over Baseline-1 and Baseline-2 are particularly significant, suggesting that our method excels at hero-specific combat decisions and general strategic execution.

4.2.2 Ablation Study

Table 3: Ablation Study of PPG and Multi-stage Reward

PPG + Adv Norm	Multi-stage Reward	Win Rate (%)	K/D
✓	✓	90	3.62
✓	✗	68	3.64
✗	✓	36	0.85
✗	✗	8	0.44

To further assess the effectiveness of our approach, we conduct an ablation study by comparing multiple variants of our model against Baseline-1, as shown in Table 3. Specifically, we isolate two core components:

1. **PPG + Adv Norm**: This refers to the enhanced Phasic Policy Gradient (PPG) algorithm paired with advantage normalization.
2. **Multi-Stage Reward**: A progressive reward curriculum that shifts emphasis from basic tasks (e.g. last-hitting) to advanced objectives (e.g. skill using) between different training stages and withing a single game (using a time-varying reward).

These results confirm the critical roles of: (1) Architectural separation between policy/value optimization coupled with gradient stabilization techniques (2) Structured reward engineering that orchestrates exploration patterns through curriculum-based learning.

The synergistic combination of these elements enables our framework to attain both training stability and strategic sophistication, as evidenced by comprehensive baseline comparisons and component ablation analyses.

5 Conclusion

This study presents a multi-stage reinforcement learning framework specifically designed for 1v1 games in Honor of Kings. Our key insights include:

- **Dynamic Reward Adaptation**: A progress-sensitive reward system that automatically modulates incentive weights according to game phase transitions, ensuring alignment between tactical priorities and learning objectives across different stages.
- **Progressive Reward Curriculum**: A three-phase training architecture that systematically shifts focus from frequent basic rewards (e.g., resource collection) to sparse strategic incentives (e.g., combat precision), enabling balanced skill acquisition and complex behavior emergence.
- **Algorithm-Reward Synergy**: Empirical validation of the critical interdependence between reward engineering and algorithmic capabilities, where neither component alone guarantees optimal performance.

Experimental results demonstrate our agent achieves 90% victory rates against novice opponents and 70% against intermediate adversaries, exhibiting refined behaviors including resource optimization, combat disengagement, and high skill accuracy. However, performance declines to 10% win rates when confronting expert-level strategies.

Key findings reveal:

- Sparse rewards require precise calibration and progressive introduction to avoid undermining learning through inadequate feedback
- Curriculum structuring is indispensable for managing multi-objective reward complexity and enabling hierarchical skill development
- Reward effectiveness fundamentally depends on the learning algorithm's capacity to interpret and optimize complex incentive signals

Current limitations include suboptimal decision-making under disadvantage conditions (e.g., excessive focus on survival over tower defense) and absence of expert-level tactics like predictive skill evasion. Future directions involve implementing ability-specific reward shaping for advanced strategies and integrating hybrid algorithms to enhance learning robustness.

REFERENCES

Karl W Cobbe, Jacob Hilton, Oleg Klimov, and John Schulman. Phasic policy gradient. In *International Conference on Machine Learning*, pp. 2020–2027. PMLR, 2021.

Yun Qu, Boyuan Wang, Jianzhun Shao, Yuhang Jiang, Chen Chen, Zhenbin Ye, Liu Linc, Yang Feng, Lin Lai, Hongyang Qin, Minwen Deng, Juchao Zhuo, Deheng Ye, Qiang Fu, YANG GUANG, Wei Yang, Lanxiao Huang, and Xiangyang Ji. Hokoff: Real game dataset from honor of kings and its offline reinforcement learning benchmarks. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 22166–22190. Curran Associates, Inc., 2023. URL `https://proceedings.neurips.cc/paper_files/paper/2023/file/464fefa022aaefc85d901317bbf13f85-Paper-Datasets_and_Benchmarks.pdf`.

Hua Wei, Jingxiao Chen, Xiyang Ji, Hongyang Qin, Minwen Deng, Siqin Li, Liang Wang, Weinan Zhang, Yong Yu, Liu Linc, Lanxiao Huang, Deheng Ye, Qiang Fu, and Wei Yang. Honor of kings arena: an environment for generalization in competitive reinforcement learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 11881–11892. Curran Associates, Inc., 2022. URL `https://proceedings.neurips.cc/paper_files/paper/2022/file/4dbb61cb68671edc4ca3712d70083b9f-Paper-Datasets_and_Benchmarks.pdf`.

Deheng Ye, Guibin Chen, Wen Zhang, Sheng Chen, Bo Yuan, Bo Liu, Jia Chen, Zhao Liu, Fuhao Qiu, Hongsheng Yu, Yinyuting Yin, Bei Shi, Liang Wang, Tengfei Shi, Qiang Fu, Wei Yang, Lanxiao Huang, and Wei Liu. Towards playing full moba games with deep reinforcement learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 621–632. Curran Associates, Inc., 2020a. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/06d5ae105ea1bea4d800bc96491876e9-Paper.pdf`.

Deheng Ye, Zhao Liu, Mingfei Sun, Bei Shi, Peilin Zhao, Hao Wu, Hongsheng Yu, Shaojie Yang, Xipeng Wu, Qingwei Guo, et al. Mastering complex control in moba games with deep reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 6672–6679, 2020b.

Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and YI WU. The surprising effectiveness of ppo in cooperative multi-agent games. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 24611–24624. Curran Associates, Inc., 2022. URL `https://proceedings.neurips.cc/paper_files/paper/2022/file/9c1535a02f0ce079433344e14d910597-Paper-Datasets_and_Benchmarks.pdf`.