GDC | March 21-25, 2022
San Francisco, CA

Multi-Agent Reinforcement Learning Invades MMORPG: Lineage Clone Wars

NCSOFT AI Center
Game AI Lab
Inseok Oh

NCSOFT AI Center
Game AI Lab
Jinhyung Ahn

#GDC22

Agenda

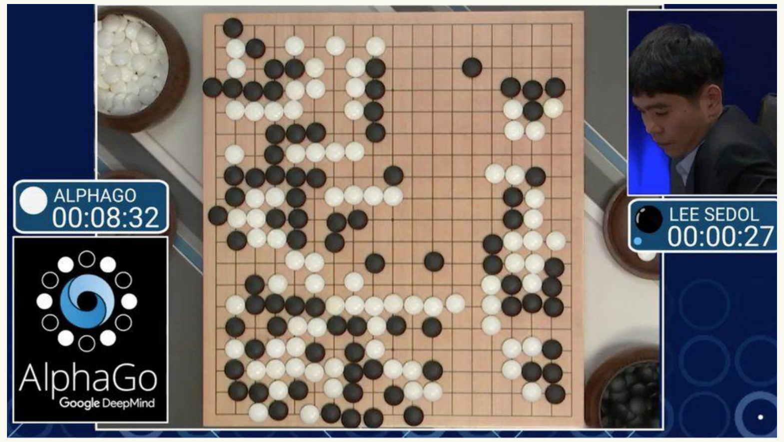
- Our Goal & Method
- Part I. Game Introduction
- Part II. Reinforcement Learning Method
- Part III. Reinforcement Learning Framework
- Conclusion

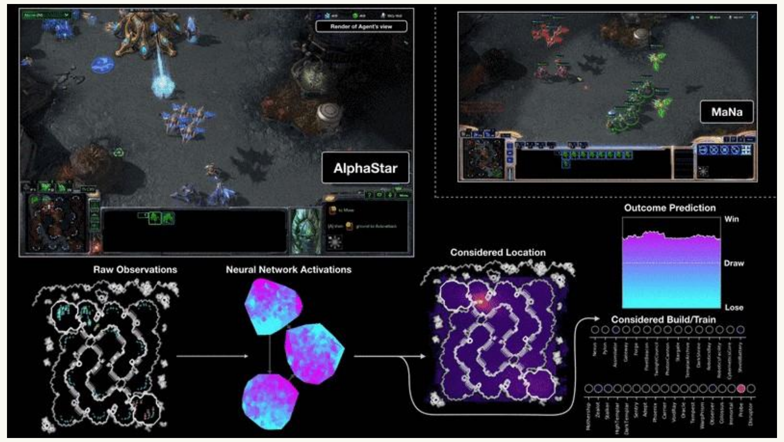
Our Goal

- Develop AIs that can fight against human users in large-scale battle
    - New gaming experience
    - Different kind of opponent

Our Method

- Reinforcement Learning
- Why reinforcement learning?
  - Human behavior is too complex for traditional methods (e.g., FSM, behavior tree)
  - Constant updates in MMORPG → easily update AIs with RL





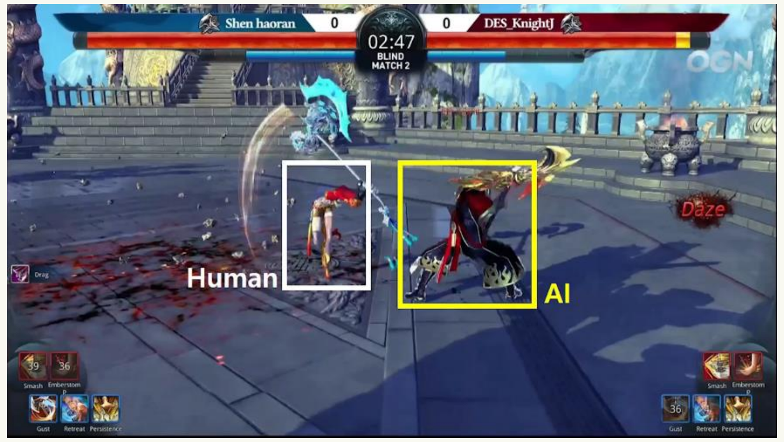
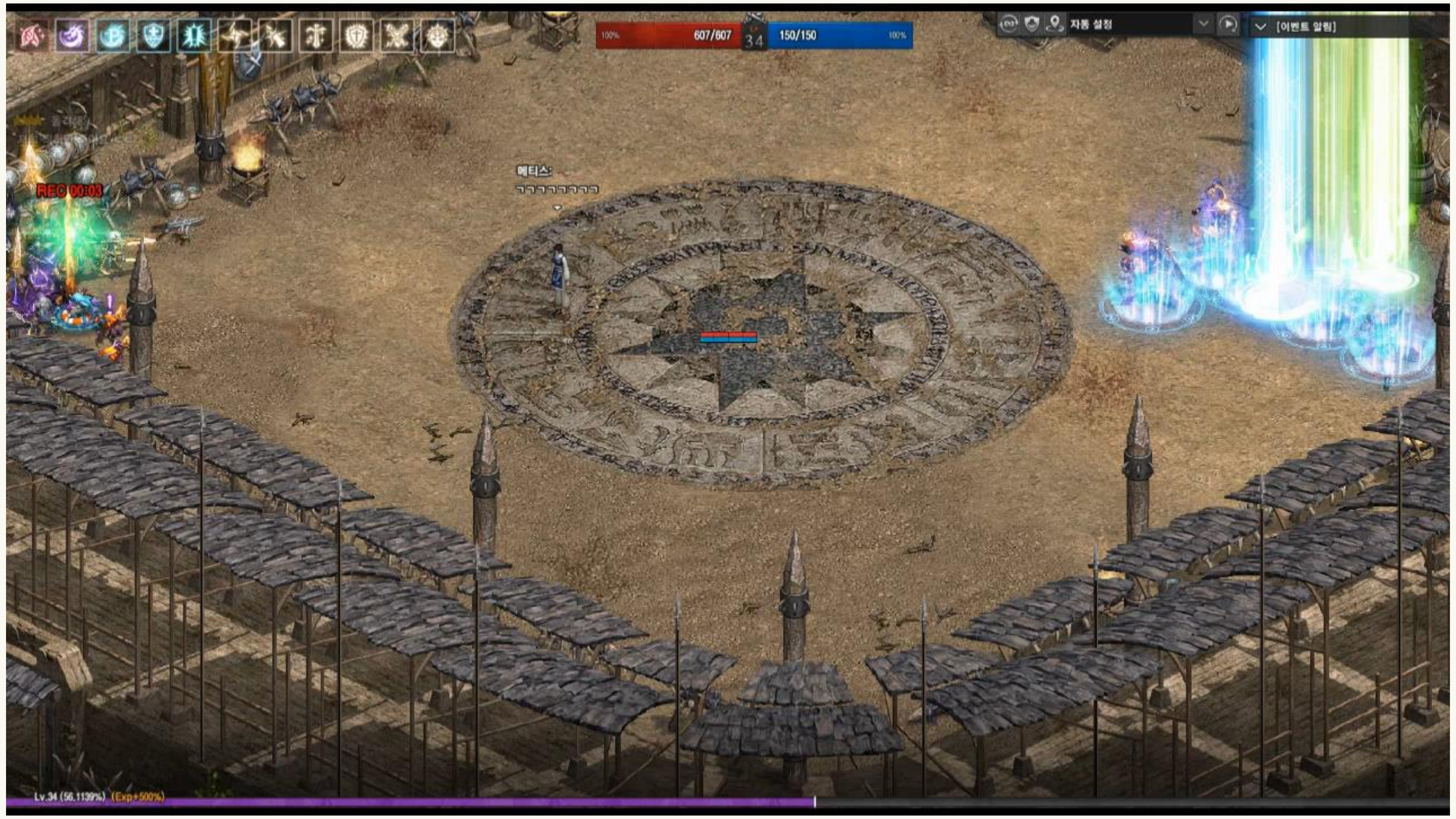


image ref :https://www.bbc.com/news/technology-35785875
https://deepmind.com/blog/article/alphastar-mastering-real-time-strategy-game-starcraft-ii
https://towardsdatascience.com/mastering-deep-reinforcement-learning-with-openais-new-spinning-up-in-deep-rl-package-b86b61ab6e54

Part I. Game Introduction

AI Content 1. Legend Reborn

- **Our goal:** create AIs that can win human players in 8 vs. 8 battle
- **Three key points**
  - Focus fire on specific targets
  - Position themselves depending on character class
  - Select skills appropriate in each situation

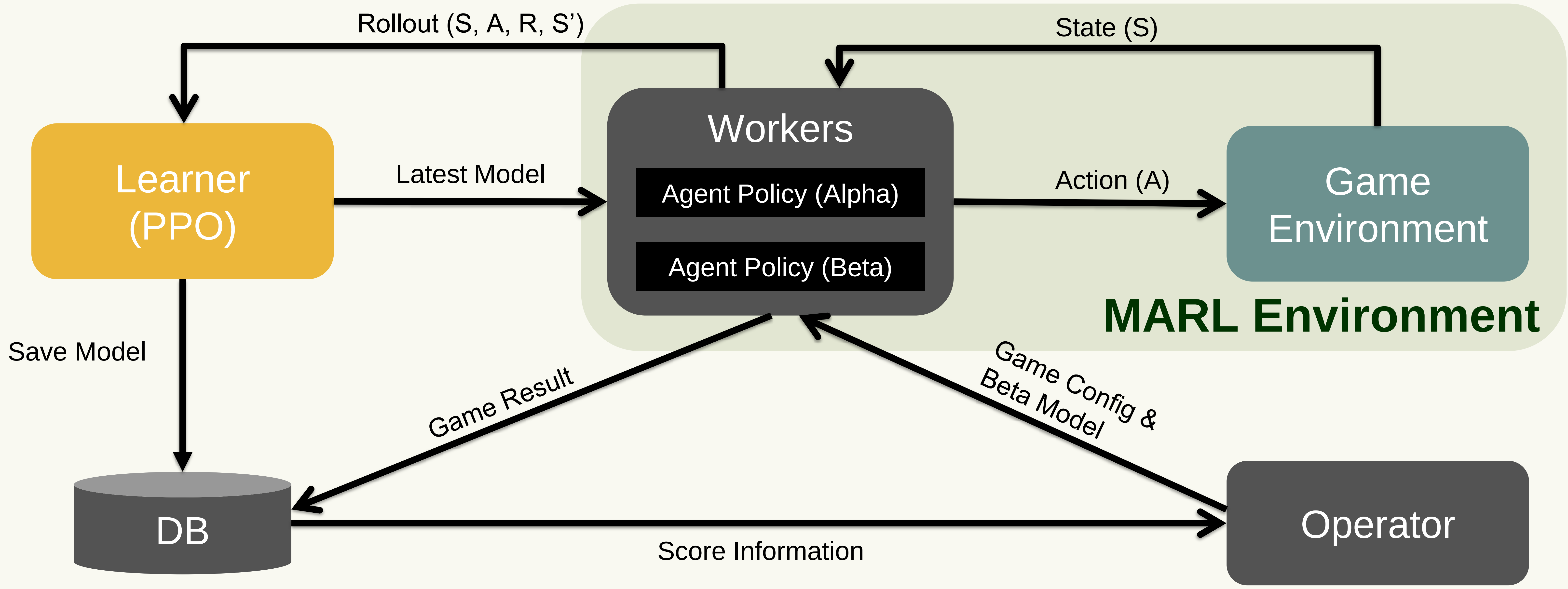


AI Content 2. Clone Wars

- **Our goal:** create AIs that behave like humans but provide more battle experience
    - Attack users
    - Hunt monsters
    - Raid bosses

Part II. Reinforcement Learning Method

Overview of Our Learning Framework

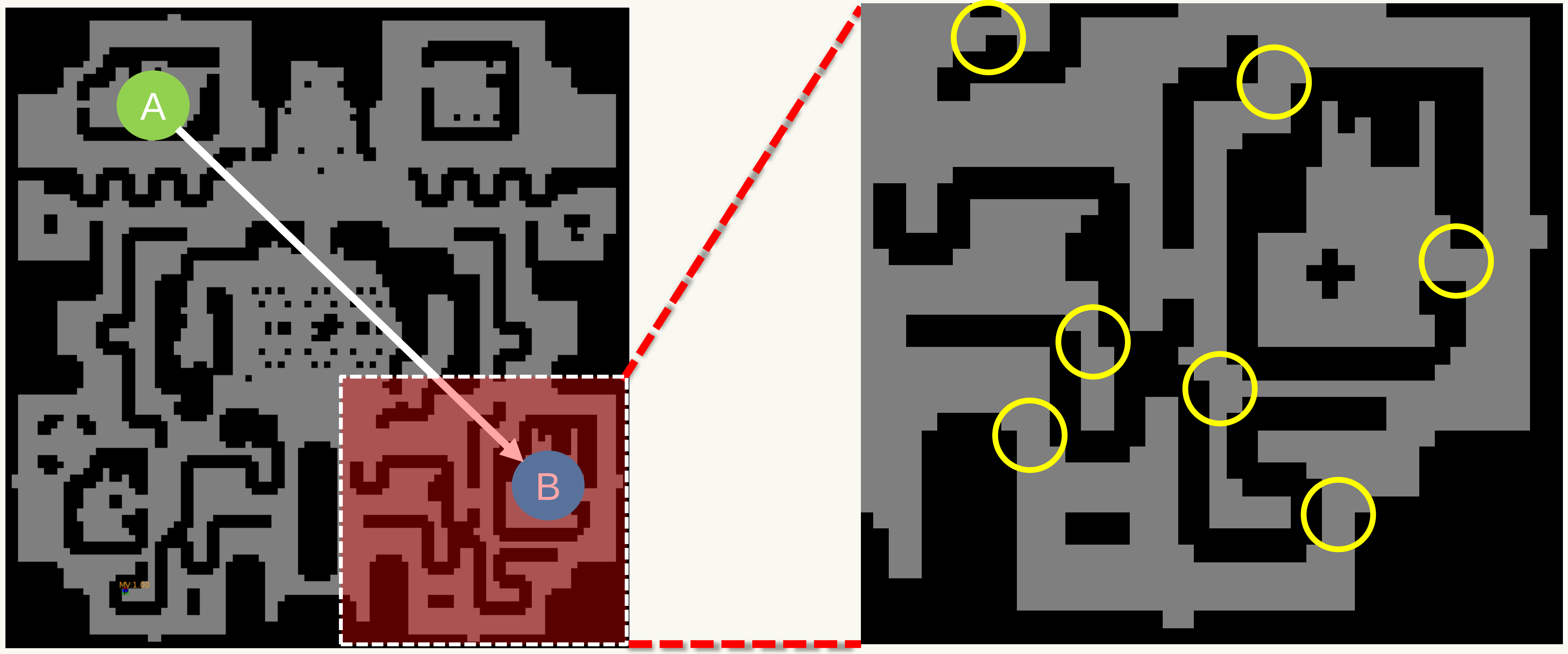


Practical Constraints

- **Limited training resources**
  - About 300 CPUs
- **Limited training time**
  - 1 week (before live service)
- **Constant updates**
  - Live service game
  - Additional training needed

Challenge 1: Geographic Complexity

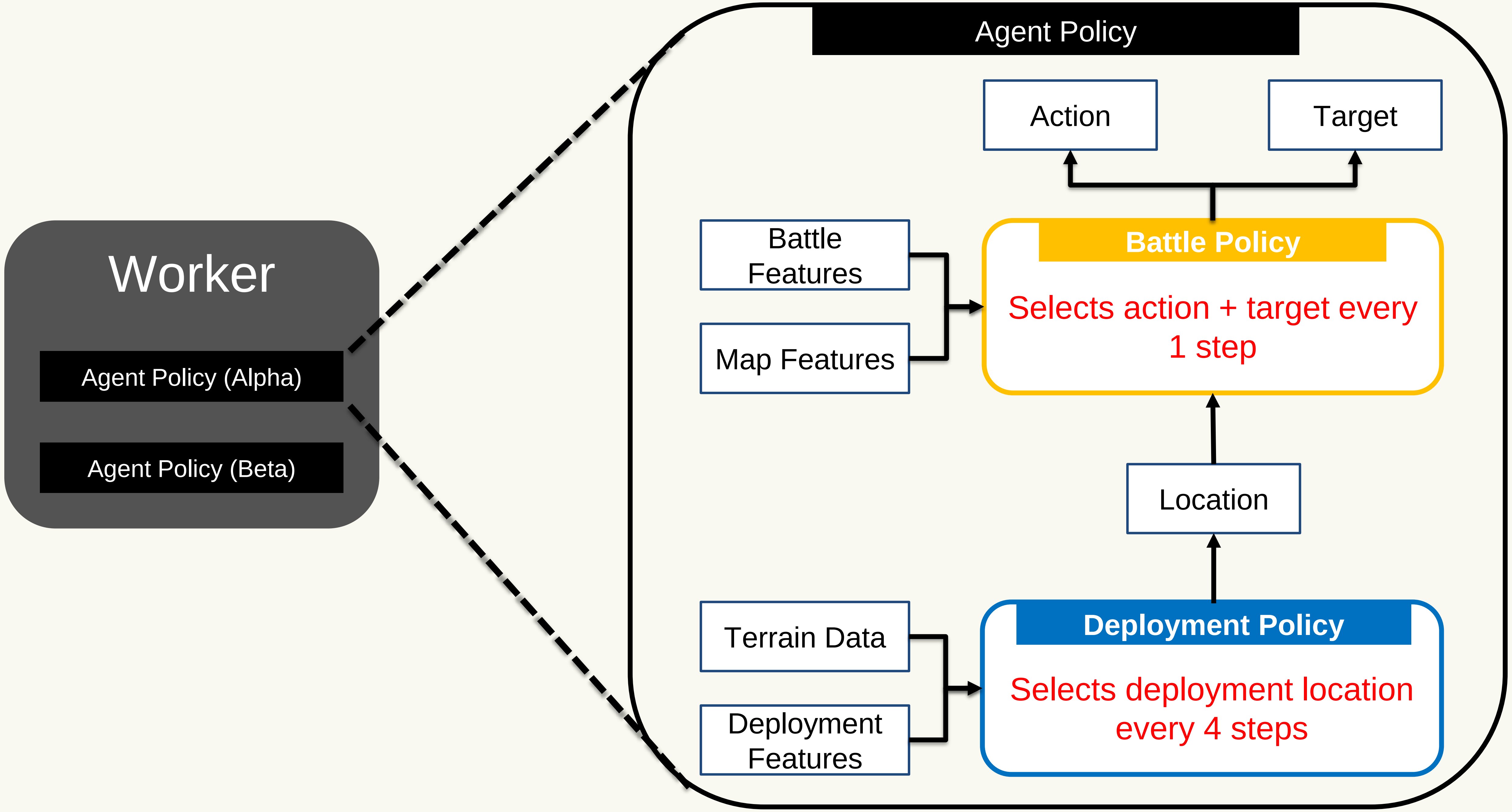
- Large map size + complex terrain



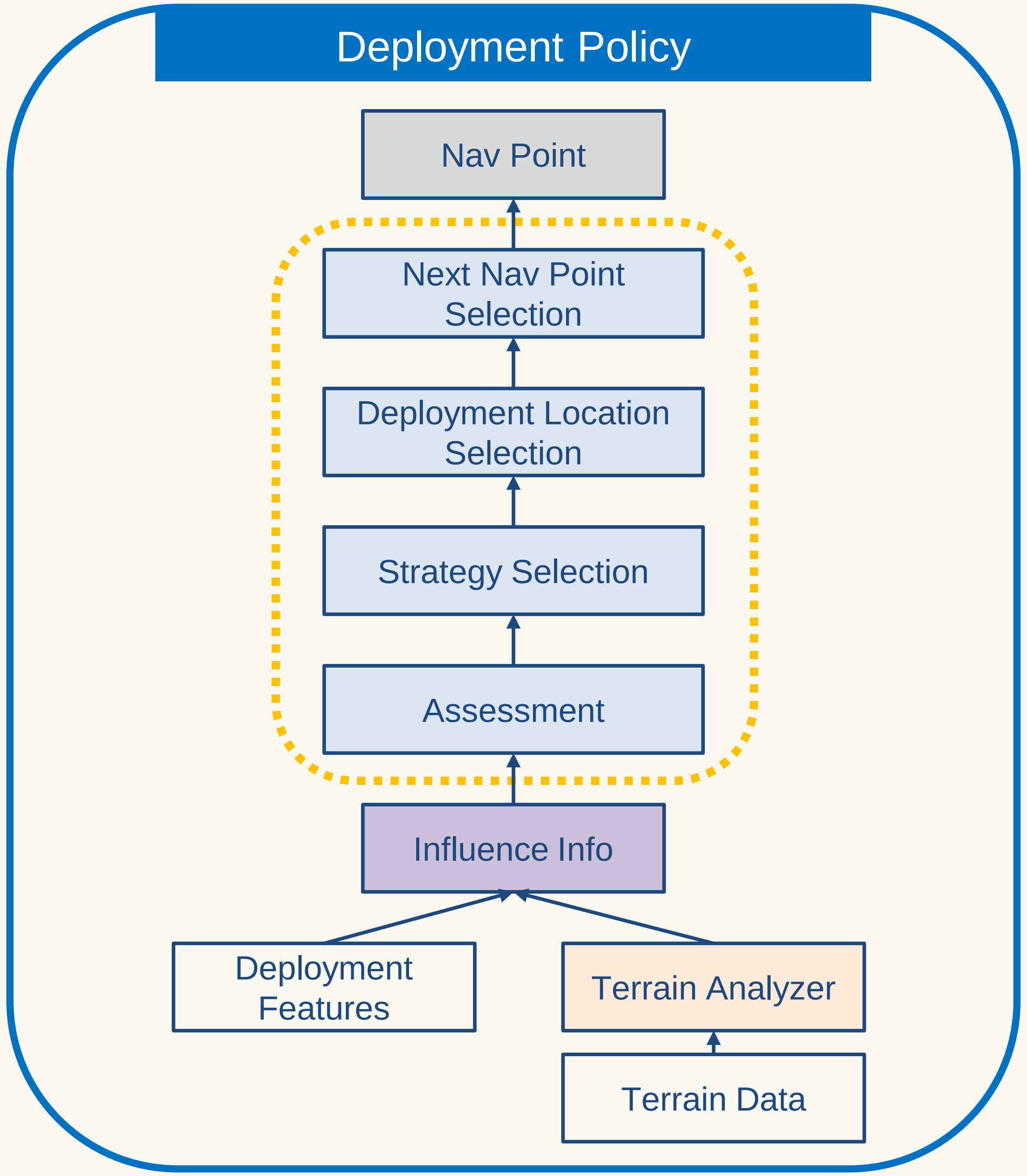
Time spent moving from
A → B: over 100 steps

Multiple narrow entryways
(Complex terrain)

Hierarchical Decision Structure

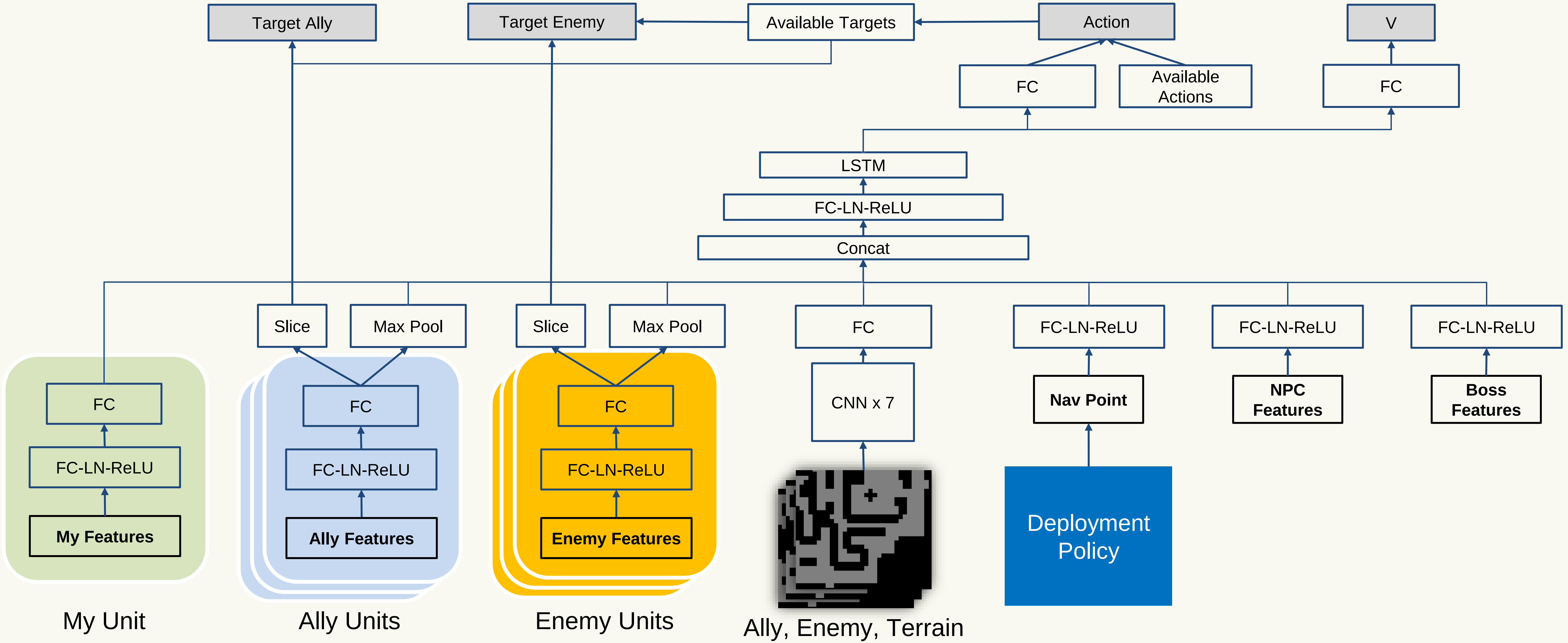


Deployment Policy

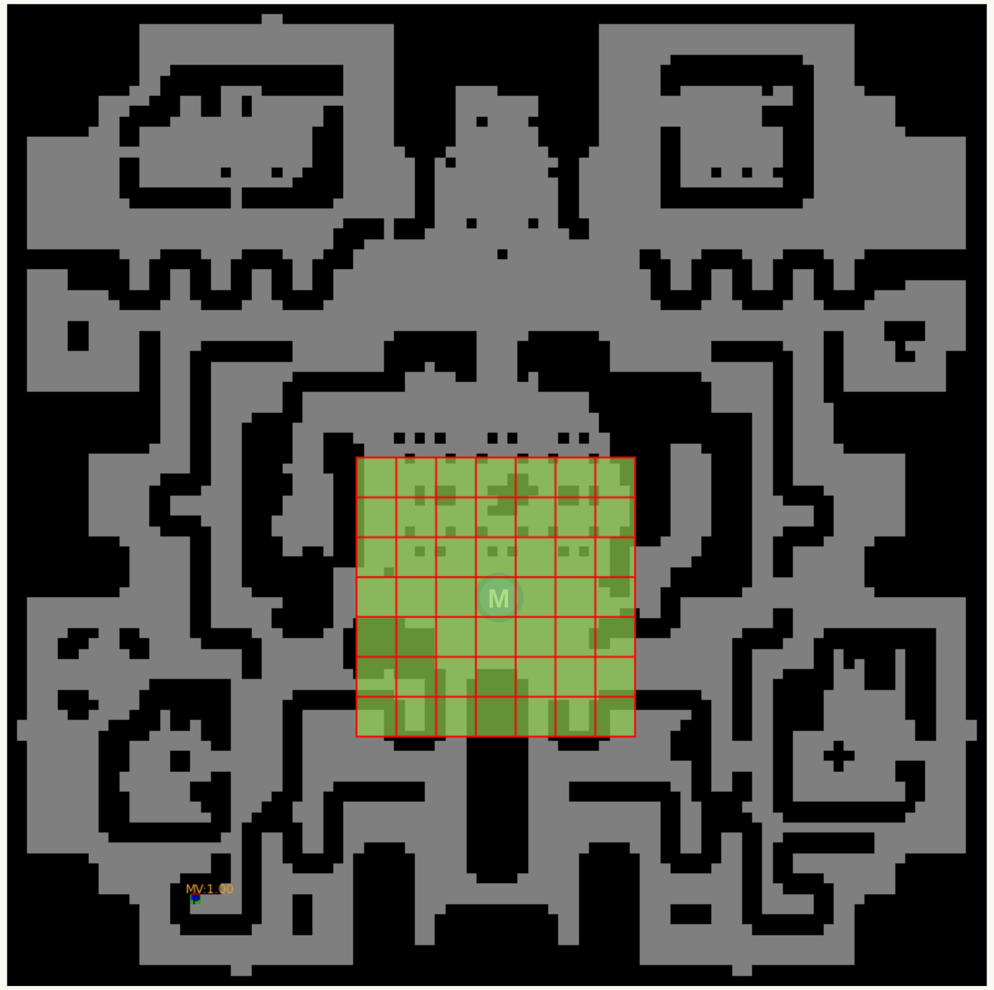


- Used to move agents to distant locations on a large map
  1. **[Influence Info]** calculates the influence of each team
  2. **[Assessment]** decides which team is dominant
  3. **[Strategy Selection]** selects strategy among offense, sweep, and defense
  4. **[Deployment Location Selection]** decides deployment location
  5. **[Next Nav Point Selection]** selects navigation points and feeds to battle policy

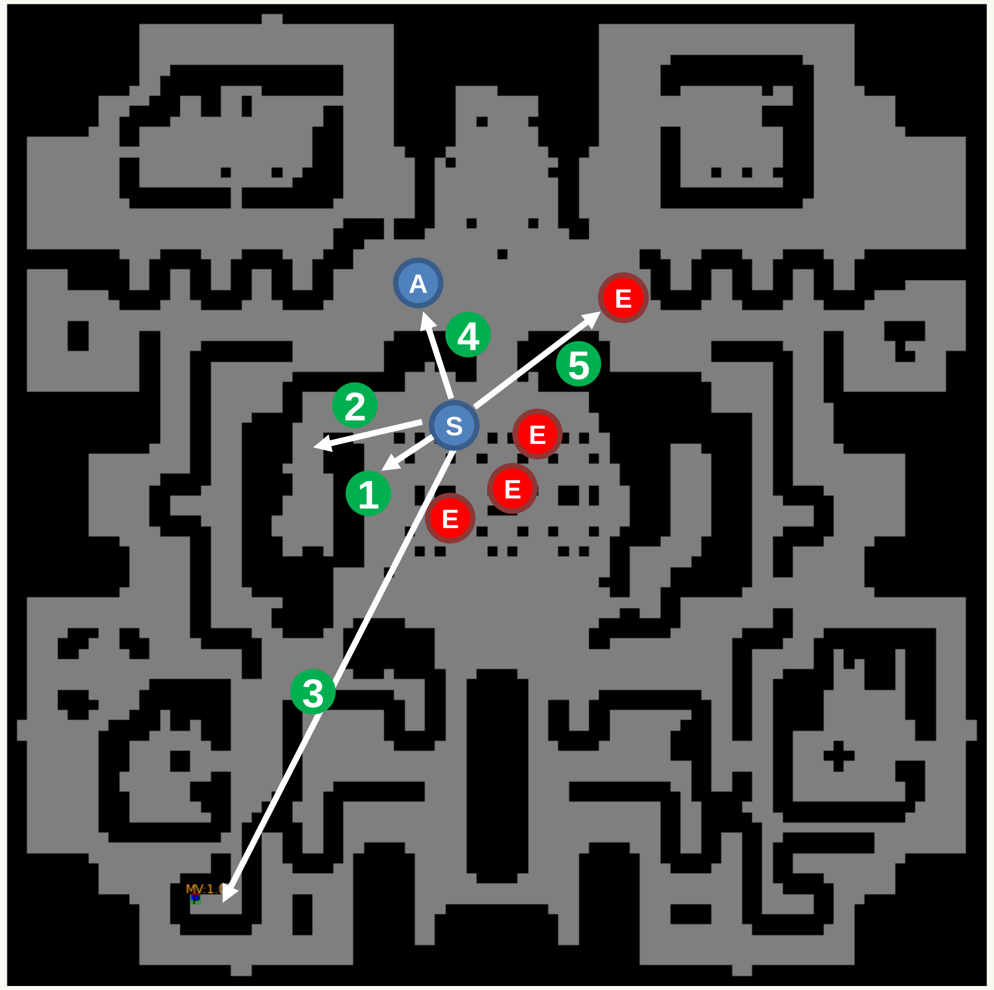
Battle Policy



Move Space Reduction



7x7 move points
[AI remains untrained]



Conceptualized move actions
[AI trained within 1 week]

Agent

- S Selected AI
- A Ally (AI)
- E Enemy (user)

Move Action

1. Hide behind wall
2. Retreat to close area
3. Retreat to distant area
4. Move to ally
5. Move to enemy

Challenge 2: Innumerable Combinations

- Numerous combinations of characters with different specifications

Number of participating characters: 8-16

Number of character classes: 10

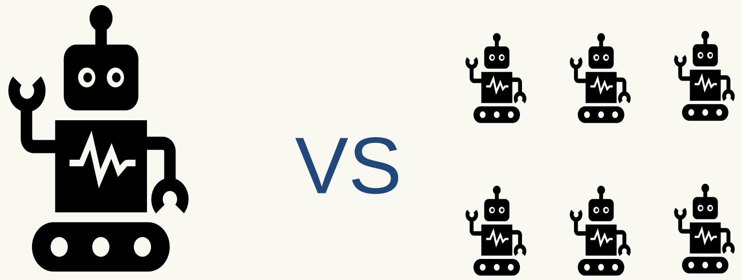
Range of character levels: 91-99

Types of character stats: 100+

Ex) STR, DEX, INT, CON, WIS, AGI, AC, etc.

Innumerable possible combinations

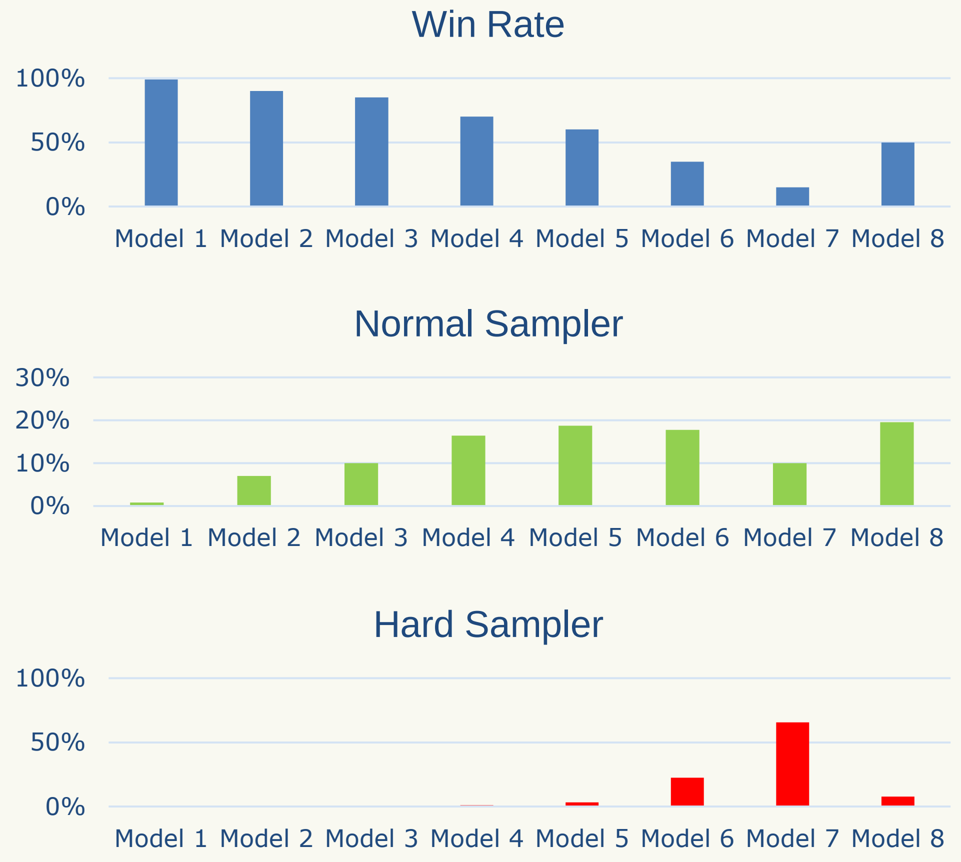
Sampling Methods for Self-Play Learning



Current Model

Past Model

Self-Play Learning



Model Sampling

- Random sampling → class combinations not used in real battles
- Combination sampling → combinations frequently used by users
  - At least 1 magician
  - At least X% melee class
  - 1-16 units per team
  - 30%↓ difference between number of ally and enemy units

Combination Sampling

Challenge 3: Behavior Shaping

- **Achieve human-likeness**
  - Example 1. AIs must occasionally prioritize their own life over allies
  - Example 2. AIs must stay close to their allies when preparing for battle
- **Make the game content fun for users**
  - Example 1. Fight to the death even when there is no chance of winning
  - Example 2. Behave aggressively → start more battles!

Reward Design for Behavior Shaping

- Achieve human-like behavior (e.g., prioritize their own life over allies')
  - -1.2 team reward if ally dies
  - Additional -0.6 individual reward if AI itself dies
- Encourage aggressive behavior
  - Penalty for time passed
- **Reward shaping: simple & effective way to satisfy the intentions of game designers**

Total Reward

Individual Reward

- -0.6 if I die
- -0.02 if no action
- 0.015 if ally is near
- 0.1 if I use buff
- 0.02 if I use debuff

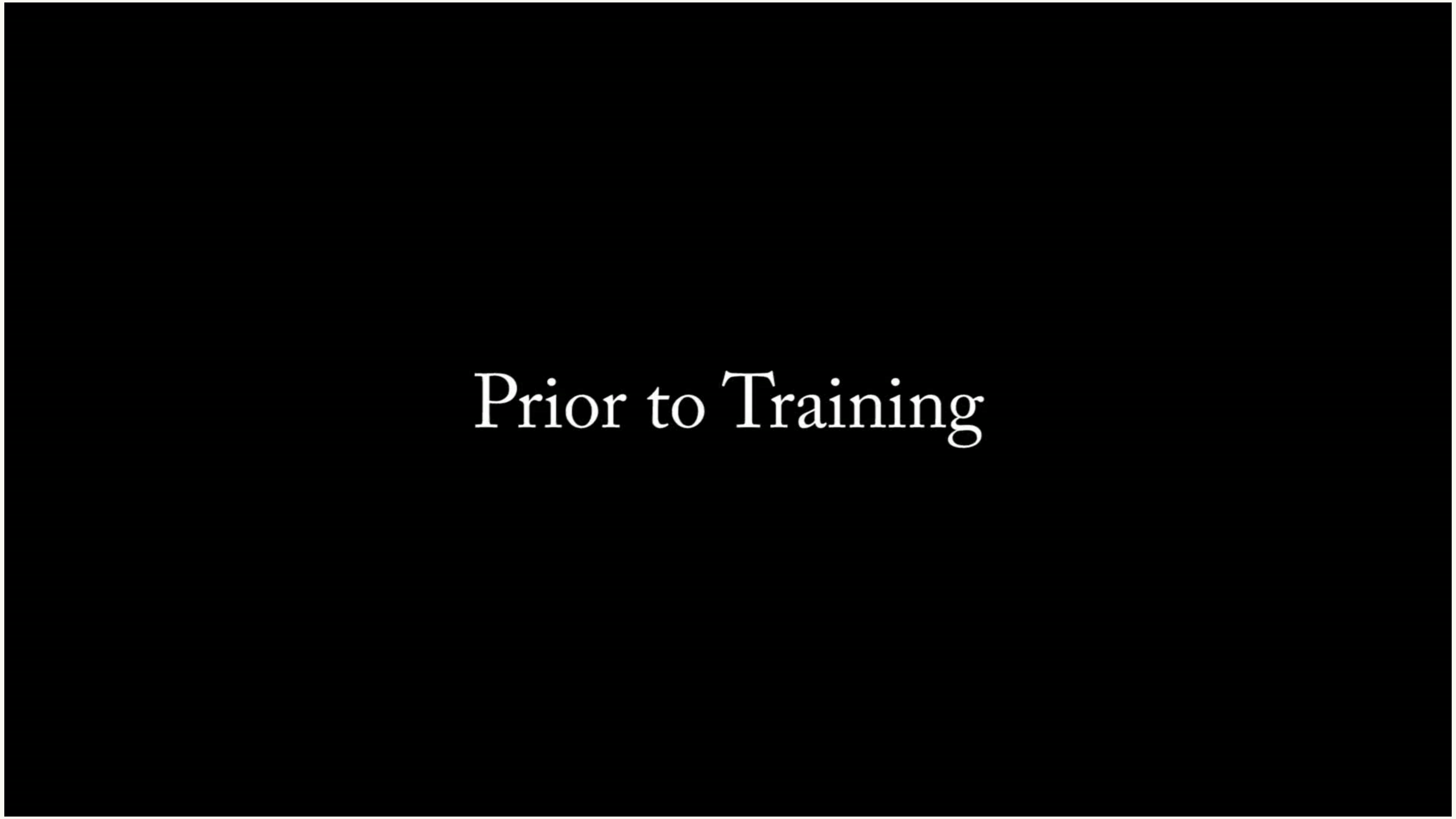
Team Reward

- -5×10^{-3} per tick
- -1.2 per dead ally
- 1.2 per dead enemy
- 0.08 per dead NPC
- 2.0 per dead boss

Other Training Methods We Used

- **Curriculum learning**
  - Make allies and enemies spawn close to each other X% of the time
- **Decay entropy loss coefficient**
  - Experience variety of actions → Increase probability of choosing better actions
- **Network surgery**
  - Respond to live service updates
  - Action-wise entropy loss → help AIs experience added skills

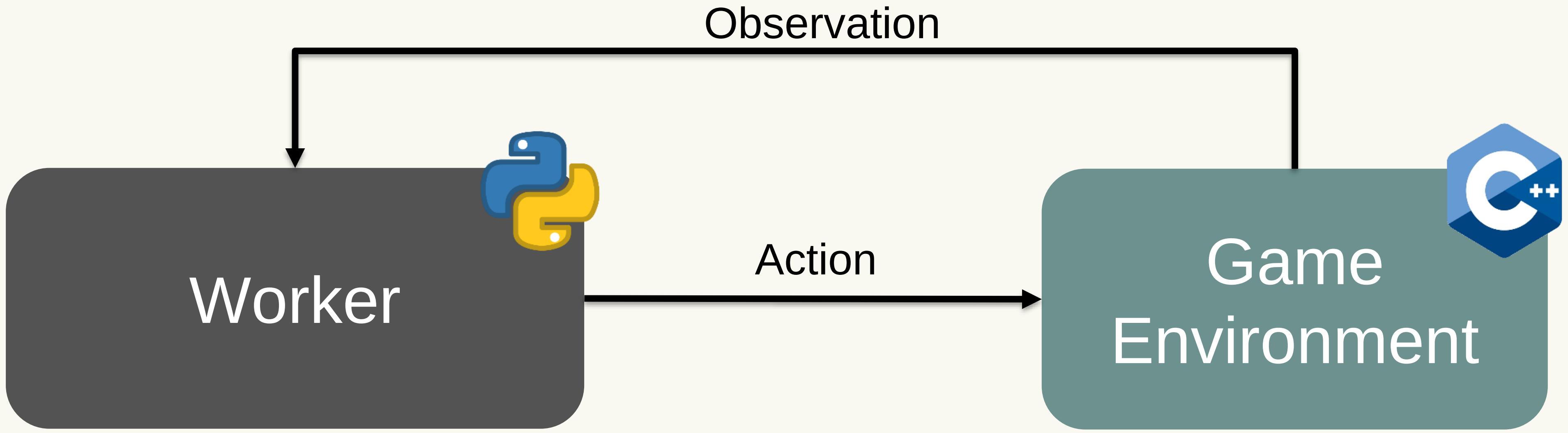
Video 1. Legend Reborn



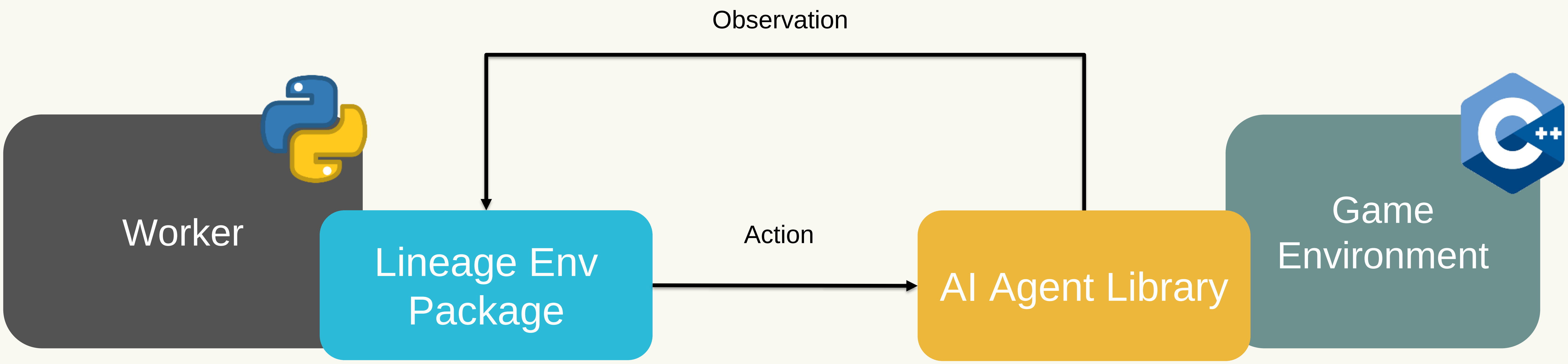
Video 2. Clone Wars

Part III. Reinforcement Learning Framework

Structure of Simulator



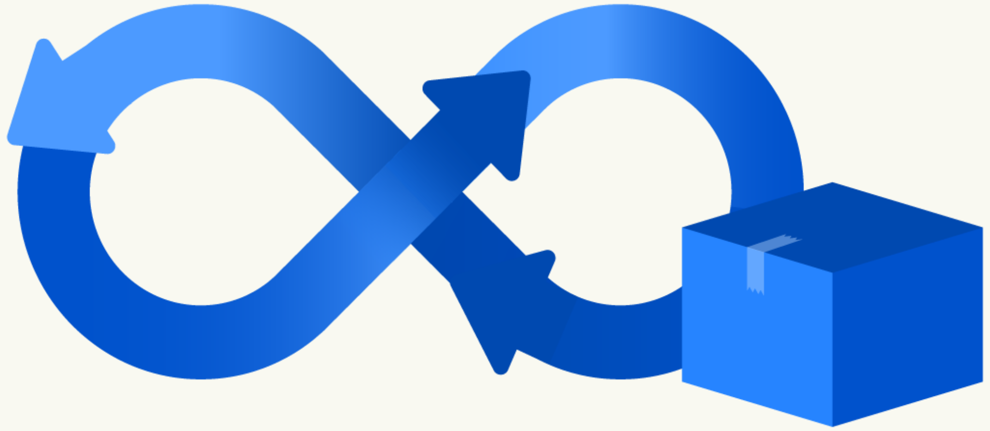
Structure of Simulator



RL Contents Development Pipeline

Challenge 1: Constant Game Updates

- **Reducing Simulator Development Cost**
  - Online game environments are constantly being updated
  - These updates affect the learning process
  - Without prompt support, the AI reveals weaknesses (e.g., AI fails to learn new skills)



110 environment updates in 20 days

Simulator Updating Process

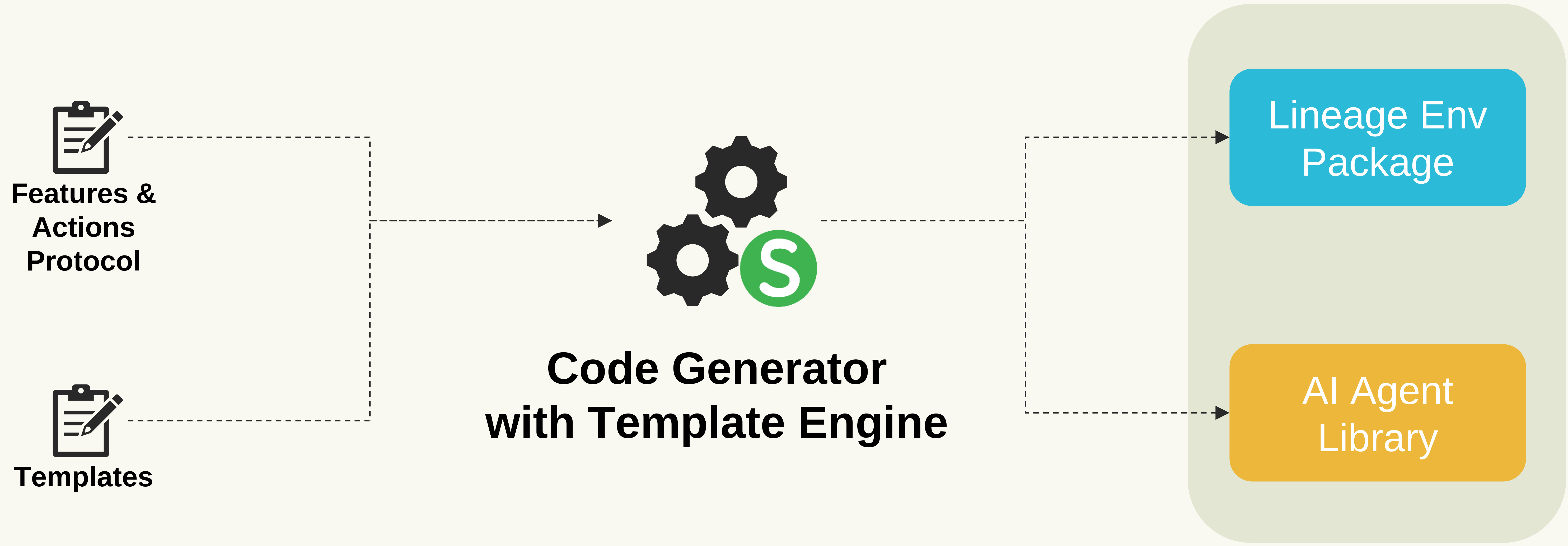
Check Requirements → Update Lineage Env Package → Update AI Agent Library → Distribute Simulator

Approx. 1 hour

Check Requirements → Generate (Env Package and AI Agent Library) → Distribute Simulator

Up to 10 min

The Automatic Code Generating System

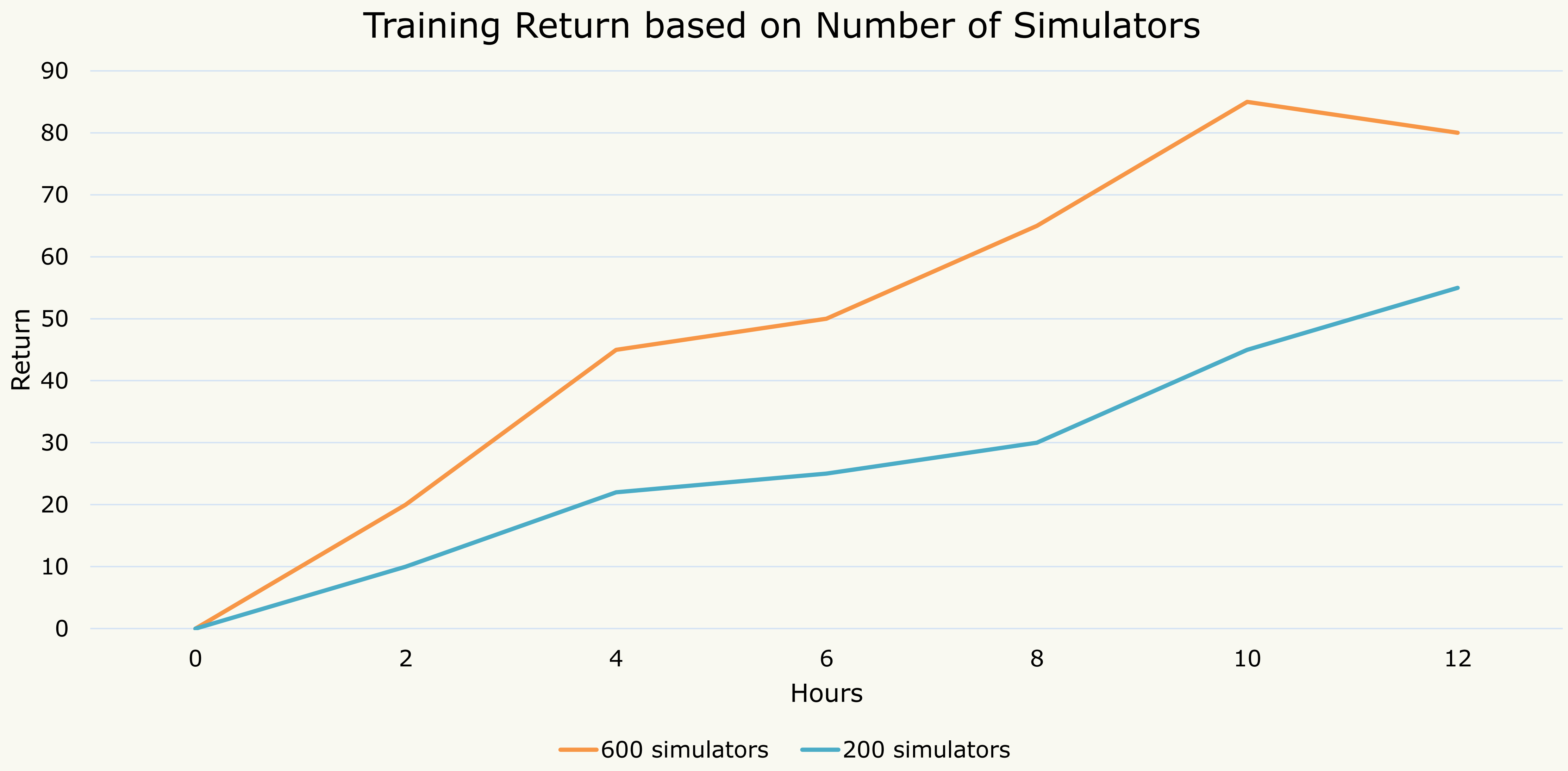


Automatic Code Generation

- **Results**
  - Observations and actions
  - Feature management
  - Simulation control API

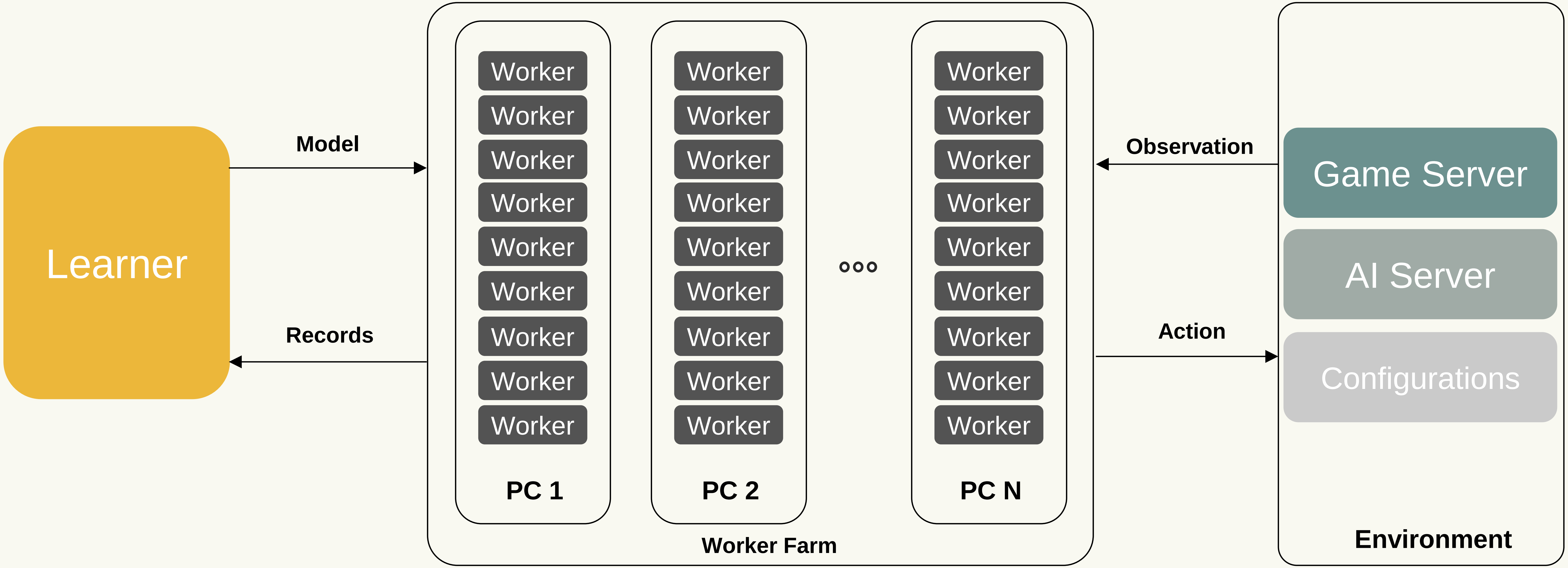
- **Effects**
  - Saving Coding Time
  - Eliminates mistakes
  - Improve productivity

Challenge 2: Maximizing Number of Simulations

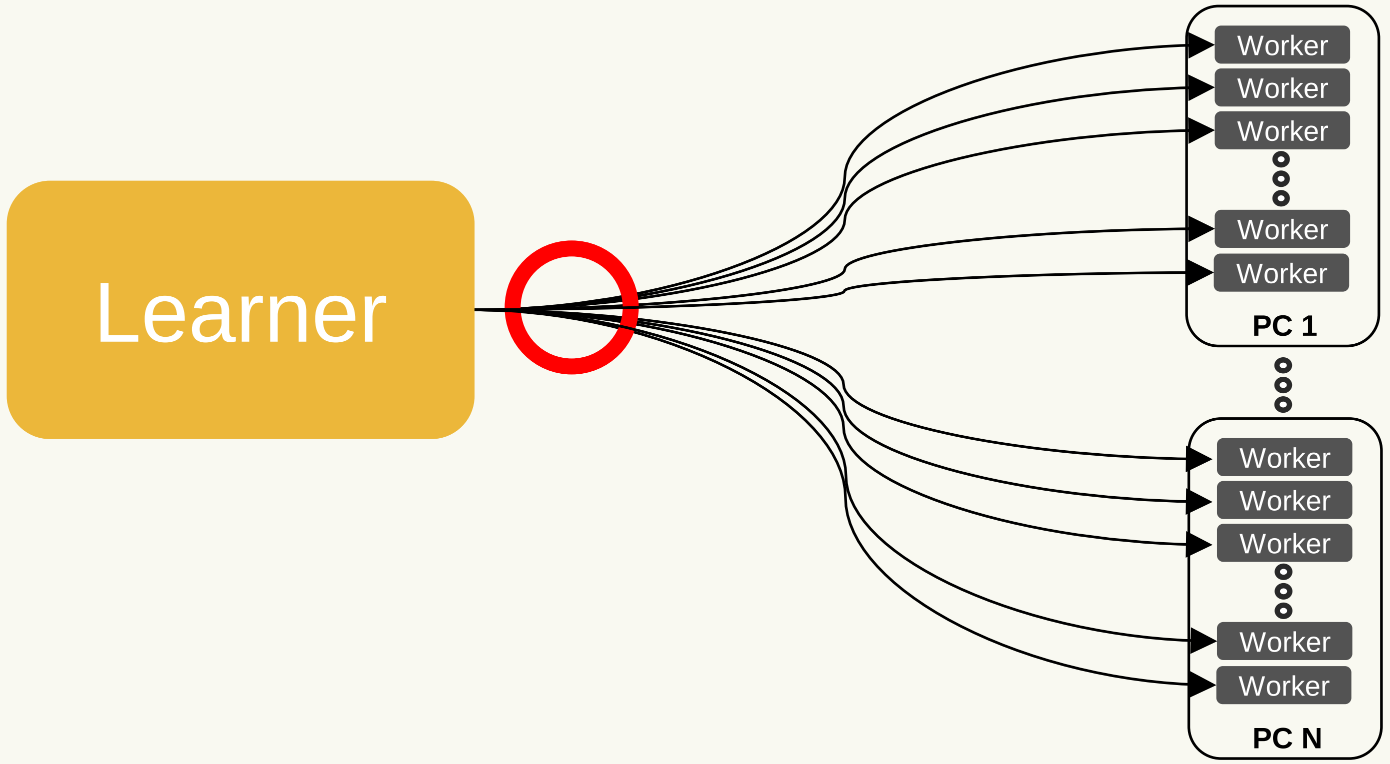


- DRL model learns through data
- More simulations and data required

Training System

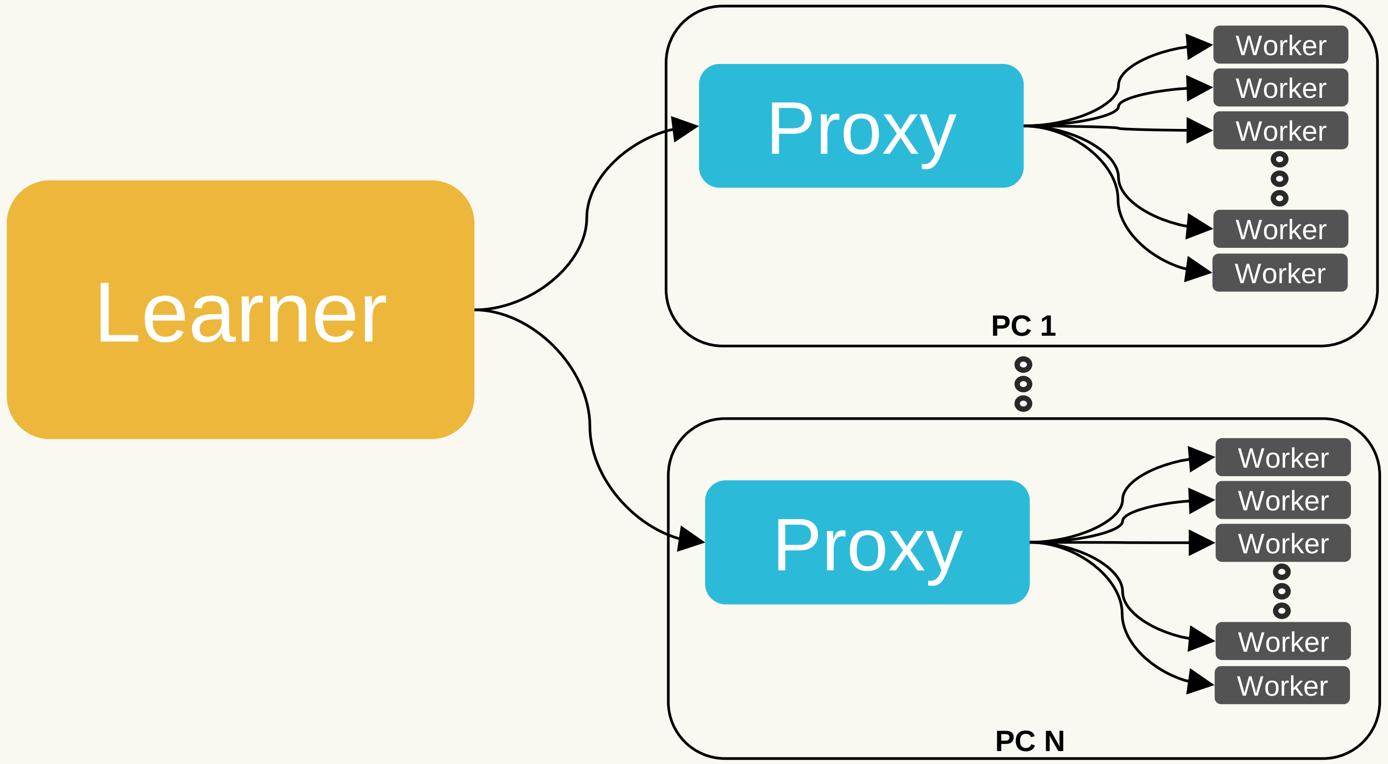


Reducing Network Traffic



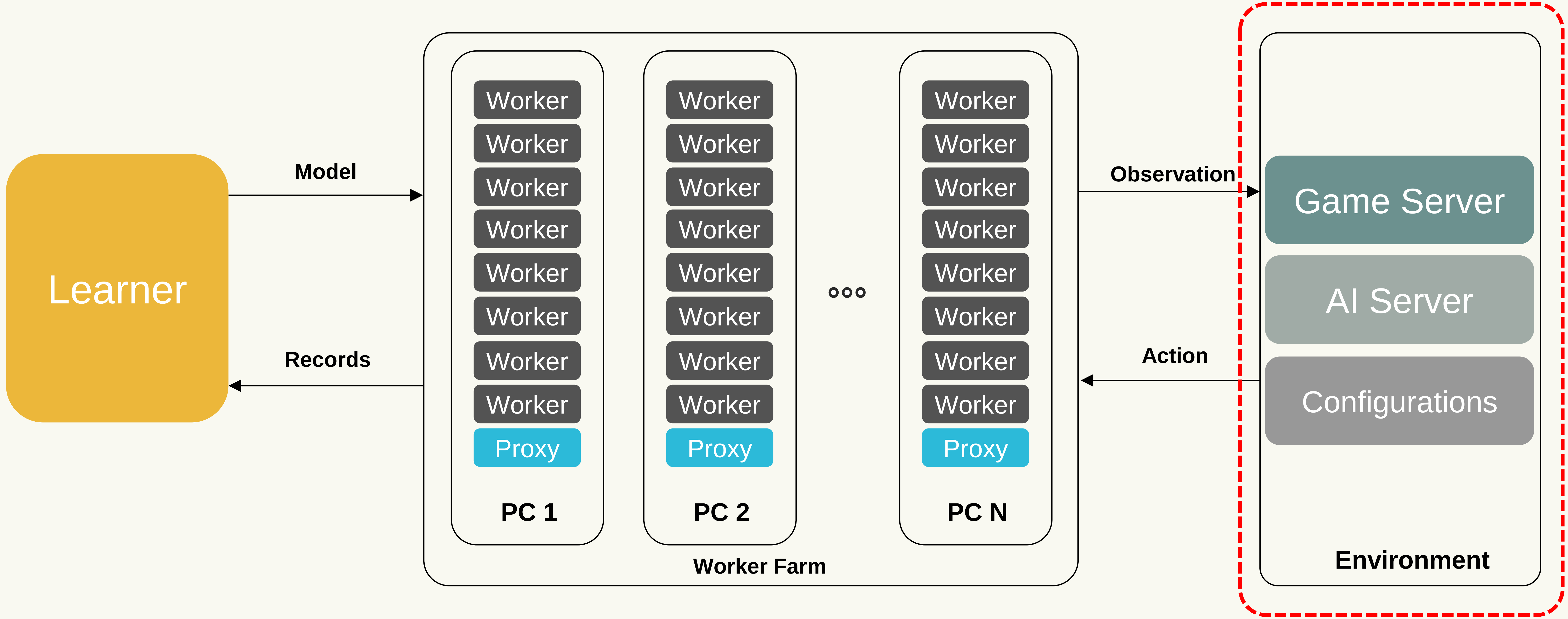
- Learner sends a model to each worker every 5 seconds
- Increased number of workers leads to amount of traffic beyond network capacity

Reducing Network Traffic

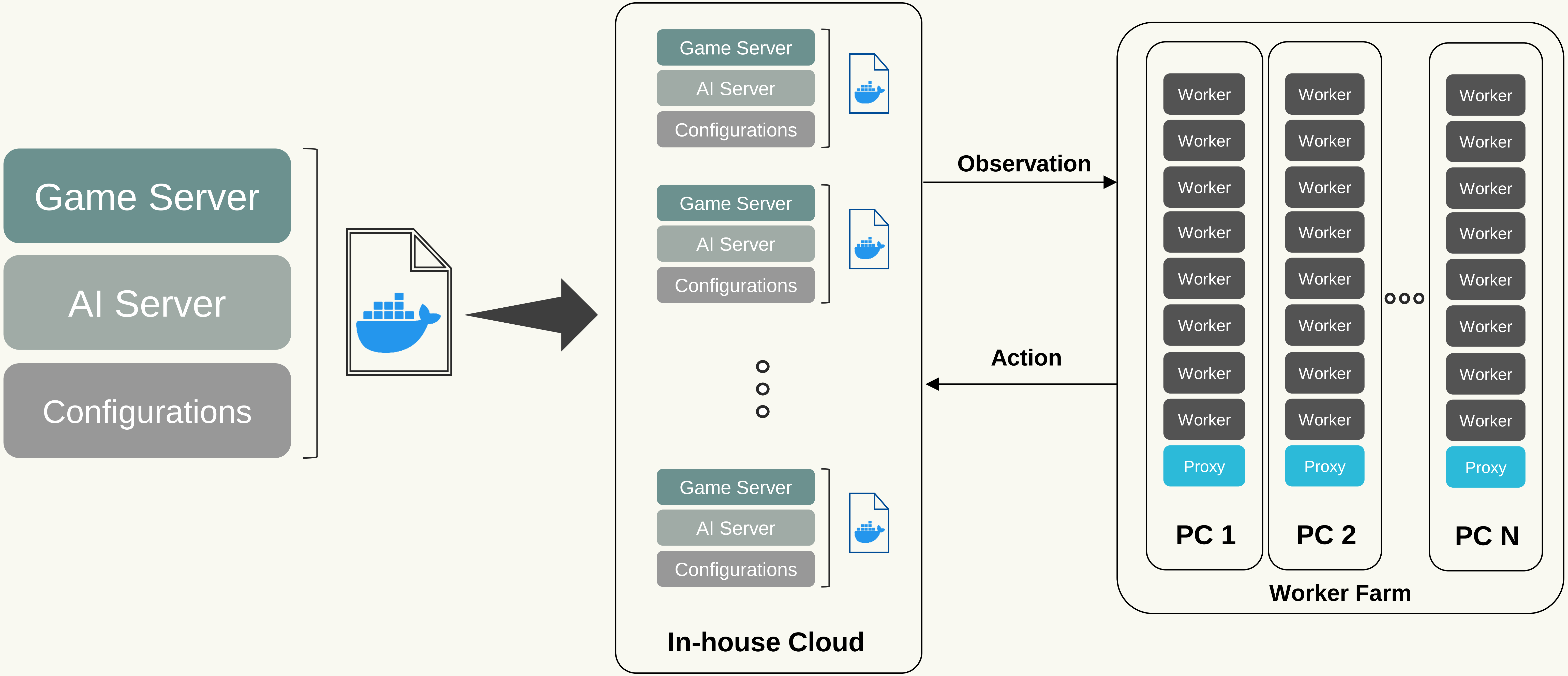


- Sending models only to the proxy process of each PC greatly reduces network traffic
- Proxy process publishes models to local workers

Limited Performance from Using a Single Server



Server Expansion



Challenge 3: Detecting Abnormalities

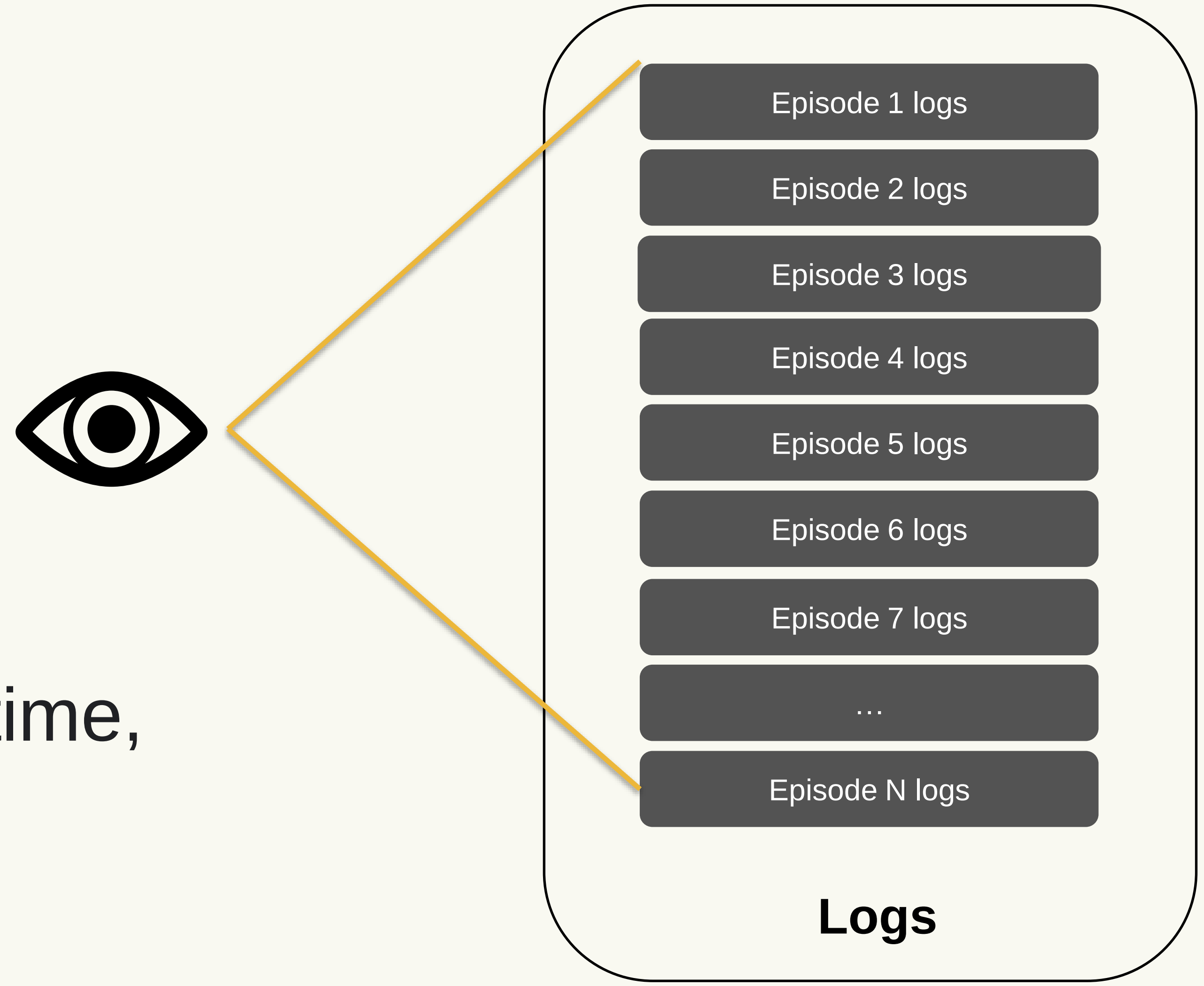
- Performing prompt analyses and responses to abnormal AI behavior
  - Too many games to analyze
  - Over 1,000 workers required to run 24/7

Saving Information to Detect Abnormalities

- Information that needs to be saved: features and packets at every step
- Process information used in the Lineage env package such as functions and parameters are already being logged

Watcher

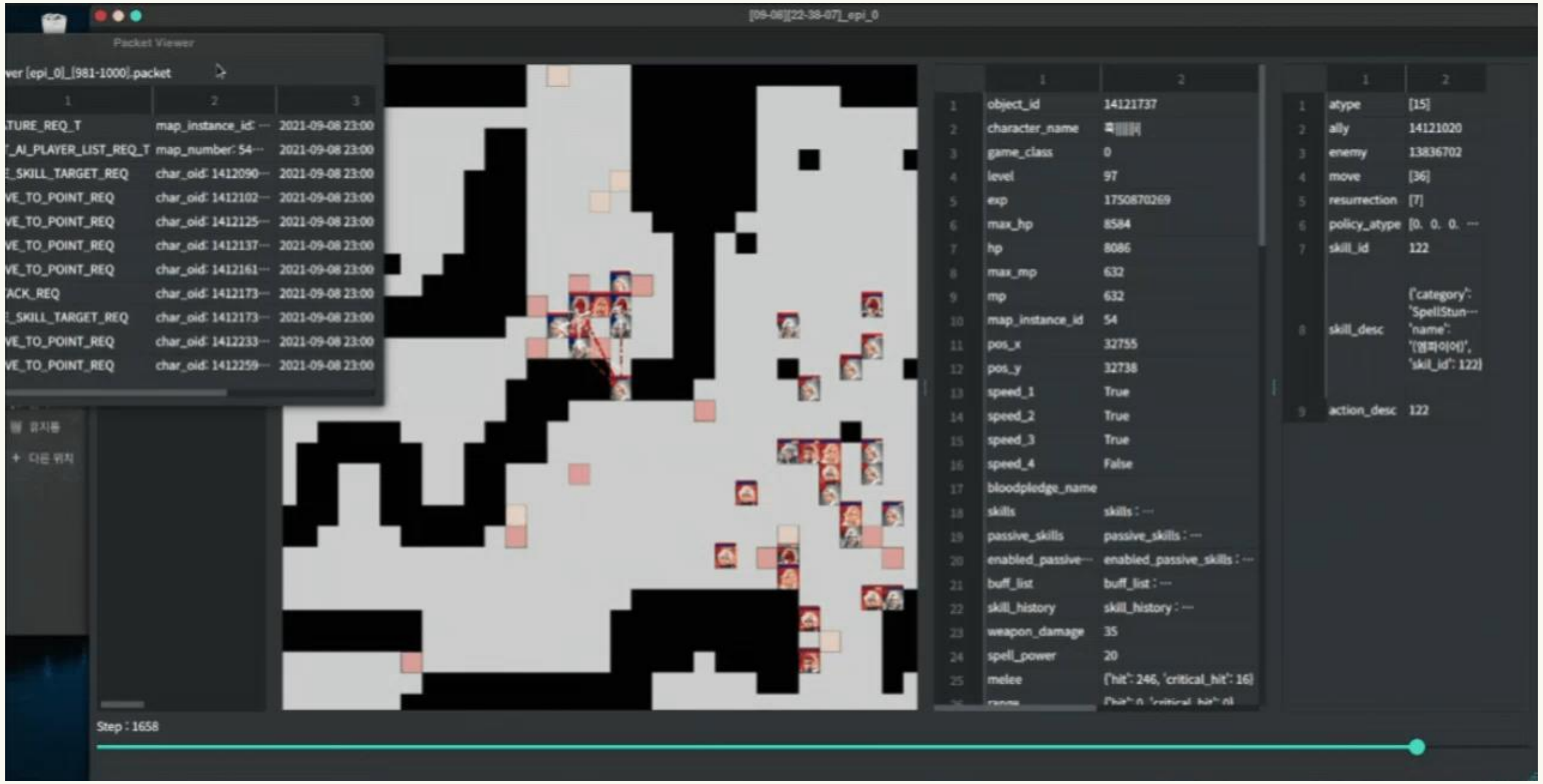
- Abnormality
  - Feature
  - Packets
- Game statistics
  - E.g., The Average kills of AI, The battle time, The maximum number of players



Visualization Tools

Game Environment
- Packet
- Feature

AI Behaviors
- Policy
- Decision
- Feature



Conclusion

- Reinforcement Learning Method
  - Hierarchical decision structure, model sampling & combination sampling, reward shaping
  - Successfully train AIs within 1 week with about 300 CPU cores
- Reinforcement Learning Framework
  - Code automatic generation system, Docker for Windows → reduce time & cost of updating environment
  - Abnormality detection and analysis tools → solve game environment issues & conduct AI behavior analysis
- We were able to create AIs for large-scale battle in MMORPG with MARL

THANK YOU

ohinsuk@ncsoft.com
zenoengine@ncsoft.com

NCSOFT AI Center Game AI Lab & Lineage Camp

Acknowledgements

Key Colleagues

Research Team

Hyunsoo Park

Inseok Oh

Jinyun Chung

Mingyu Yang

Sangbin Moon

Yongchan Park

Engineering Team

Hyoil Lee

Jinhyung Ahn

Kwangsub Kim

Minchul Jung

Taekyung Han