GDC March 20-24, 2023 San Francisco, CA

SQUARE ENIX

The Dimensional Curse of AAA Game Balancing: RL Solution

Edgar Handy
Machine Learning Engineer
SQUARE ENIX CO., LTD.
AI-Division
hanedgar@square-enix.com

#GDC23

AGENDA

- Background
- Basic of Reinforcement Learning (RL)
- Challenges
- RL Algorithm
- Engineering

GAME BALANCING

- Critical part of the development
- Iterative
- Especially expensive for AAA titles

GAME BALANCING

- Critical part of the development
- Iterative
- Especially expensive for AAA titles

AI (Reinforcement Learning)

AI provides references for the designers

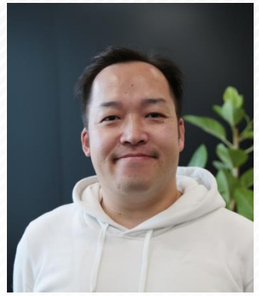
TEAM

AI Division & Advanced Technology Division

Edgar Handy

Kazuhiro Shigekuni

Yuta Mizuno



Youichiro Miyake

Internal Studio

Tomokazu Shibata

REQUIREMENT FOR BATTLE BALANCING

- Have AI play the game many times automatically
- Automatically gather insight useful for the battle designers
  - Human balances the battle.
- Adapts to new unknown stages or enemies
- Learns fast enough

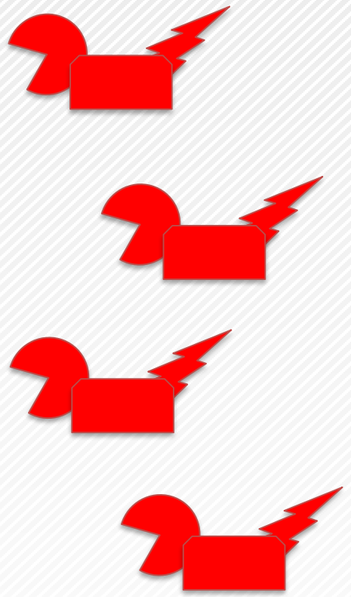
INTRODUCING THE GAME

THE BATTLE MECHANICS

Turn-based battle

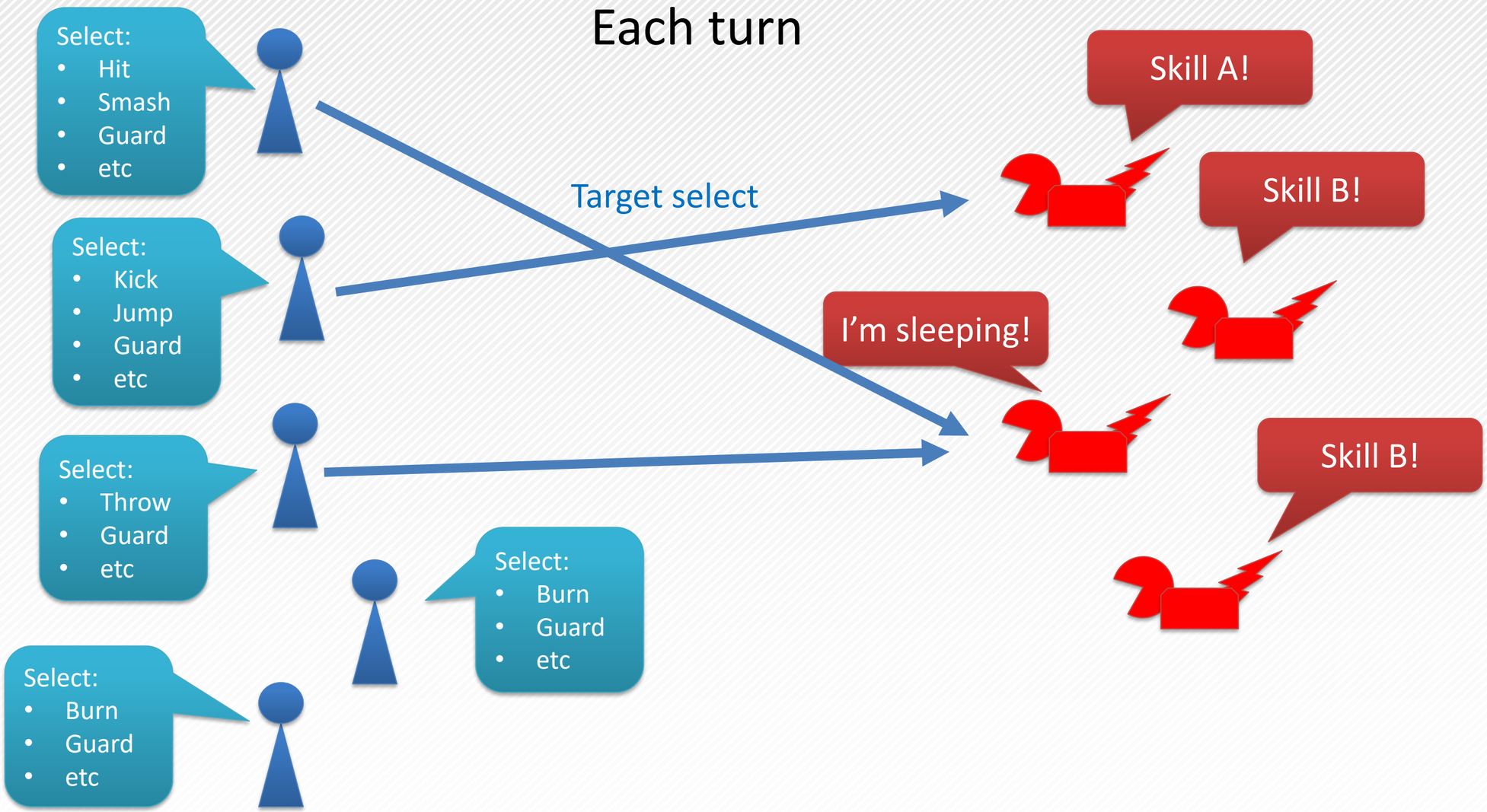
1 to 5 allies

VS

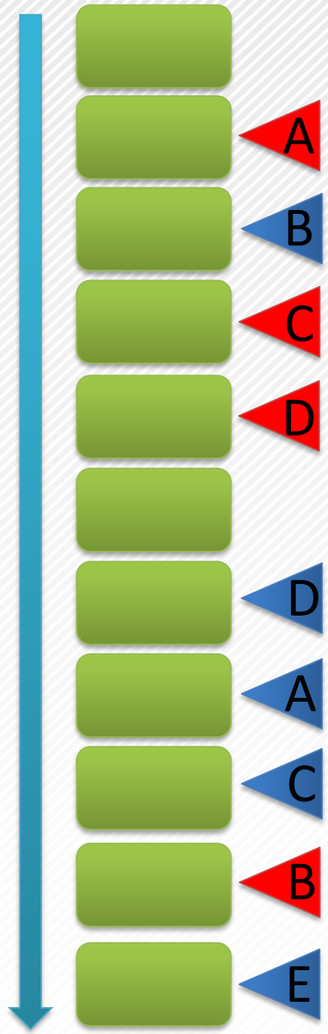


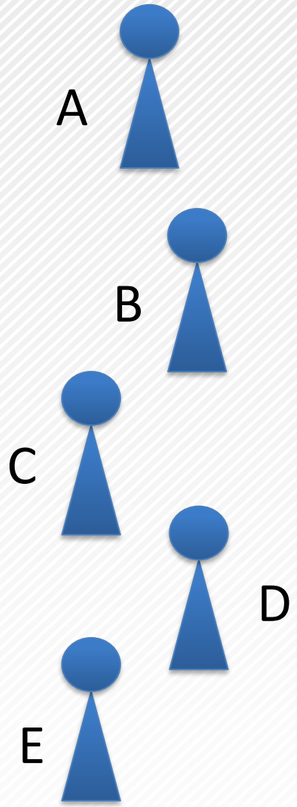
1 to 5 enemies

THE BATTLE MECHANICS

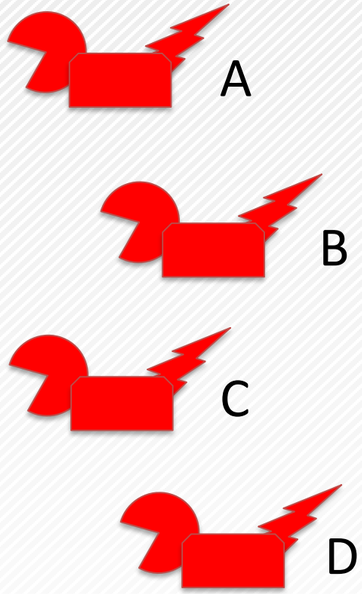


THE BATTLE MECHANICS



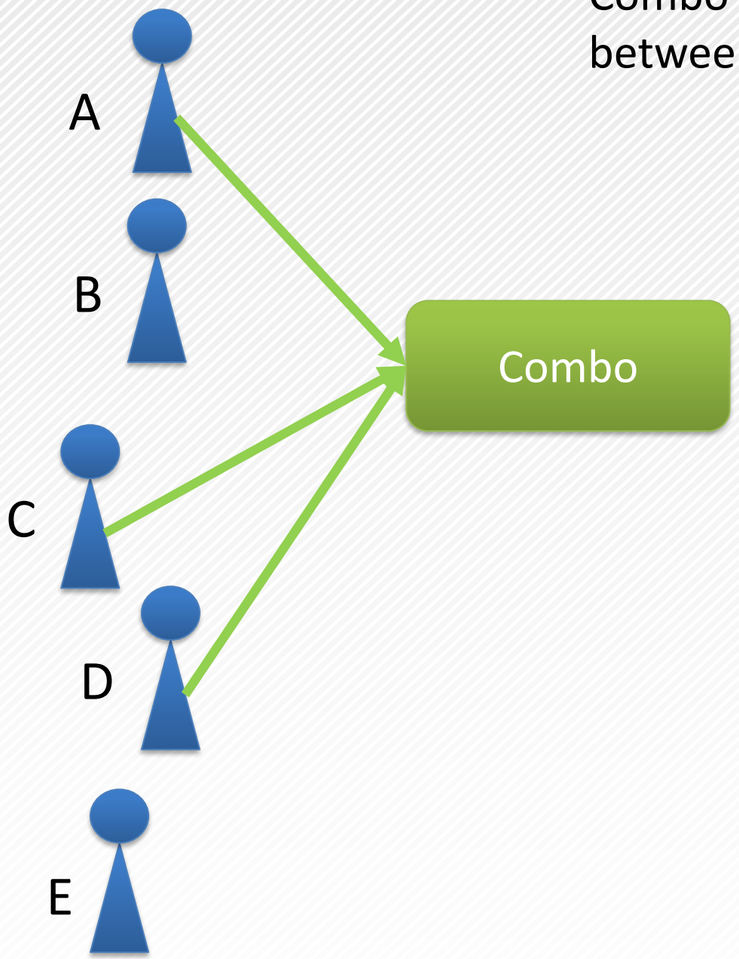


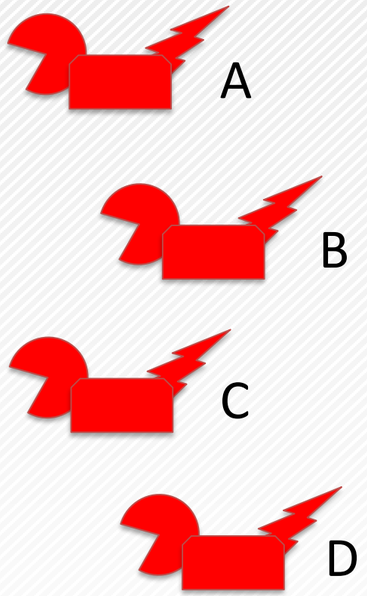
Move Order



THE BATTLE MECHANICS

Combo & other interactions
between allies and enemies



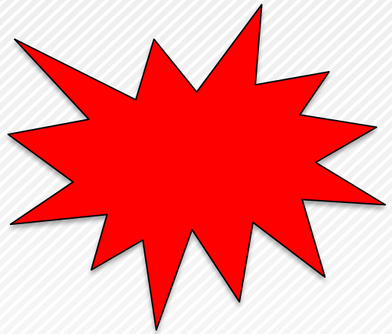


PROJECT A BATTLE FEATURES

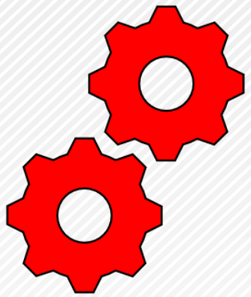
Battle Features:

- 1~5+ players vs 1~5+ enemies (rule-based AI) turn-based battle
- 250+ different enemy units
- 8+ different player units
- 400+ enemy skills and 100+ player skills
- 10+ types of buff and de-buff each
- 100+ different weapons (which affect skills)
- Strategical elements such as combo and other effects.
- Etc.

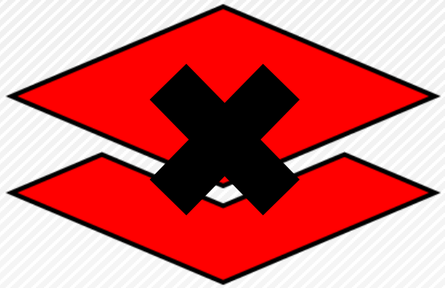
BALANCING OBJECTIVES



Game breaker



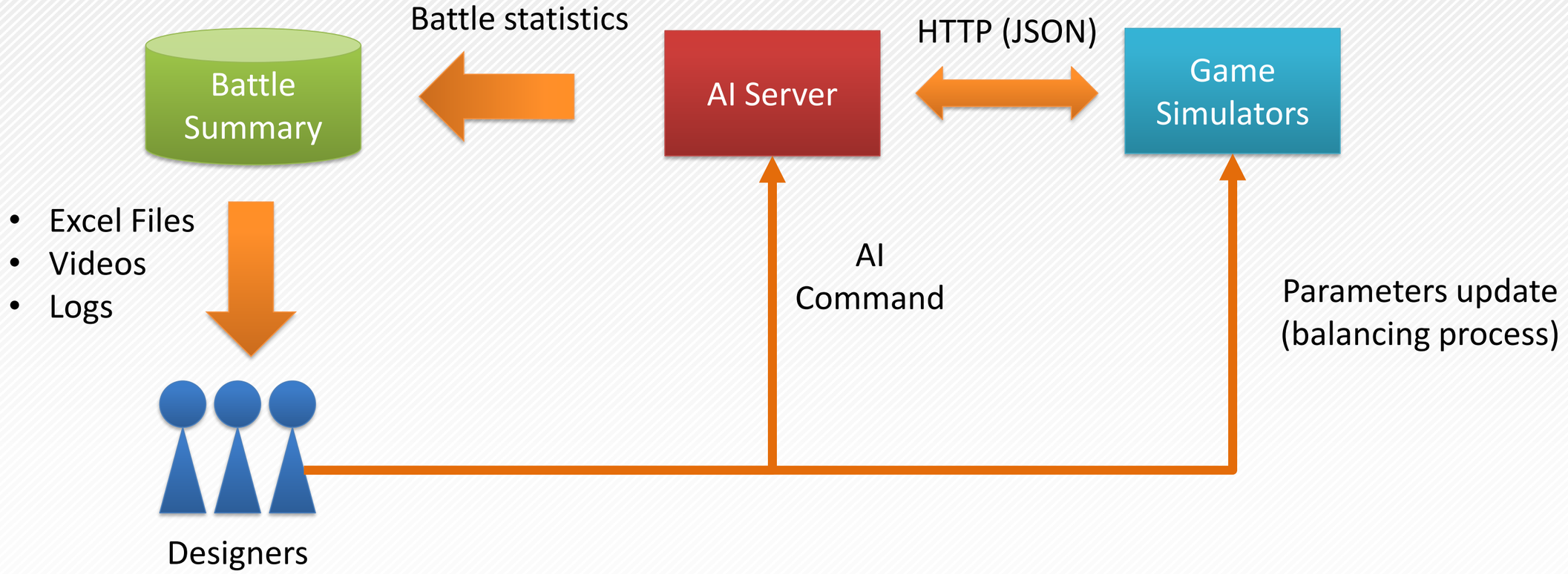
Parameters



Unintended Design

BATTLE BALANCING WITH AI

The AI assists game designers by gathering battle data.



WHY REINFORCEMENT LEARNING (RL)?

- AI that can play smart enough is needed

- RL can be adapted (optimized) to many stages without re-programming

However:

- Comparison against human expert is still needed.

AGENDA

- Background
- Basic of Reinforcement Learning (RL)
- Challenges
- RL Algorithm
- Engineering

MACHINE LEARNING

THE THREE GRAND CRYSTALS

Supervised

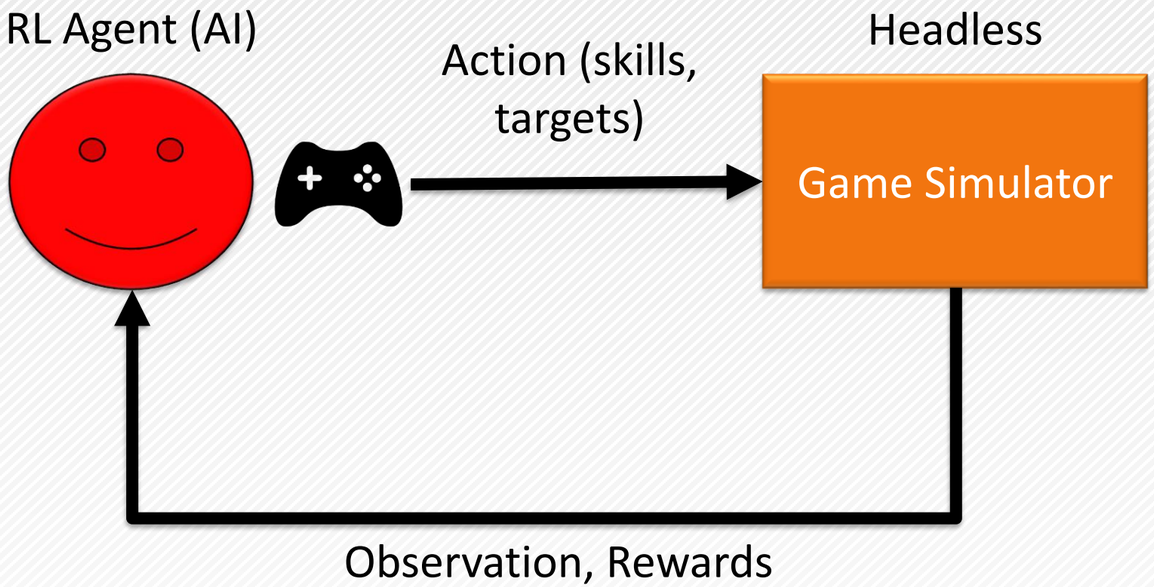
Semi-supervised

What?? Isn't this self-supervised?!

Unsupervised

Reinforcement Learning (RL)

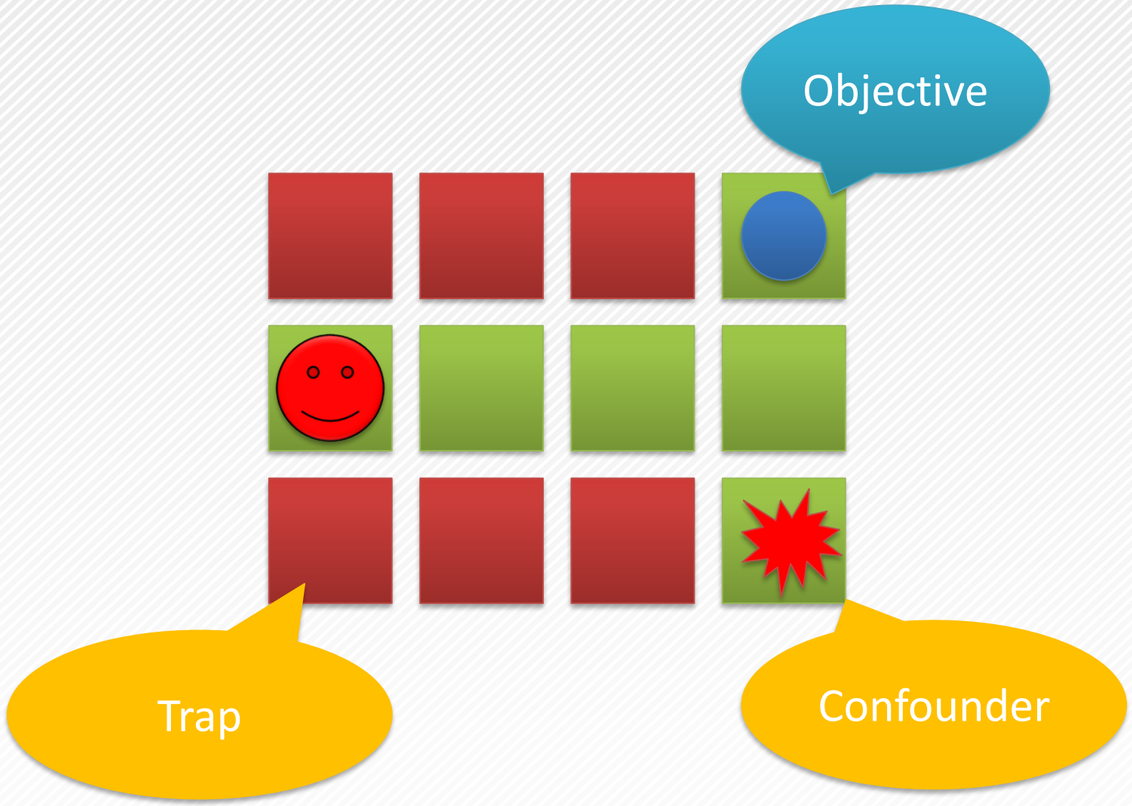
WHAT IS REINFORCEMENT LEARNING (RL)?



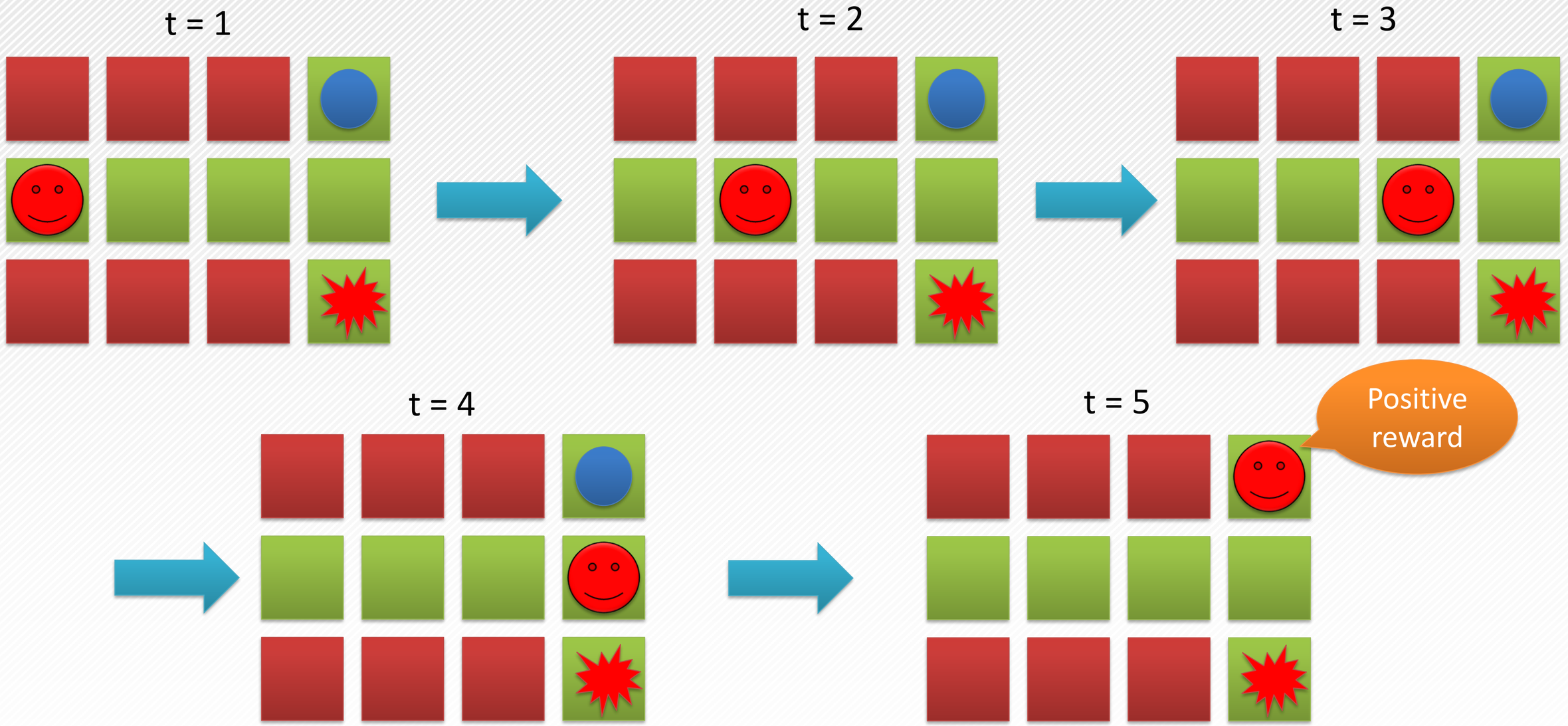
- Action == Agent's decision per timestep (or turn)
- Observation == game state
- Rewards == indicates how good an action is
- Headless == skip rendering

TINY DETAILS

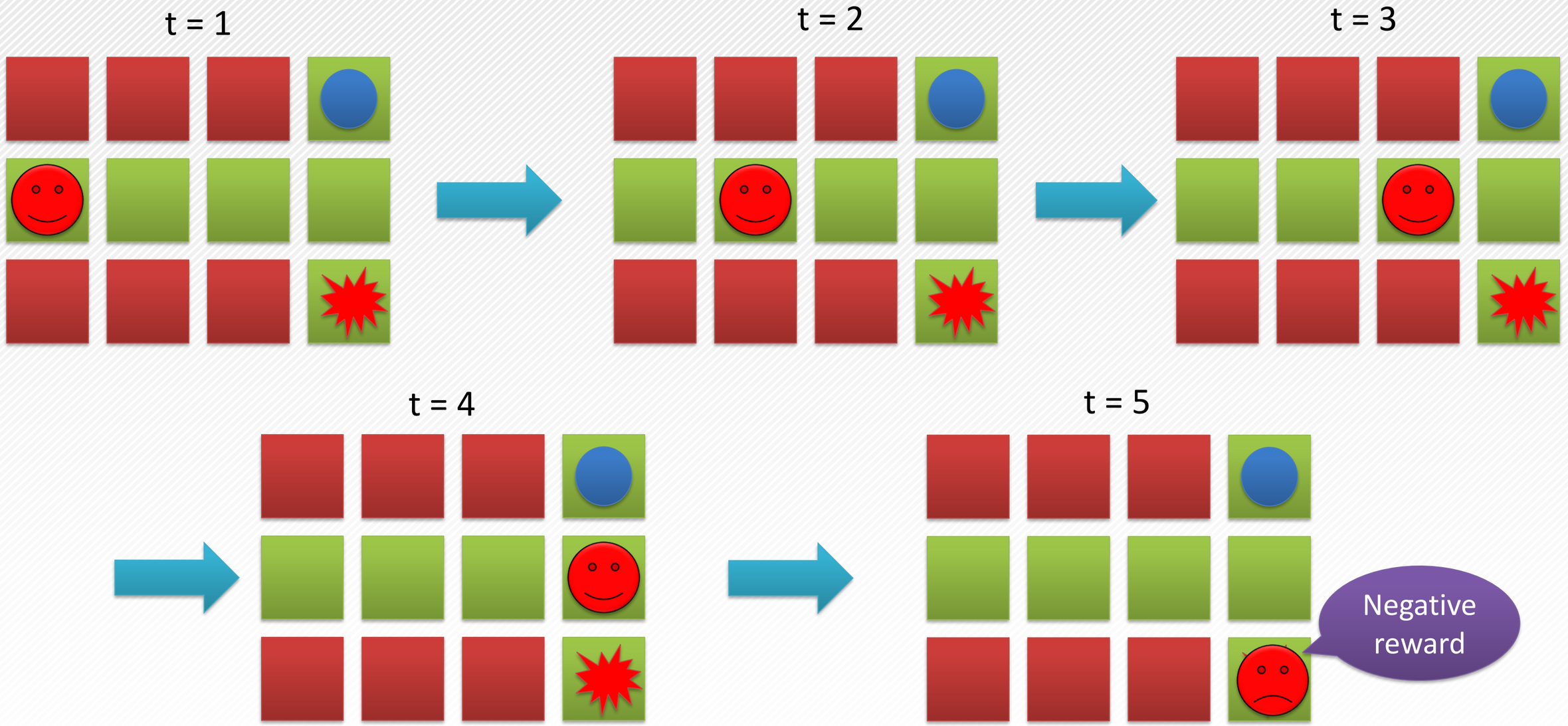
HOW DOES REWARD WORK?



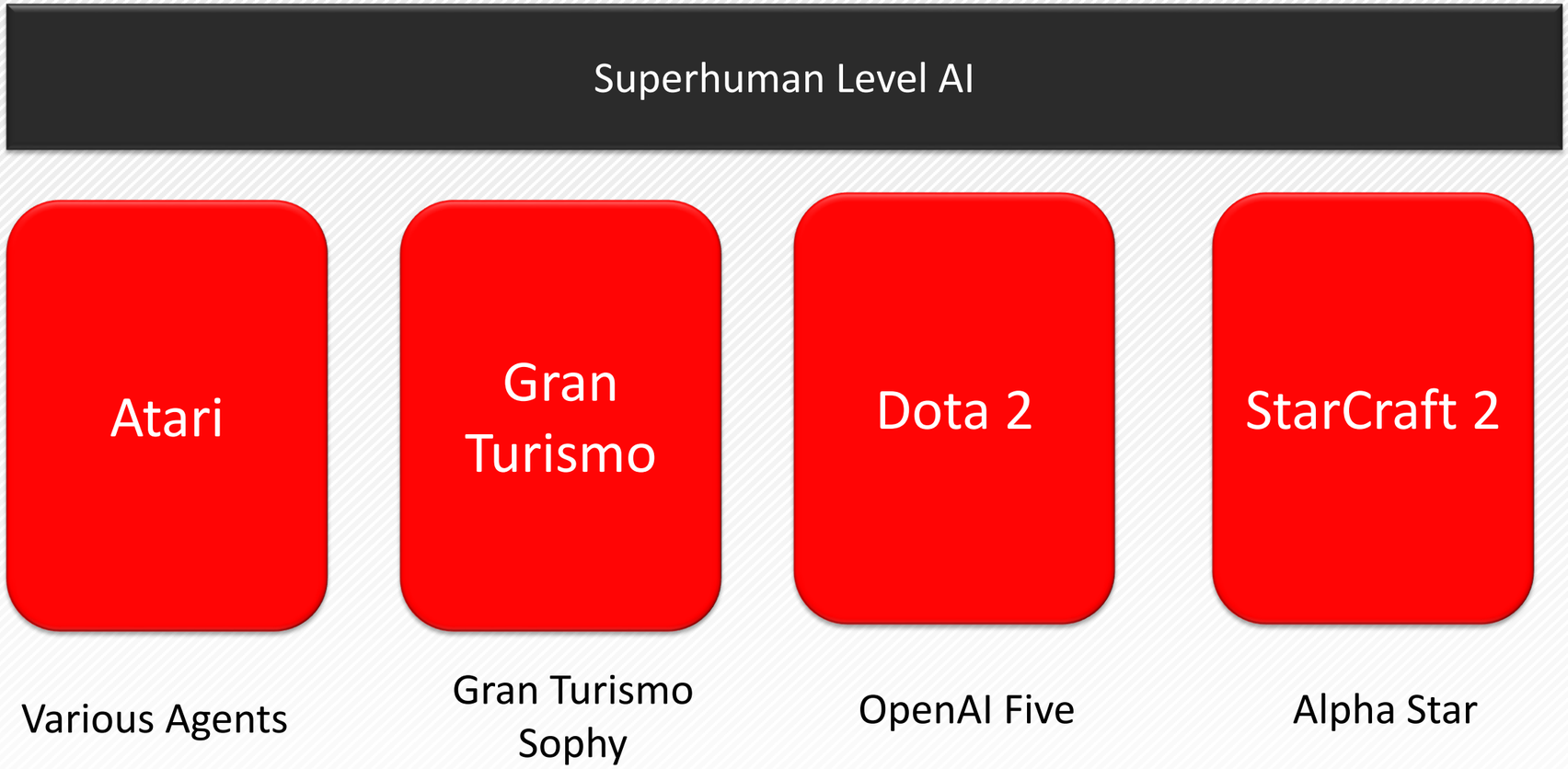
POSITIVE EXAMPLE



NEGATIVE EXAMPLE



RAISE OF THE RL

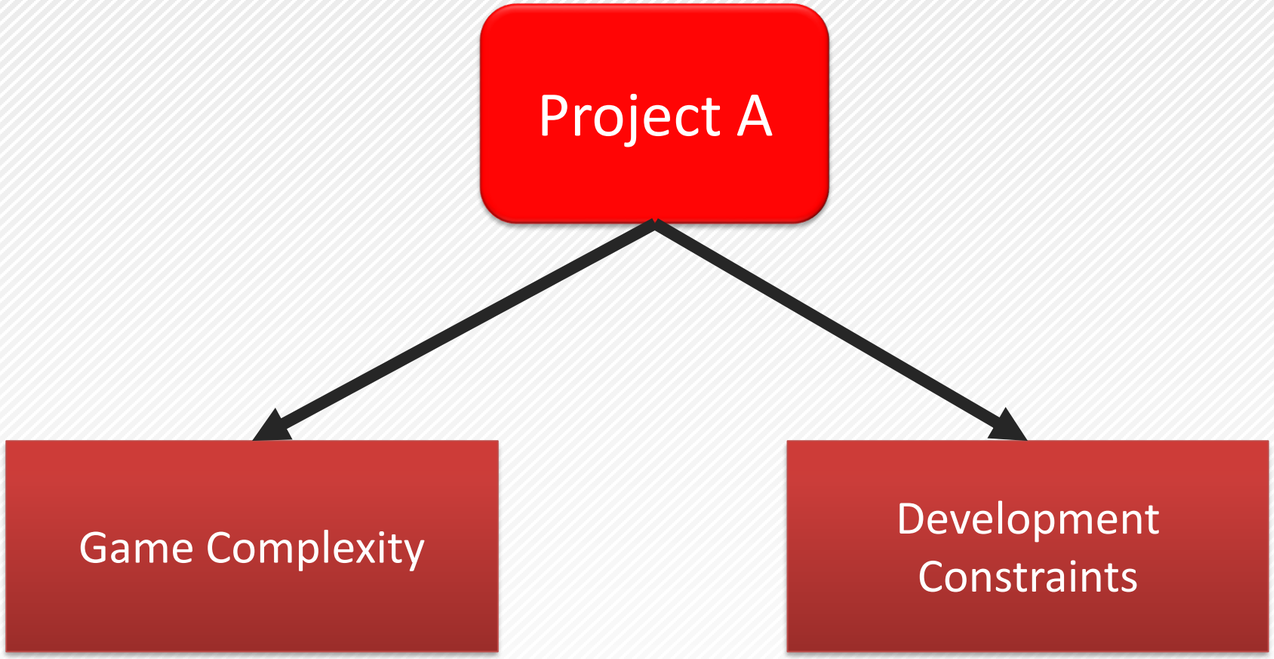


https://www.gran-turismo.com/us/gran-turismo-sophy/
https://github.com/mgbellemare/Arcade-Learning-Environment
https://openai.com/five/
https://www.deepmind.com/blog/alphastar-mastering-the-real-time-strategy-game-starcraft-ii

AGENDA

- Background
- Basic of Reinforcement Learning (RL)
- Challenges
- RL Algorithm
- Engineering

CHALLENGES



CHALLENGES

GAME COMPLEXITY

Project A

Action Space	~20 million to ~40 million
Observation Space	200+ dimensions, continuous values
Rewards Density	Extremely sparse on hard stages

CHALLENGES

GAME COMPLEXITY

https://openai.com/five/
https://www.deepmind.com/blog/alphastar-mastering-the-real-time-strategy-game-starcraft-ii

Project A

- ~20 million actions
- sparse rewards
- Unknown enemies & actions

Need something cheaper and faster

Superhuman level AI is not necessary

Dota 2

- ~2 million actions + multi-agent
- sparse rewards
- Known enemies & actions

80,000 – 178,000x CPUs
2000 - 3000x GPUs
PPO

Superhuman level AI

StarCraft 2

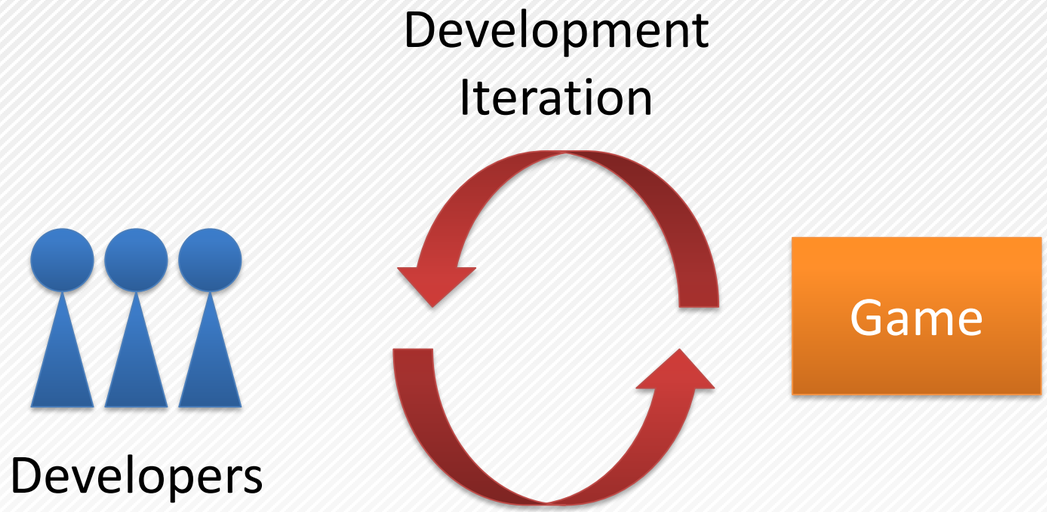
- ~1 billion+ actions
- sparse rewards
- Known enemies & actions

128x TPU Cores
Years of supervision data
V-Trace Variant

Superhuman level AI

CHALLENGES

DEVELOPMENT COMPLEXITY: WALL-CLOCK TIME



CHALLENGES

DEVELOPMENT COMPLEXITY: CONTINUOUS PATCHES

CHALLENGES

DEVELOPMENT COMPLEXITY: UNSTABLE & SLOW SIMULATOR

Frequent crashes and slow data collection

SUMMARY OF CHALLENGES

- Huge action, observation space, sparse rewards
- Slow simulator == data sparsity
- Game being patched all the time
  - Unknown enemies, skills, etc.
  - Moving distribution
- Unstable Simulators
- Wall-clock time & hardware constraint

THE SOLUTIONS

Reinforcement Learning

Engineering

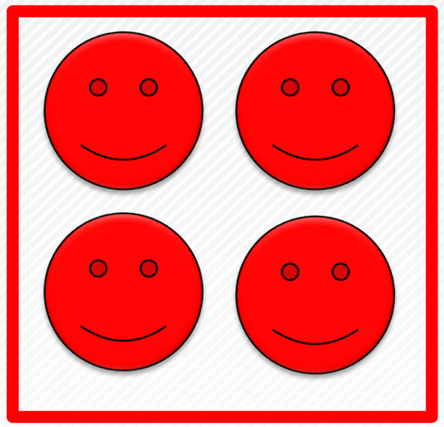
AGENDA

- Background
- Basic of Reinforcement Learning (RL)
- Challenges
- RL Algorithm
- Engineering

THE BASIC LEARNING SETUP

Send game state (observation) every turn via HTTP protocol.
Rewards are computed on the AI side.
Headless == skip rendering

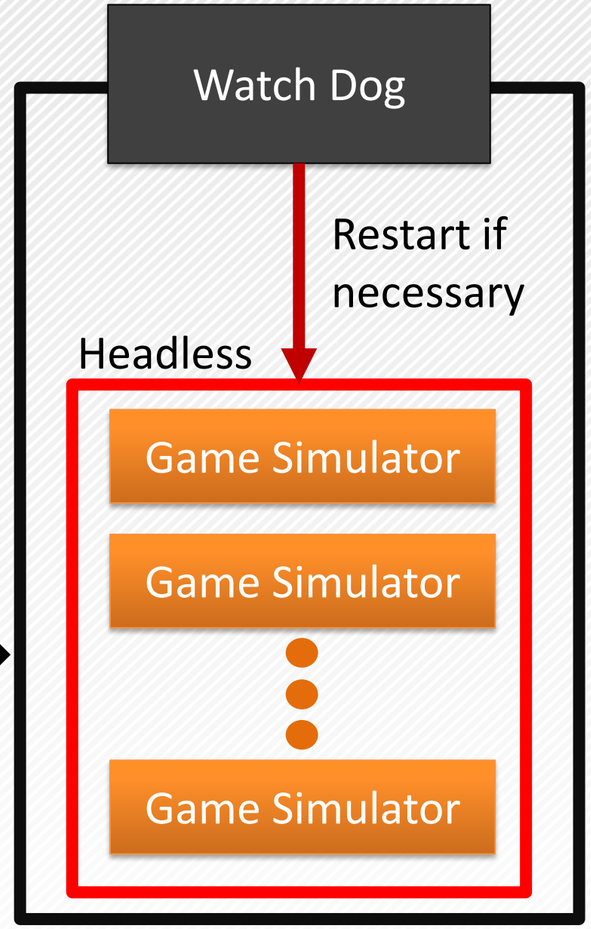
AI Server (running in Python)
Communication-error aware



Action

HTTP (JSON)

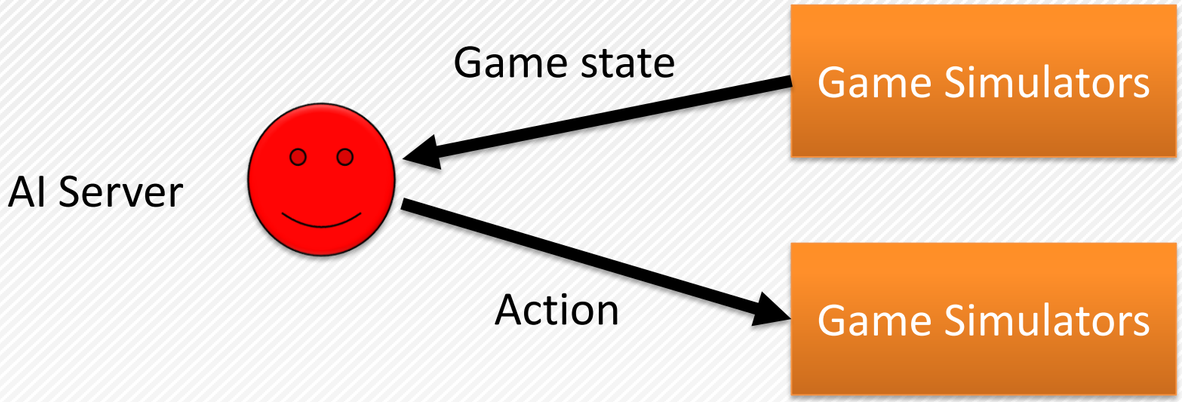
Game State

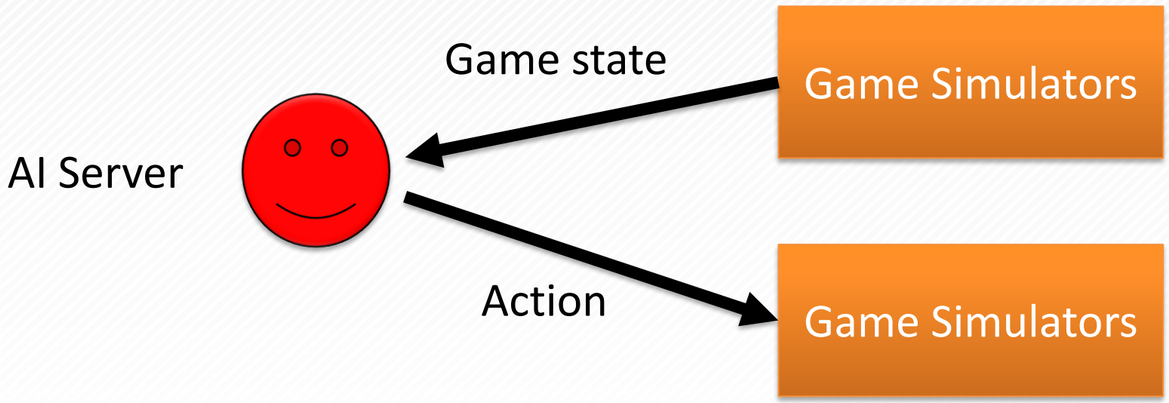


32 simulators distributed within 4 PCs

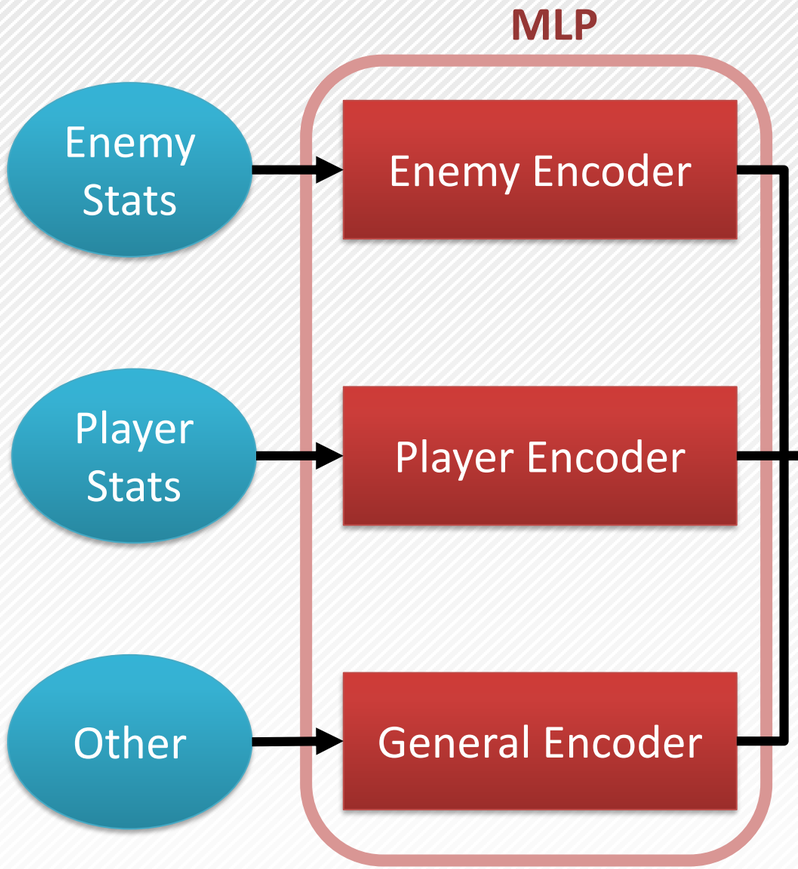
HOW THE AI PLAY?

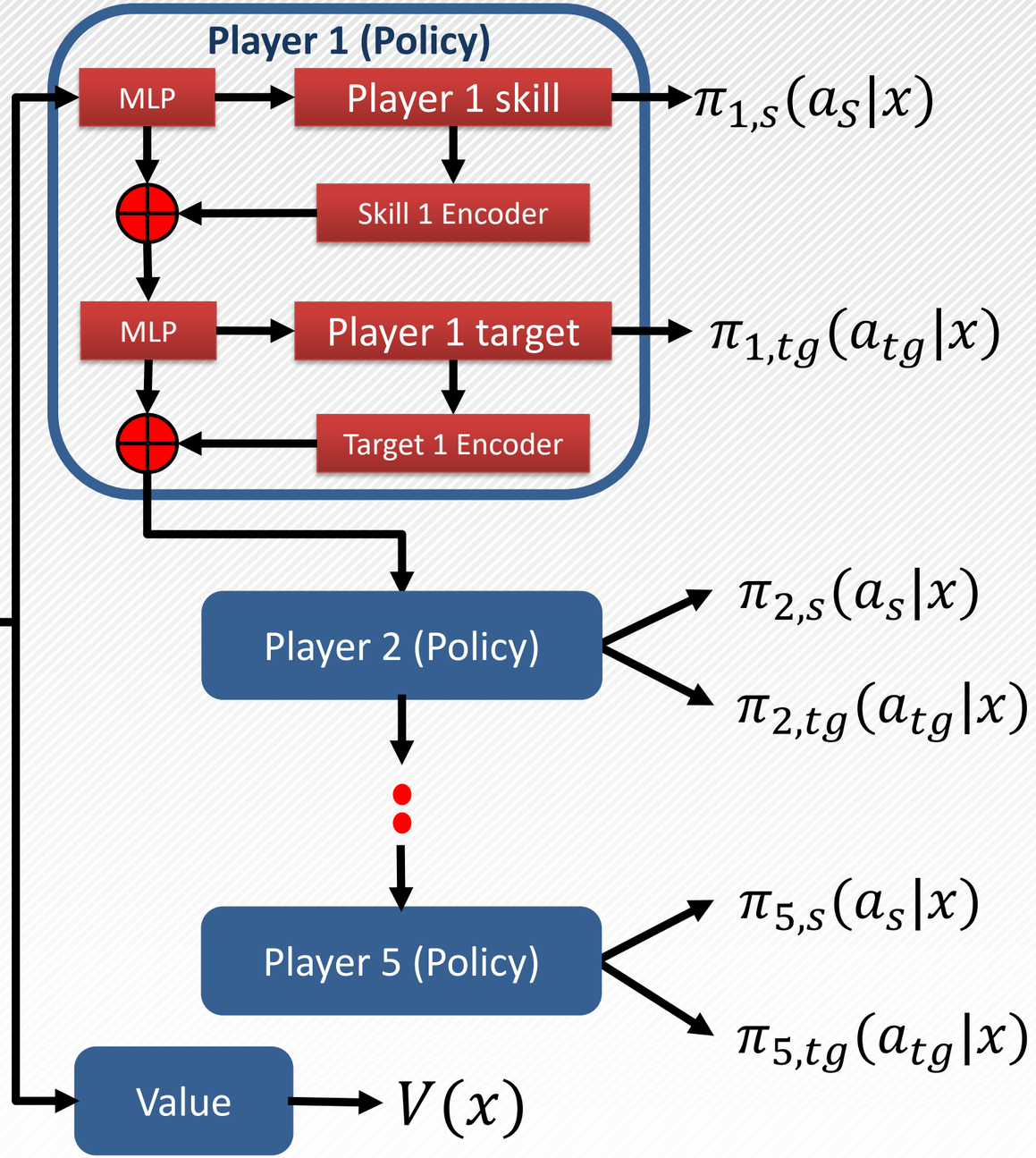
Send game state (observation) every turn via HTTP protocol.
Rewards are computed on the AI side.





THE AI MODEL



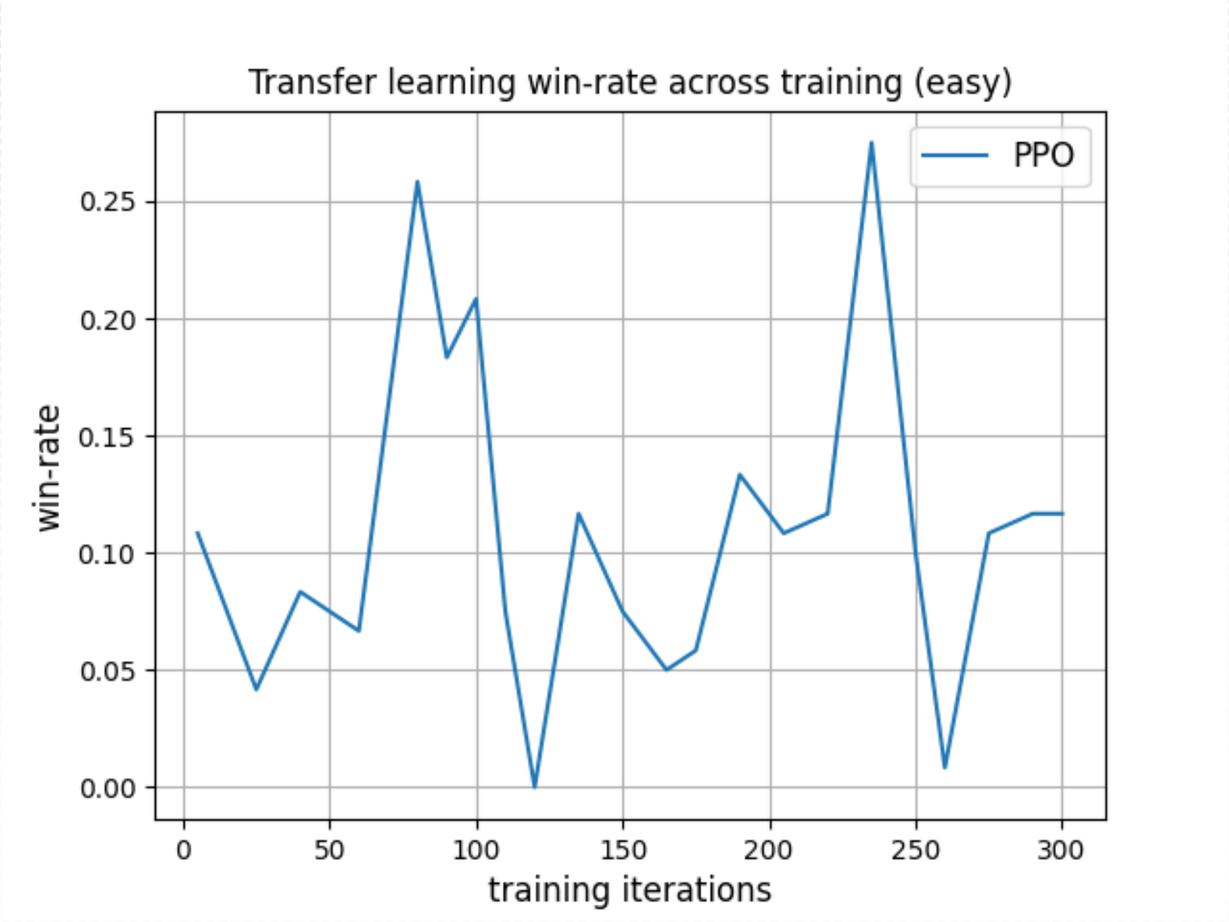


MLP = Multilayer Perceptron

= Element-wise addition

THE RL ADVENTURE

Prototyped with Proximal Policy Optimization (PPO)



Win-rate not really improving.
Cause:

- Enemies' growth are not kept fixed

2017. Schulman et al. Proximal Policy Optimization Algorithms

THE RL ADVENTURE

PPO performance after fixing enemy growth

Another problems:

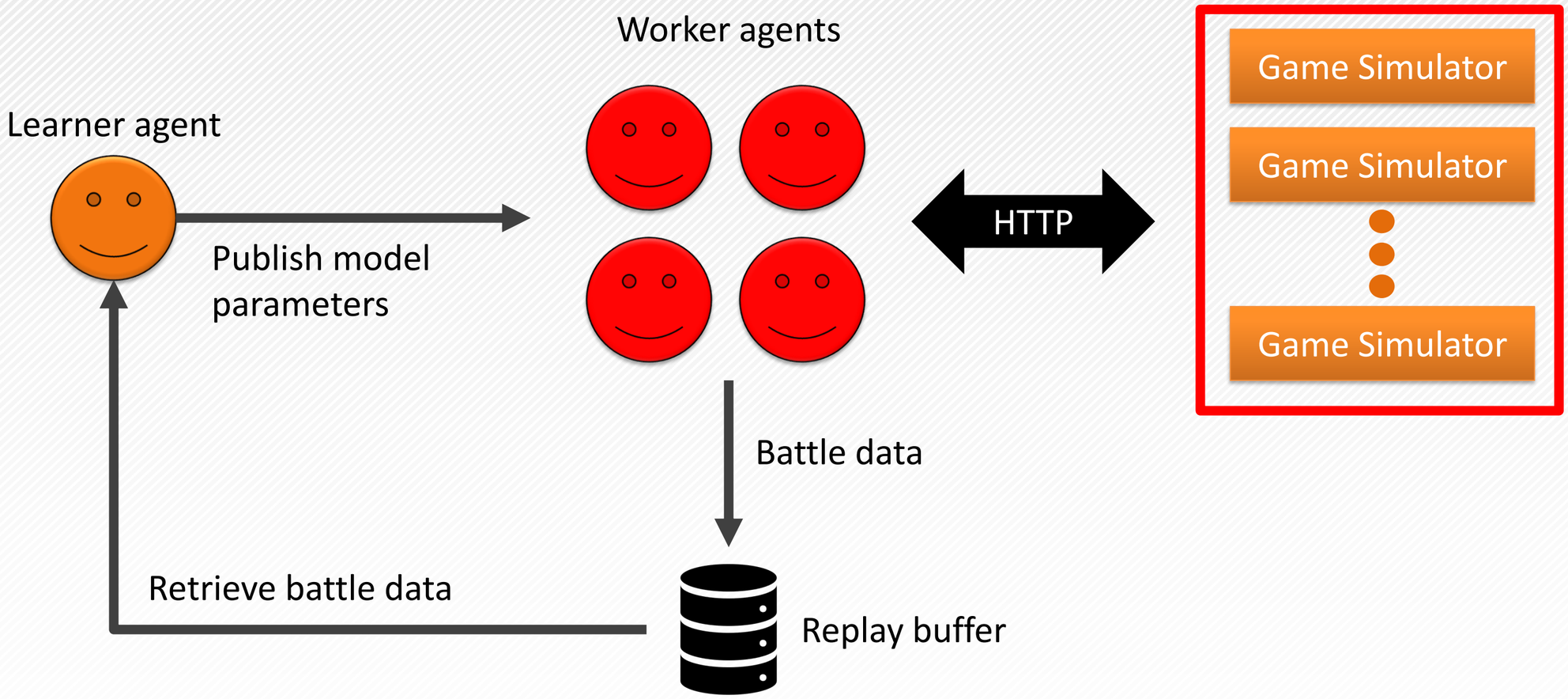
- Win-rate not stable
- Performance dropped in harder stages
- One stage training time is 18 hours

HECL

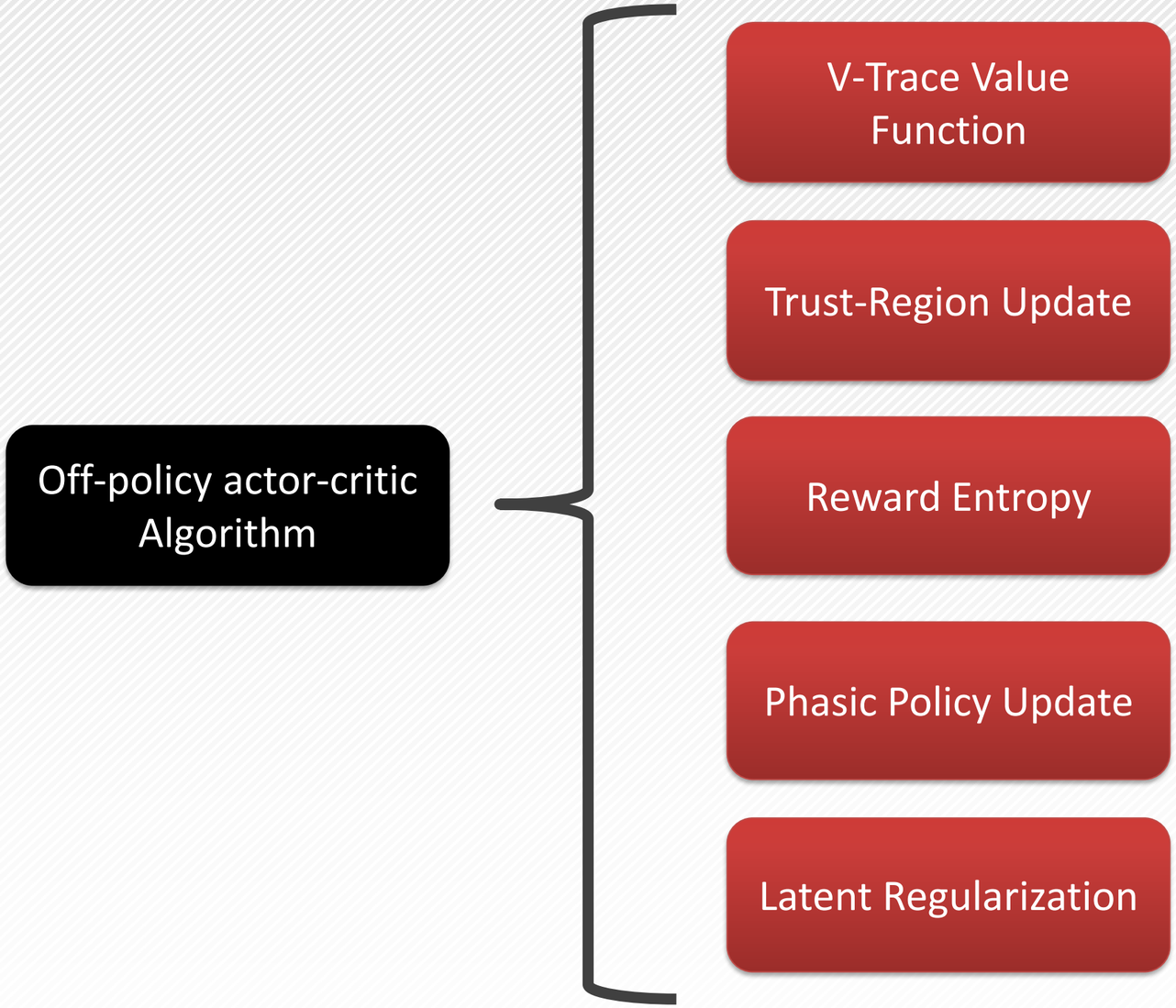
High Entropy Composite Learner

ASYNCHRONOUS WORKER AGENTS

Worker agents gather training data from the game simulators.



HECL



V-TRACE

$$\mathrm{v_t} = V(x_t) + \sum_{k=t}^{t+n-1} \gamma^{k-t} \left(\prod_{i=t}^{k-1} c_i \right) \delta_k V, \qquad where\ \delta_k V = \rho_k \left(r_k + \gamma V(x_{k+1}) - V(x_k) \right)$$

$$\mathrm{c_i} = \min\left(\bar{c}, \frac{\pi(a_i|x_i)}{\mu(a_i|x_i)} \right), \rho_i = \min\left(\bar{\rho}, \frac{\pi(a_i|x_i)}{\mu(a_i|x_i)} \right)$$

$$\pi = \text{current policy}, \mu = \text{behavior policy}, \bar{c} = 1.0, \bar{\rho} = 1.0$$

$$\text{Value function update: } \mathrm{J}(\phi) = E_{x_t \sim D}[V(x_t) - v_t], where\ D\ is\ a\ replay\ buffer.$$

Legend:

- r_t = reward at time-step t.
- $V(x_t)$ = Value of being at state x at time-step t
- $\pi(a_i|x_i)$ = Current policy (the learner). Probability of taking the action a on state x at time-step i
- $\mu(a_i|x_i)$ = Behavior policy (the worker). Probability of taking the action a on state x at time-step i
- γ = Discount factor. $0.0 \leq \gamma < 1.0$
- ϕ = model parameters

2018. Espeholt et al. IMPALA: Scalable Distributed Deep-RL with Importance Weighted Actor-Learner Architectures

TRUST-REGION UPDATE & REWARD ENTROPY

Policy update: $\mathrm{J}(\theta) = \mathrm{E}_t\left[\right.$ [redacted] $\left.\right]$

$$\mathrm{G}_t = v_t - V(x_t) - e^{\alpha} log \pi_\theta(a_t|x_t)$$

$$\mathrm{J}(\alpha) = \alpha E_{a_t \sim \pi_\theta}[-log \pi_\theta(a_t|x_t) - H]$$

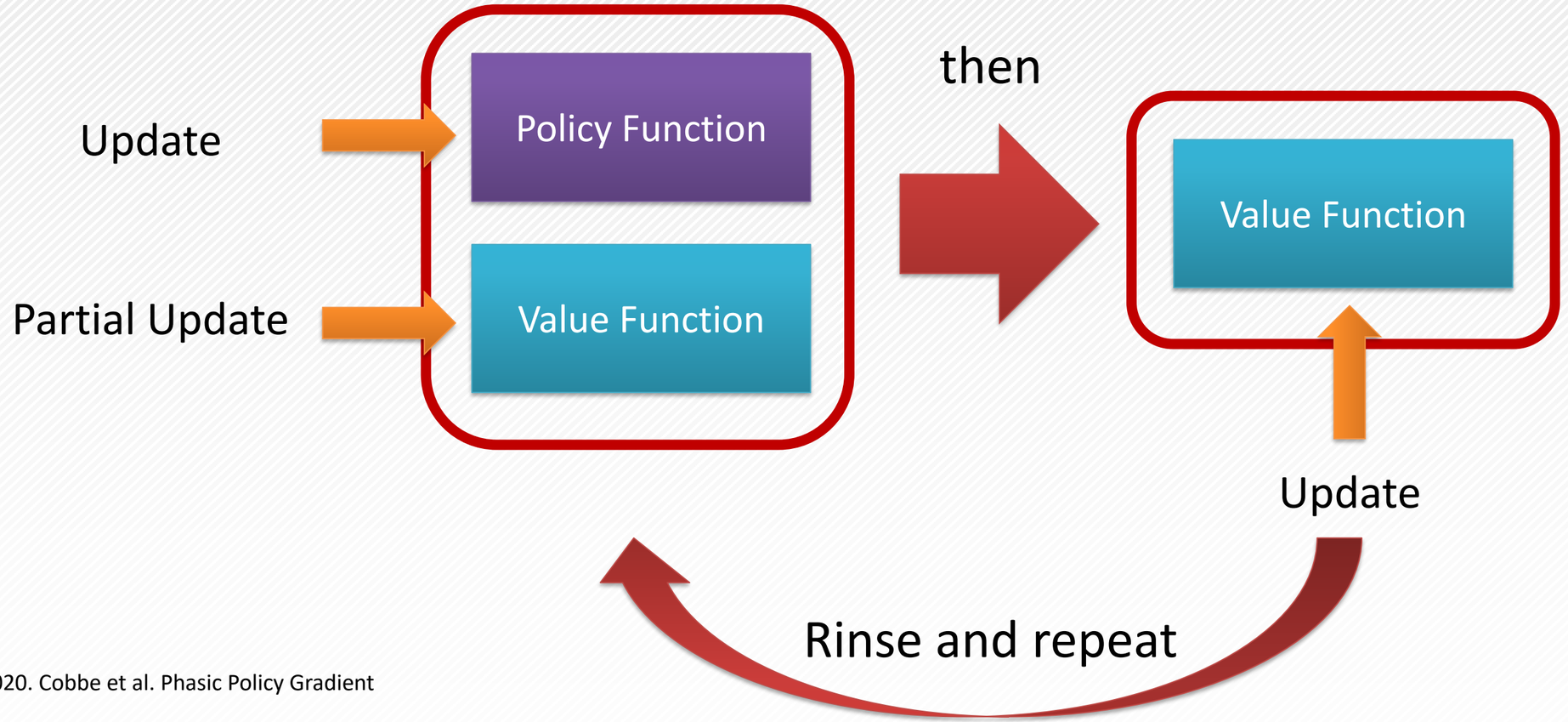
Legend:

- r_t = reward at time-step t
- $V(x_t)$ = Value of being at state x at time-step t
- $\pi_\theta(a_t|x_t)$ = Current policy (the learner). Probability of taking the action a on state x at time-step t
- $\pi_{prox}(a_t|x_t)$ = Proximal policy (the learner). Probability of taking the action a on state x at time-step t
- $\mu(a_t|x_t)$ = Behavior policy (the worker). Probability of taking the action a on state x at time-step t
- θ = model parameters
- α = entropy reward scale
- G_t = return (reward-to-go) at time t
- v_t = V-Trace value function

2018. Haarjona et al. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor
2017. Schulman et al. Proximal Policy Optimization Algorithms

PHASIC POLICY UPDATE

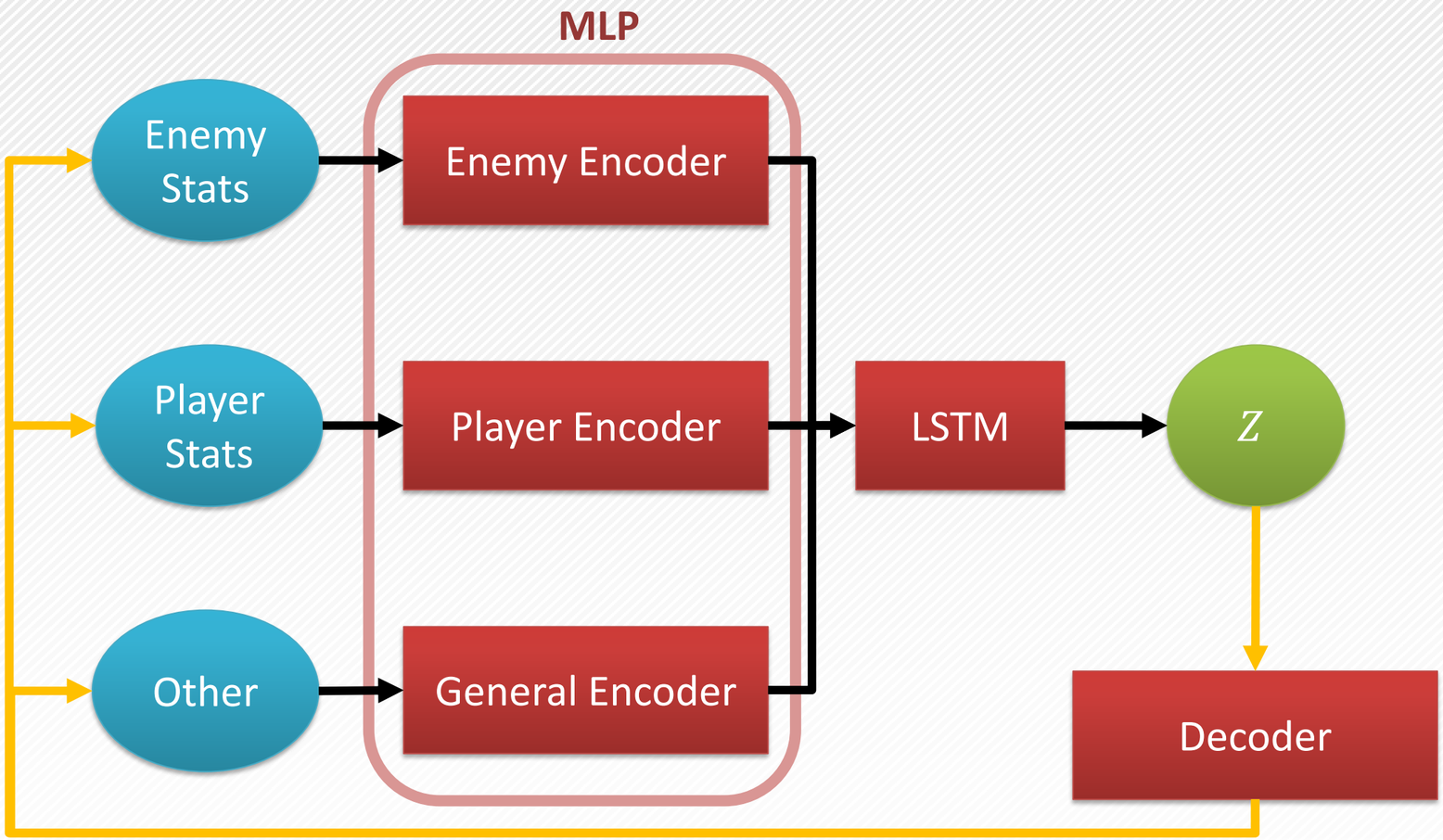
Update policy function and value function at different schedule



2020. Cobbe et al. Phasic Policy Gradient

LATENT REGULARIZATION

Reconstruct the observation using latent state. Relatively stabilizes the learning

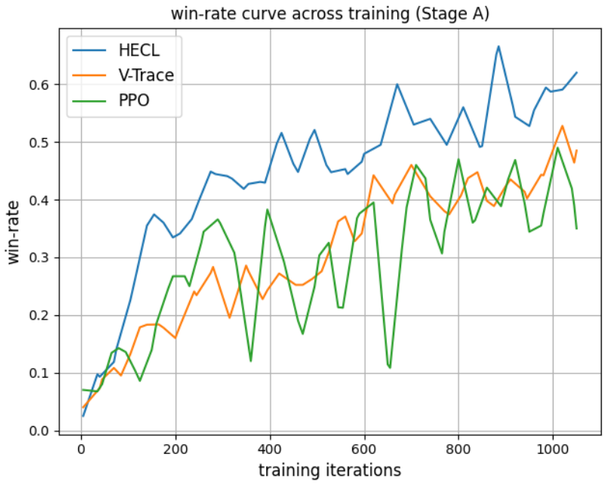


PERFORMANCE COMPARISON

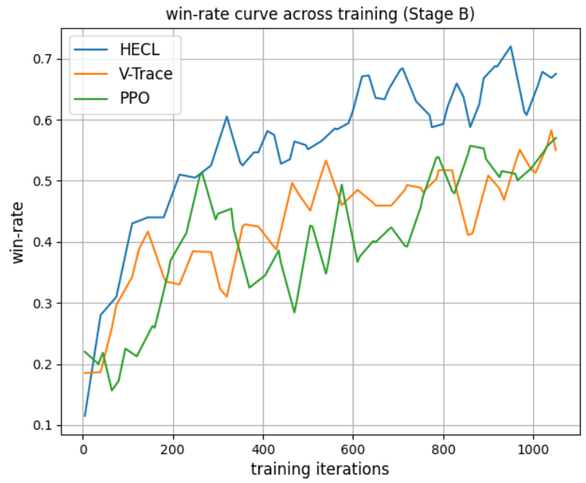
HECL VS PPO VS V-TRACE

PERFORMANCE COMPARISON (WIN-RATE)

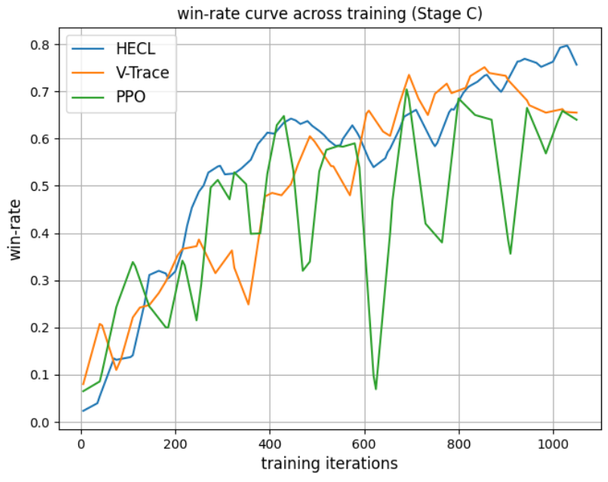
Stage A



Stage B



Stage C



Impossible

Difficulty:
Stage A > Stage B > Stage C

Impossible:
unbalanced stage which is virtually impossible to be cleared

11 hours of training

PERFORMANCE COMPARISON

Stage A

Difficulty:
Stage A > Stage B > Stage C

Impossible:
unbalanced stage which is virtually impossible to be cleared

11 hours of training

PERFORMANCE COMPARISON

Stage B

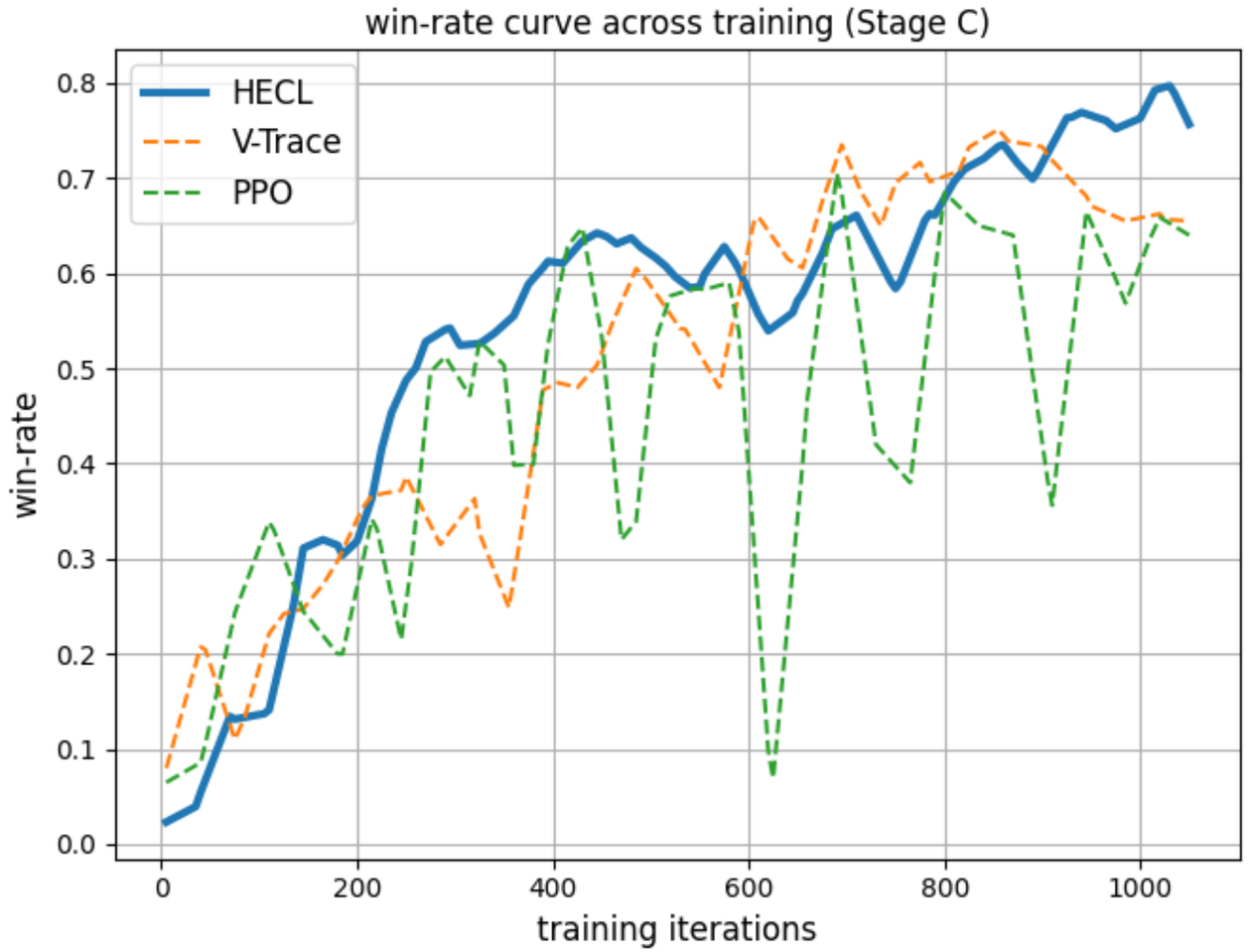
Difficulty:
Stage A > Stage B > Stage C

Impossible:
unbalanced stage which is virtually impossible to be cleared

11 hours of training

PERFORMANCE COMPARISON

Stage C



Difficulty:
Stage A > Stage B > Stage C

Impossible:
unbalanced stage which is virtually impossible to be cleared

11 hours of training

PERFORMANCE COMPARISON

Impossible!

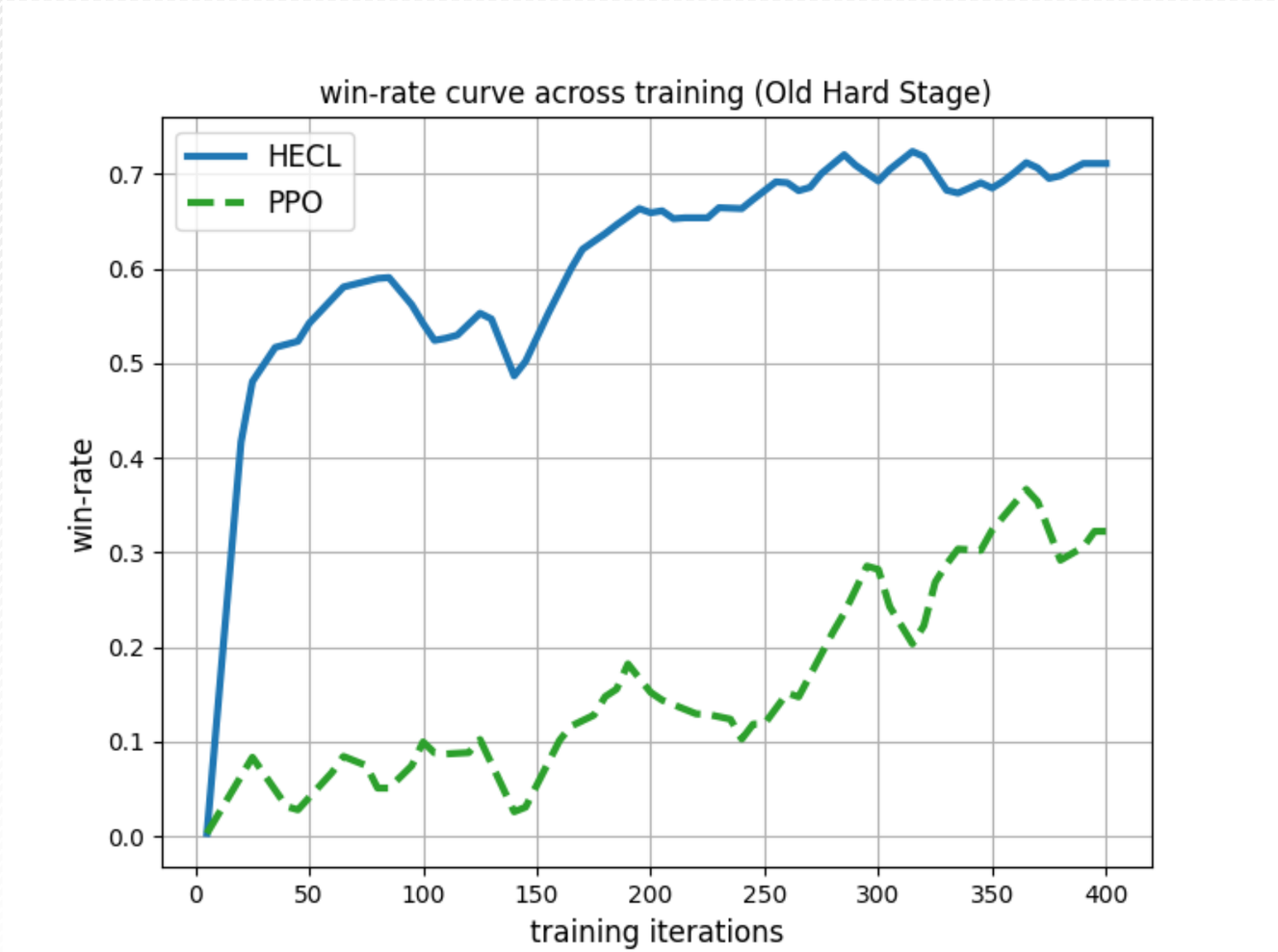
Difficulty:
Stage A > Stage B > Stage C

Impossible:
unbalanced stage which is virtually impossible to be cleared

11 hours of training

PERFORMANCE COMPARISON

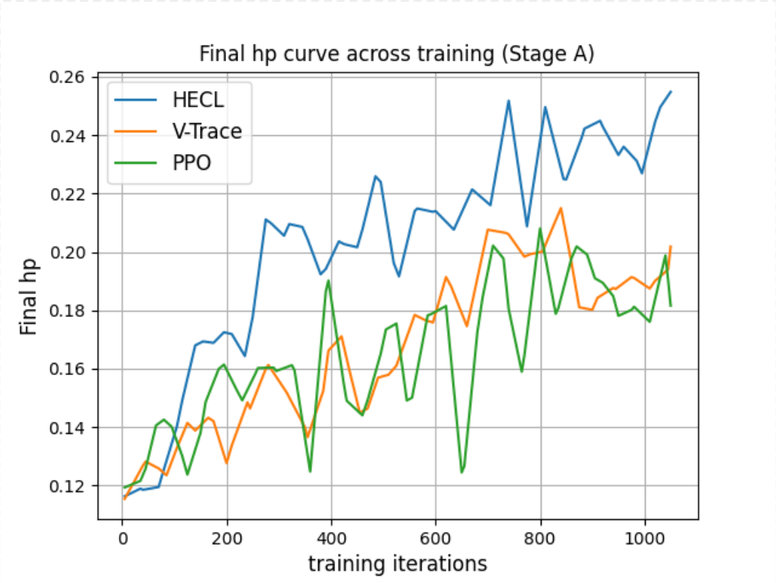
HECL vs PPO on old simulator hard stage.



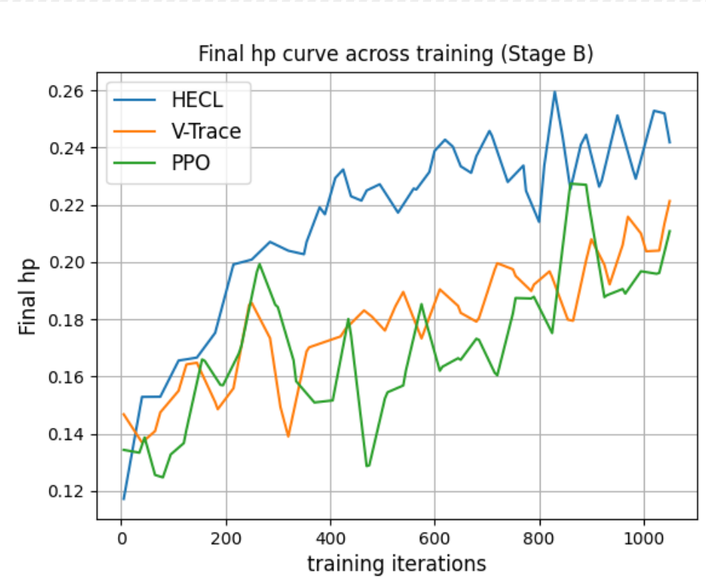
PERFORMANCE COMPARISON

NORMALIZED PLAYER FINAL HP

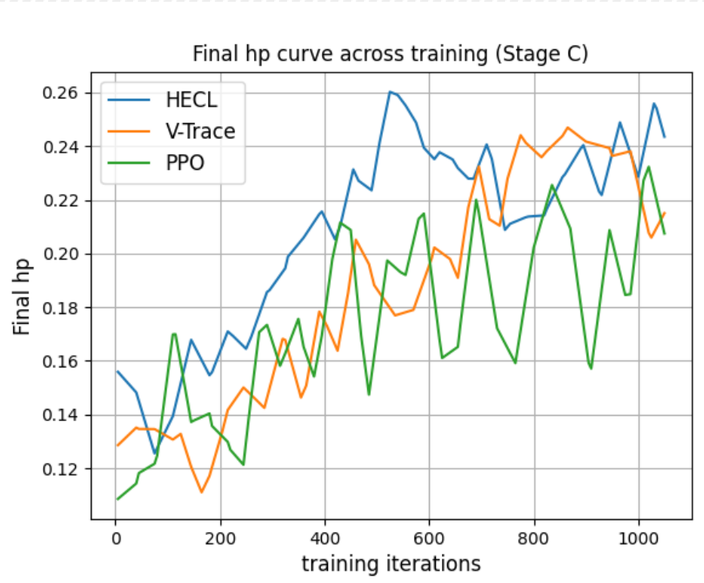
Stage A



Stage B



Stage C

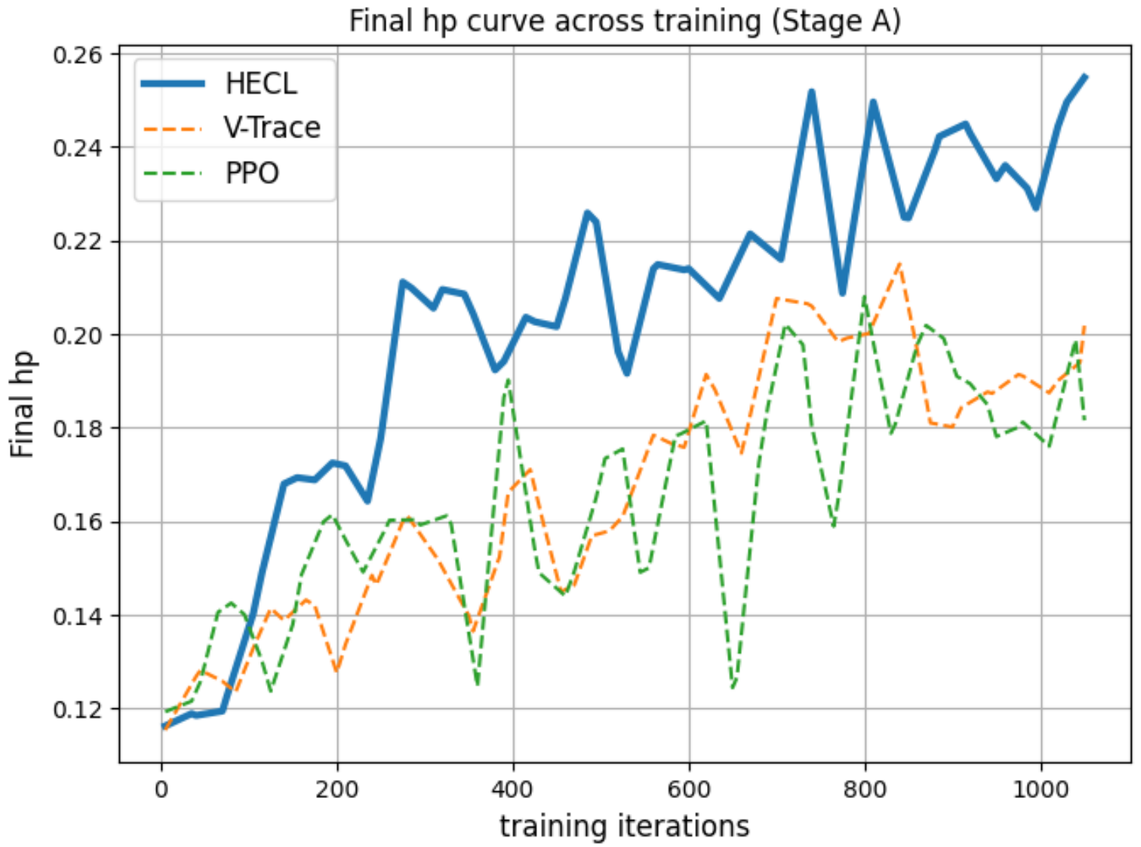


Difficulty:
Stage A > Stage B > Stage C

PERFORMANCE COMPARISON

NORMALIZED PLAYER FINAL HP

Stage A



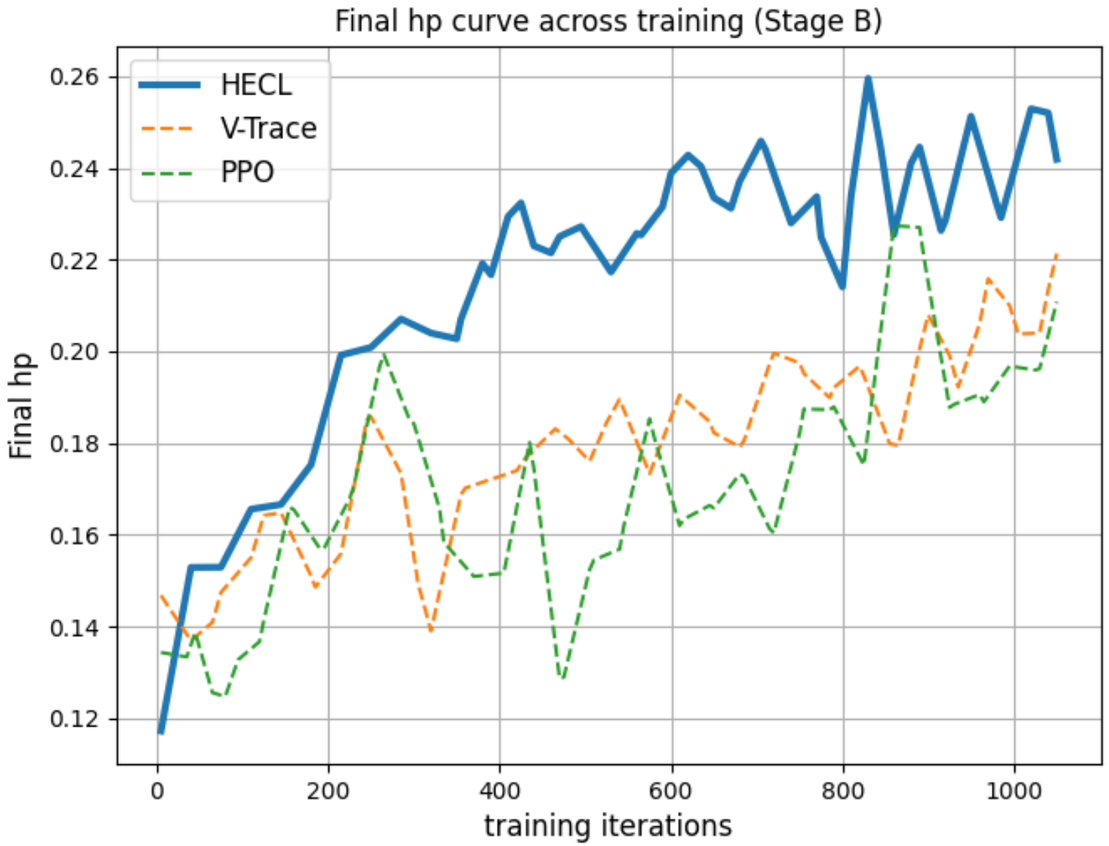
Difficulty:
Stage A > Stage B > Stage C

11 hours of training

PERFORMANCE COMPARISON

NORMALIZED PLAYER FINAL HP

Stage B



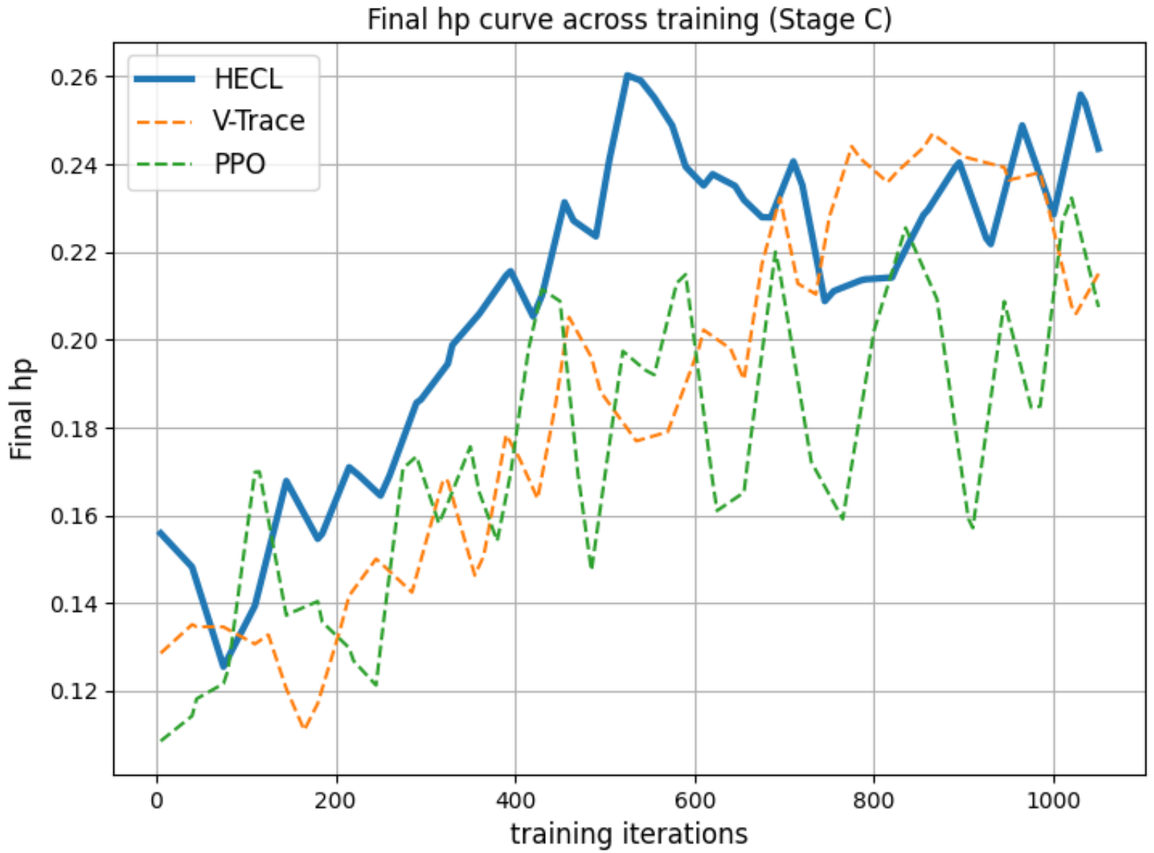
Difficulty:
Stage A > Stage B > Stage C

11 hours of training

PERFORMANCE COMPARISON

NORMALIZED PLAYER FINAL HP

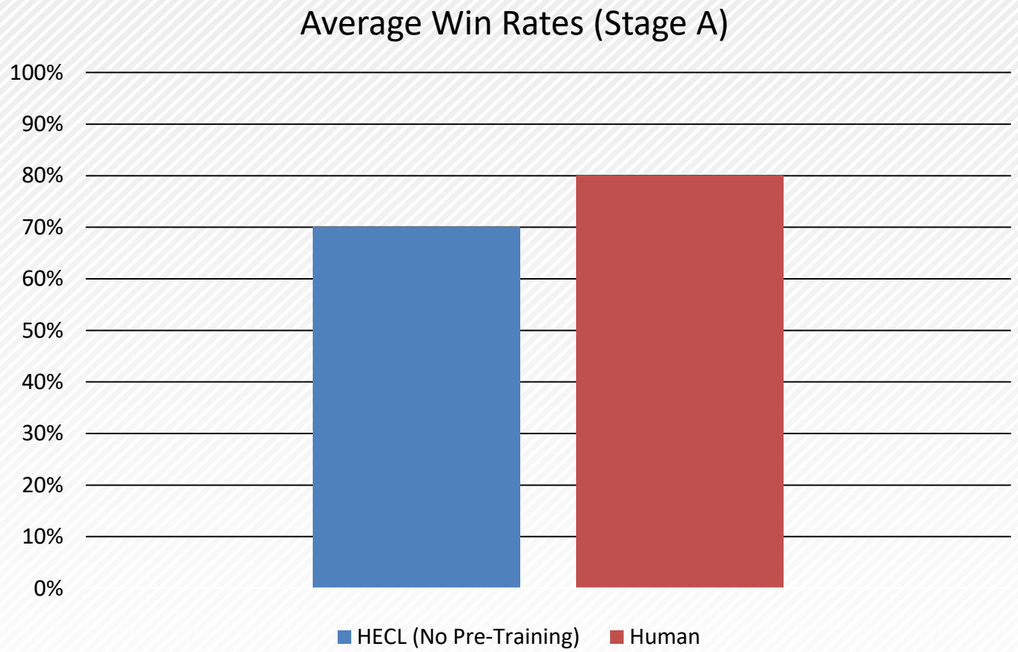
Stage C

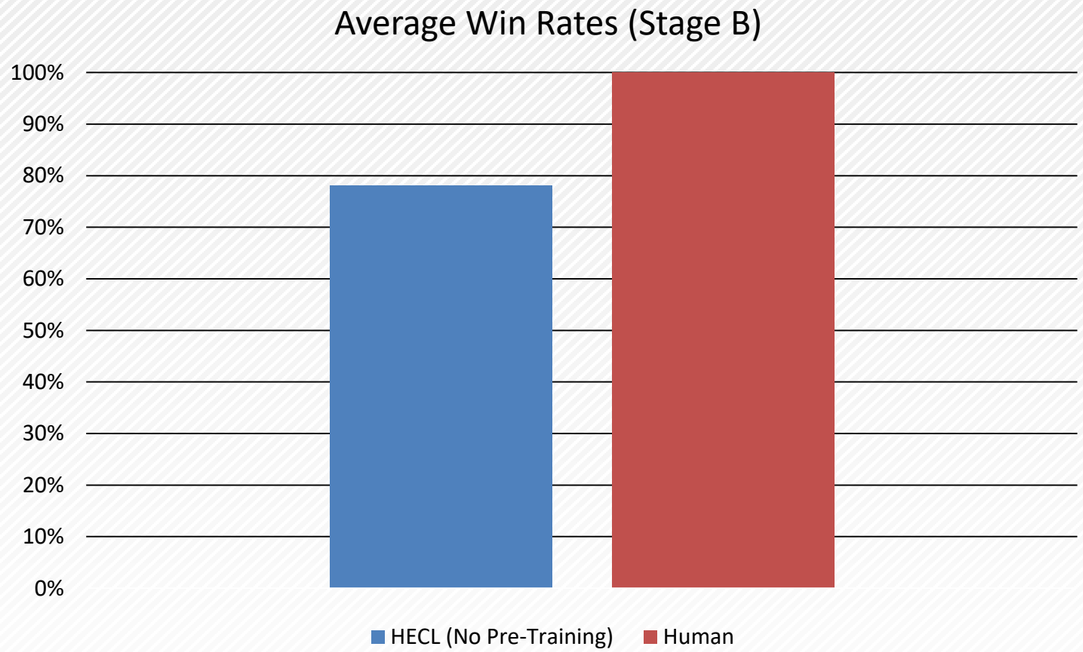


Difficulty:
Stage A > Stage B > Stage C

11 hours of training

AGAINST HUMAN EXPERT





HECL SPECIFICATION

Input space: game state (not pixels)

Number of agents: 32

Simulators: 32 simulators distributed between 4 PCs.

Batch size: 256 (rounds of battle)

GPU: NVIDIA P100x2 (total 32GB of memory)

Training time: 11 hours (1050 iterations)

Optimizer: RMSProp

BATCH SIZE COMPARISON

Bigger batch size tends to result in better stability

OTHER ALGORITHMS

World Model

- Hard to model the latent state

Imitation Learning

- Requires too much data for our case
- Would be useful for debugging-cases

Intrinsic Reward

- Could be future exploration
- Might not work well with dense reward situations

Meta Learning
Meta Gradient

- Recent works perform very well
- Might be weak on sparse-reward problems

VERY SLOW LEARNING

THE FIRST TRIAL

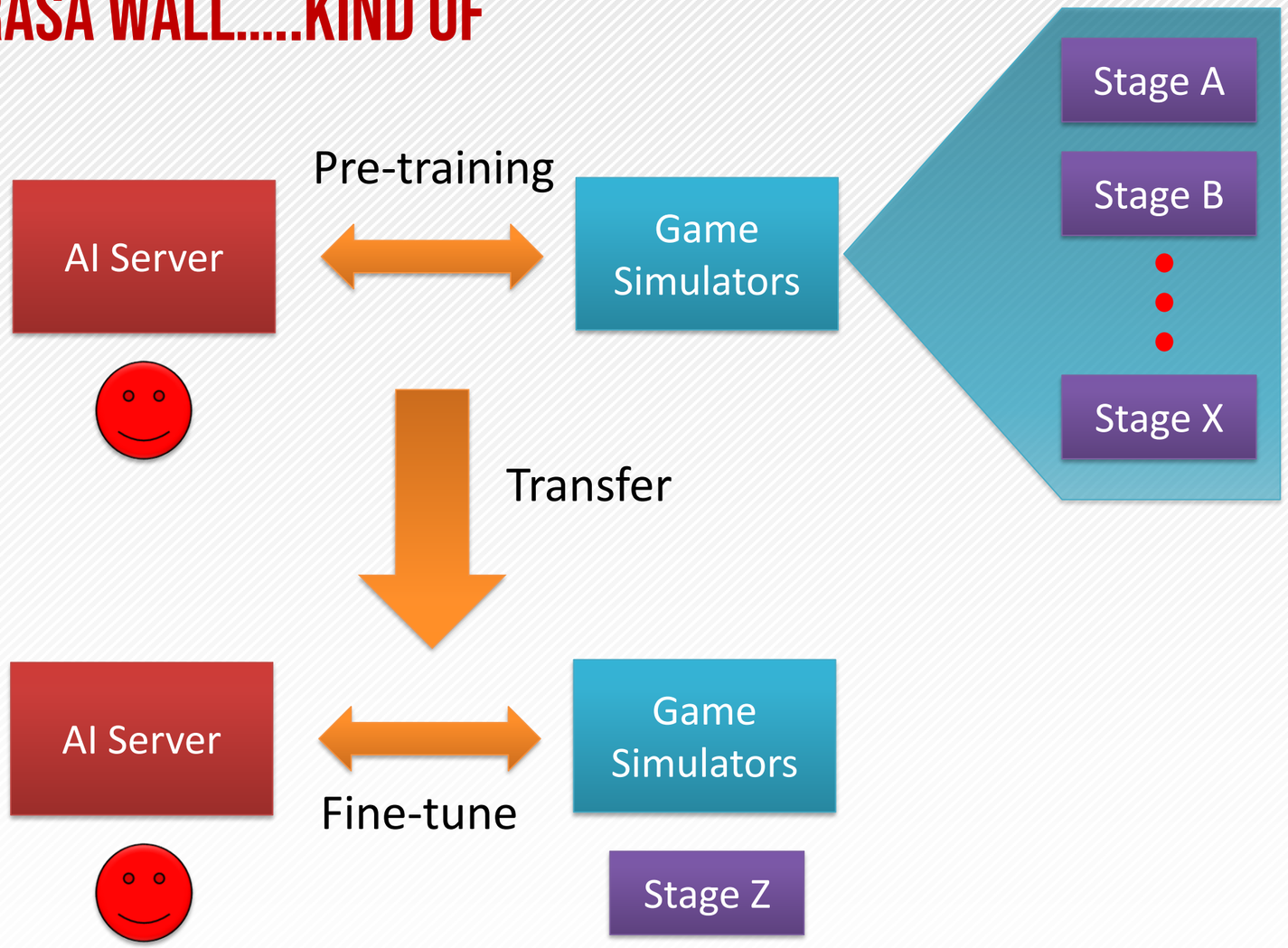
Train an AI on one stage

200 stages * 11 hours = 3 months!!!

TRANSFER LEARNING

TRANSFER LEARNING

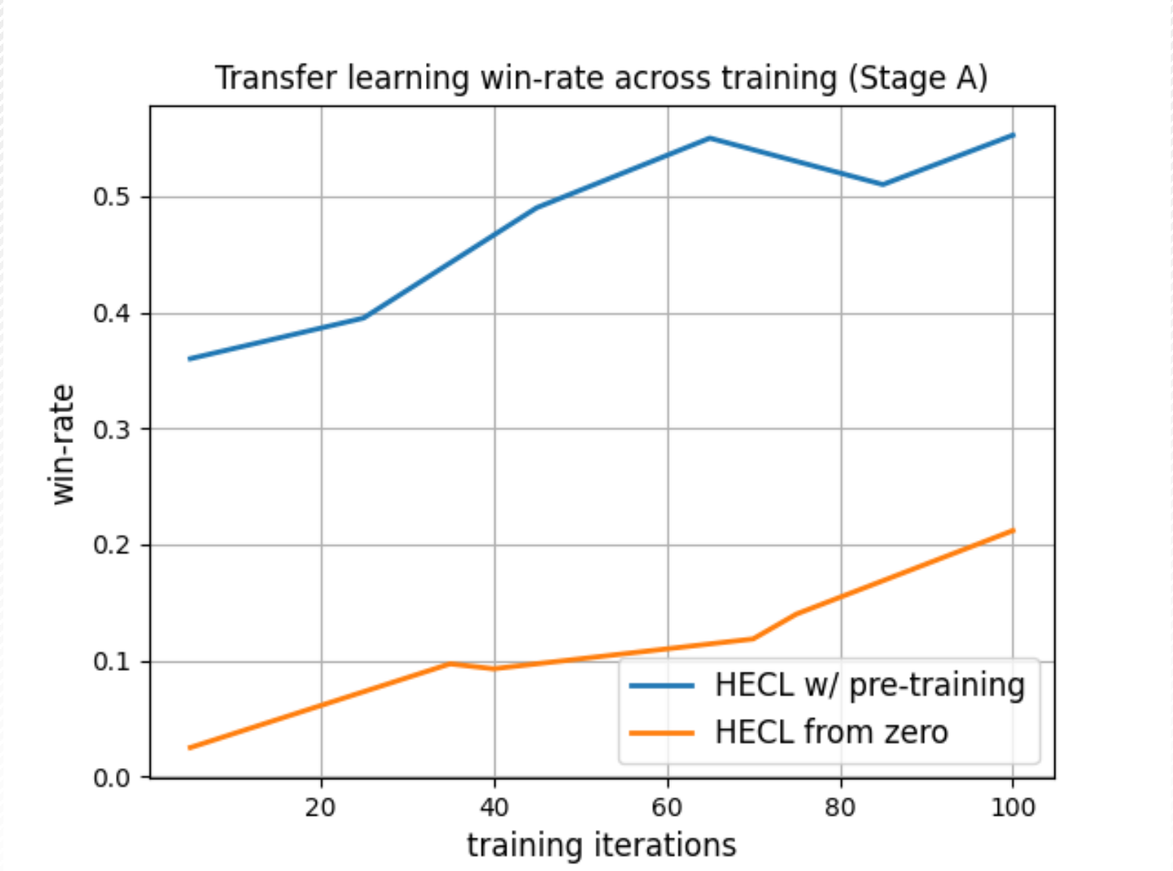
BREAKING THE TABULA RASA WALL.....KIND OF



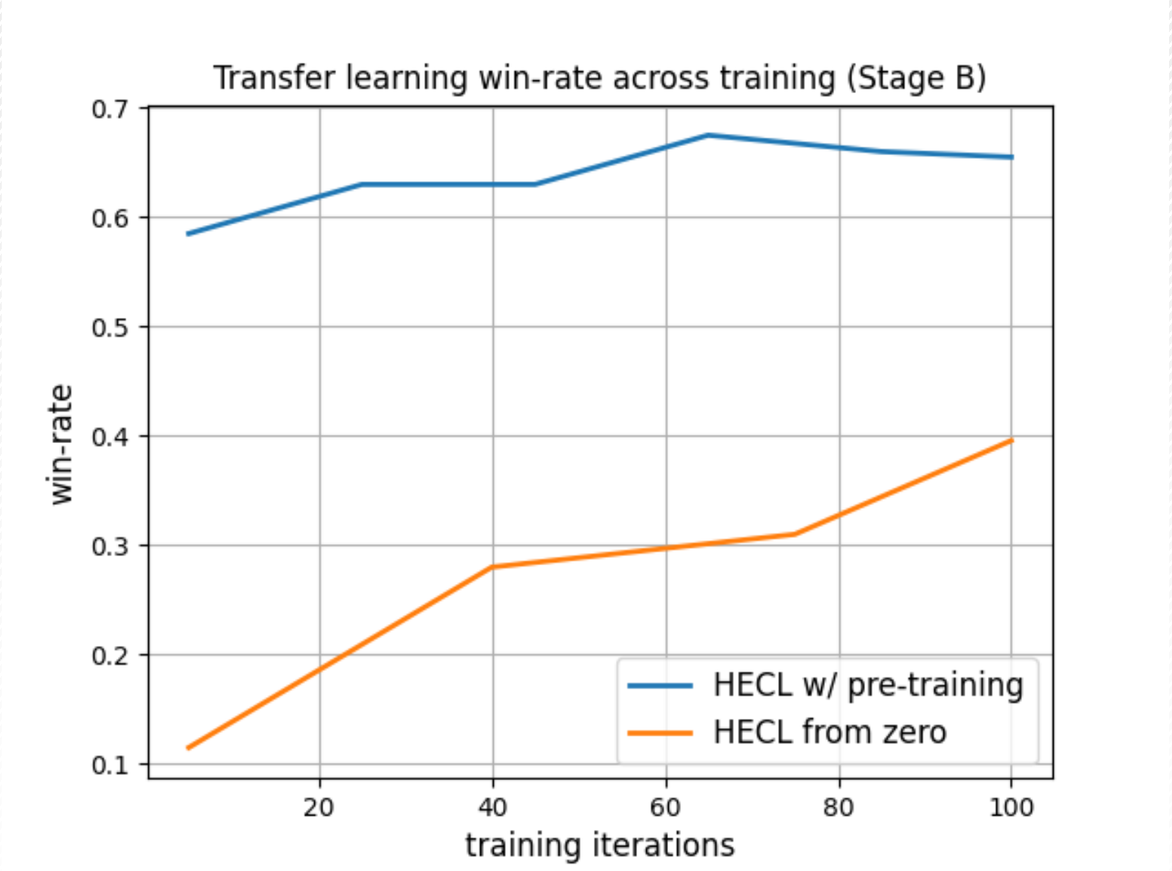
TRANSFER LEARNING PERFORMANCE

11 hours down to 1.5 hours

Stage A



Stage B



CURRENT ISSUES

- How to communicate game patches appropriately
- When to do pre-training again?
- The best time to do fine-tuning again after changing a stage parameters

THE FUTURE

- RL for playthrough debugging
- Implementing it to other large-scale games.

AGENDA

- Background
- Basic of Reinforcement Learning (RL)
- Challenges
- RL Algorithm
- Engineering

REVIEW

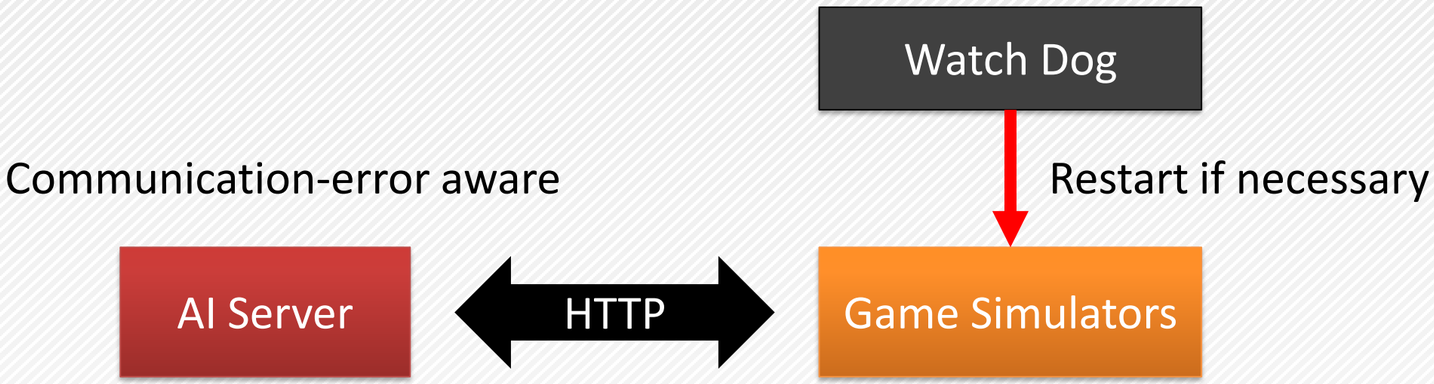
Challenges:

- Unstable simulator (frequent crashes and freeze)

- Game constantly updated: affects pre-training

SYSTEM ARCHITECTURE

ADDRESSING THE FREQUENT CRASHES AND FREEZE



What happens on restart:

- Restart causes battle progress to be truncated, causing communication-error.
- Truncated battle is automatically thrown away.
- LSTM hidden state is reset.

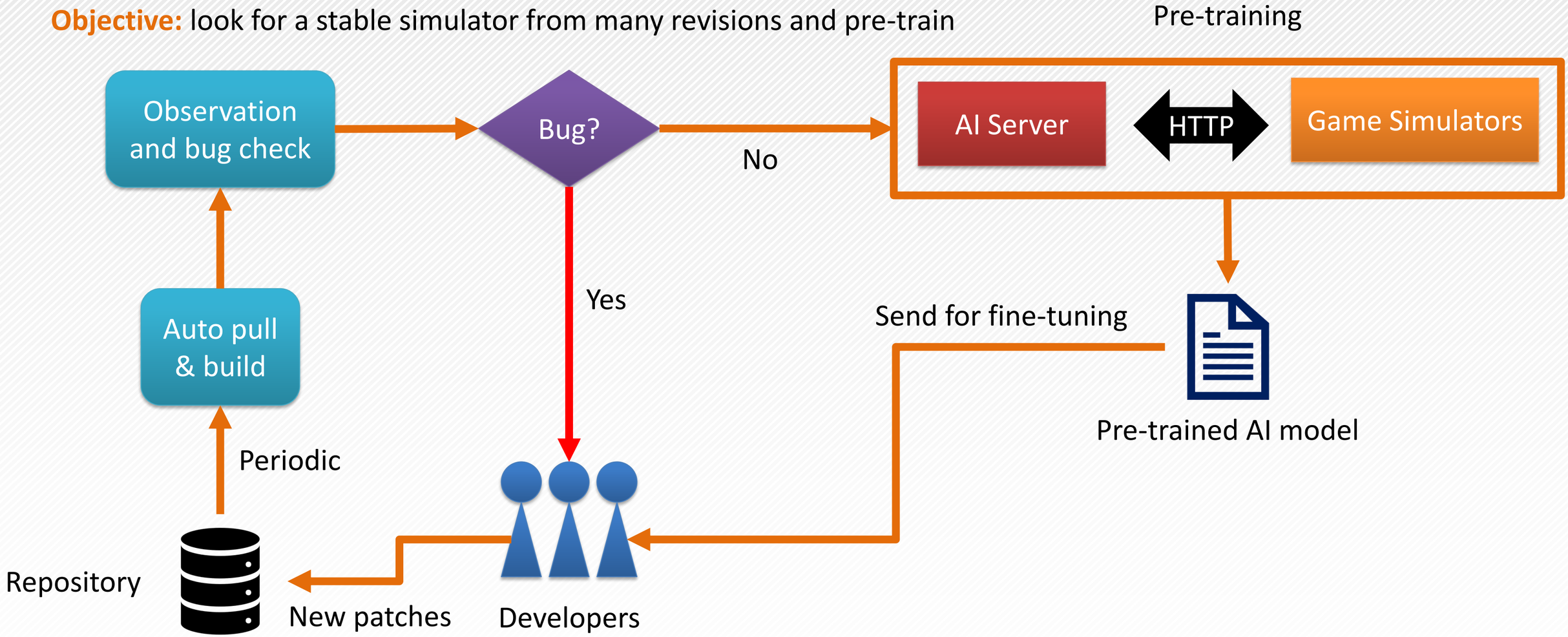
Tolerance-level:

- As long as we can clear the battles.

SYSTEM ARCHITECTURE

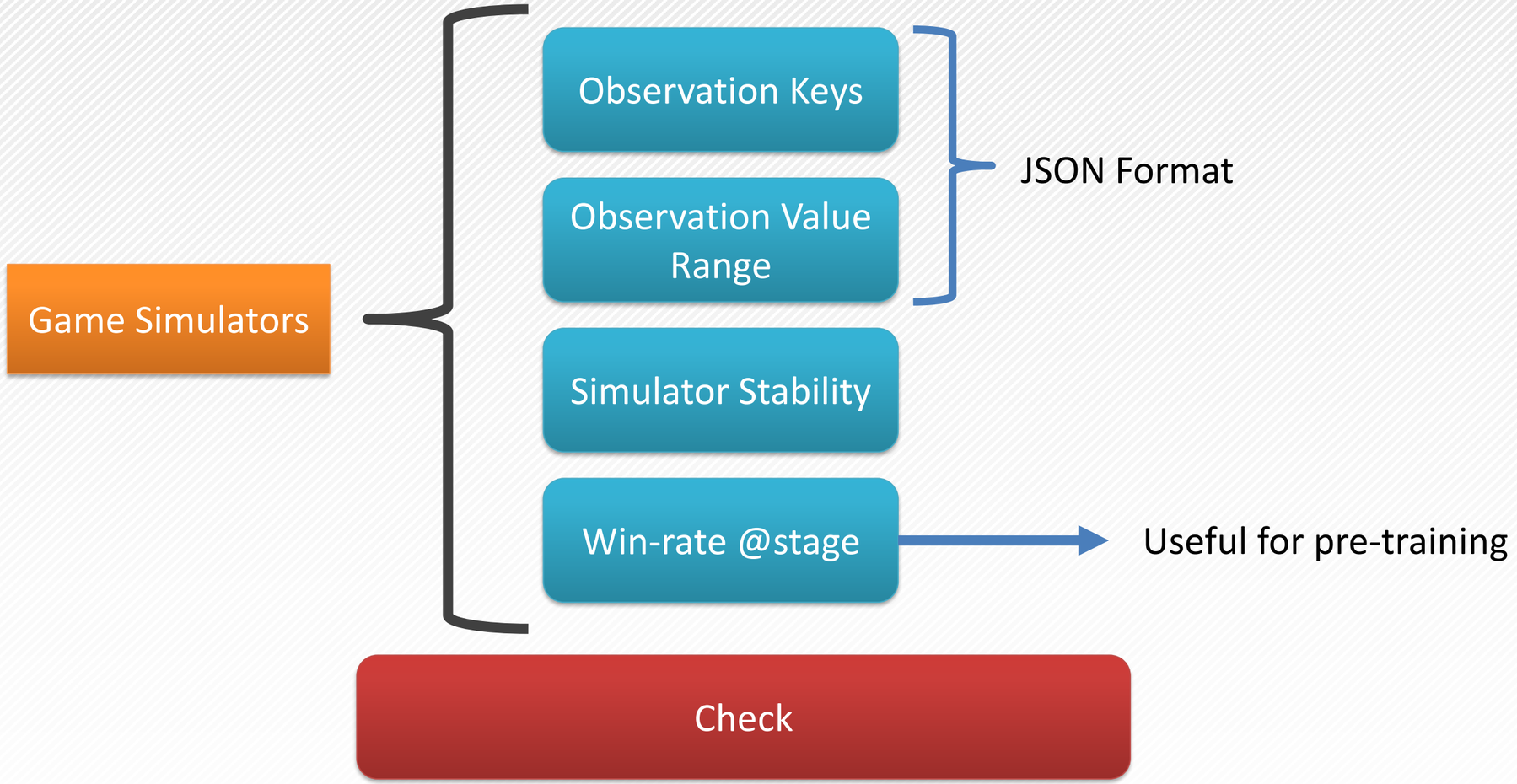
CONSTANTLY CHANGING SIMULATORS

Objective: look for a stable simulator from many revisions and pre-train



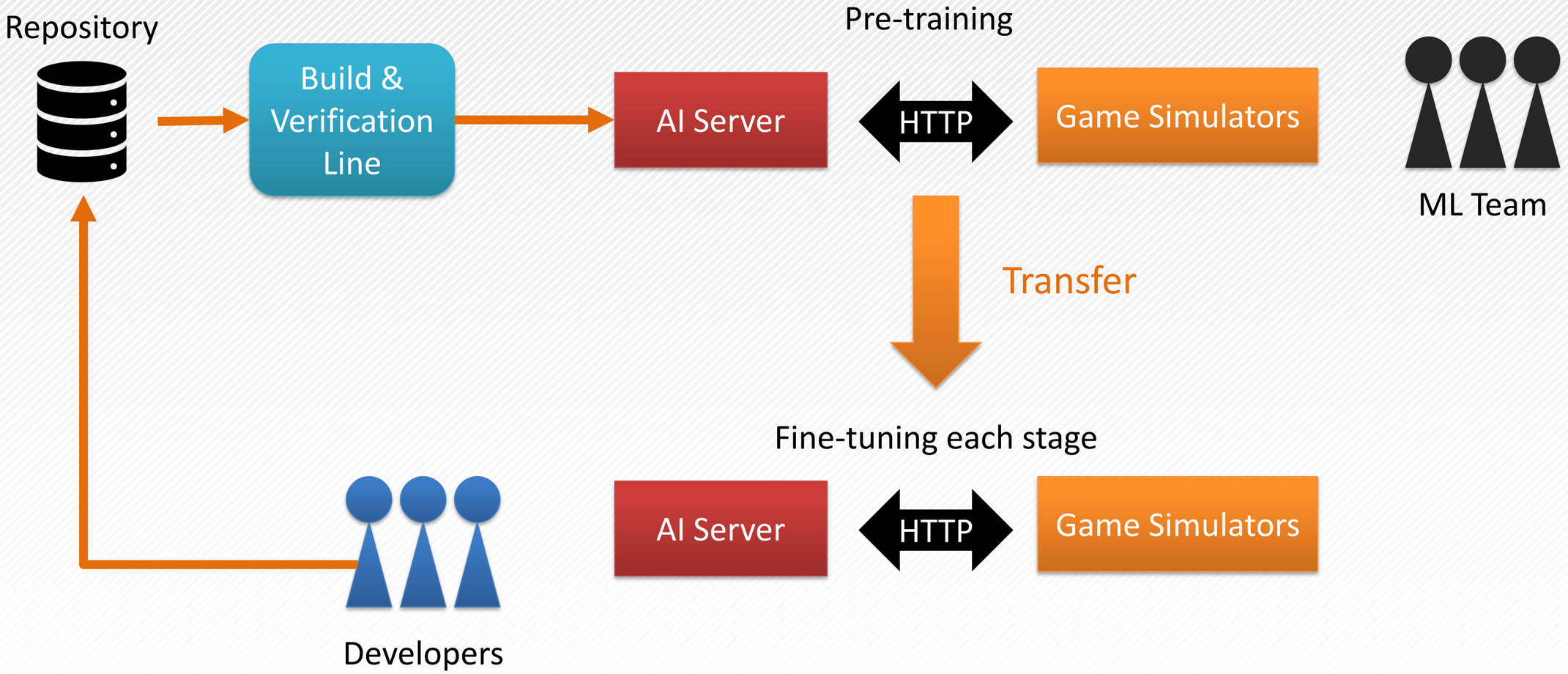
SYSTEM ARCHITECTURE

OBSERVATION/BUG CHECK



SYSTEM ARCHITECTURE

PRE-TRAINING TO FINE-TUNING

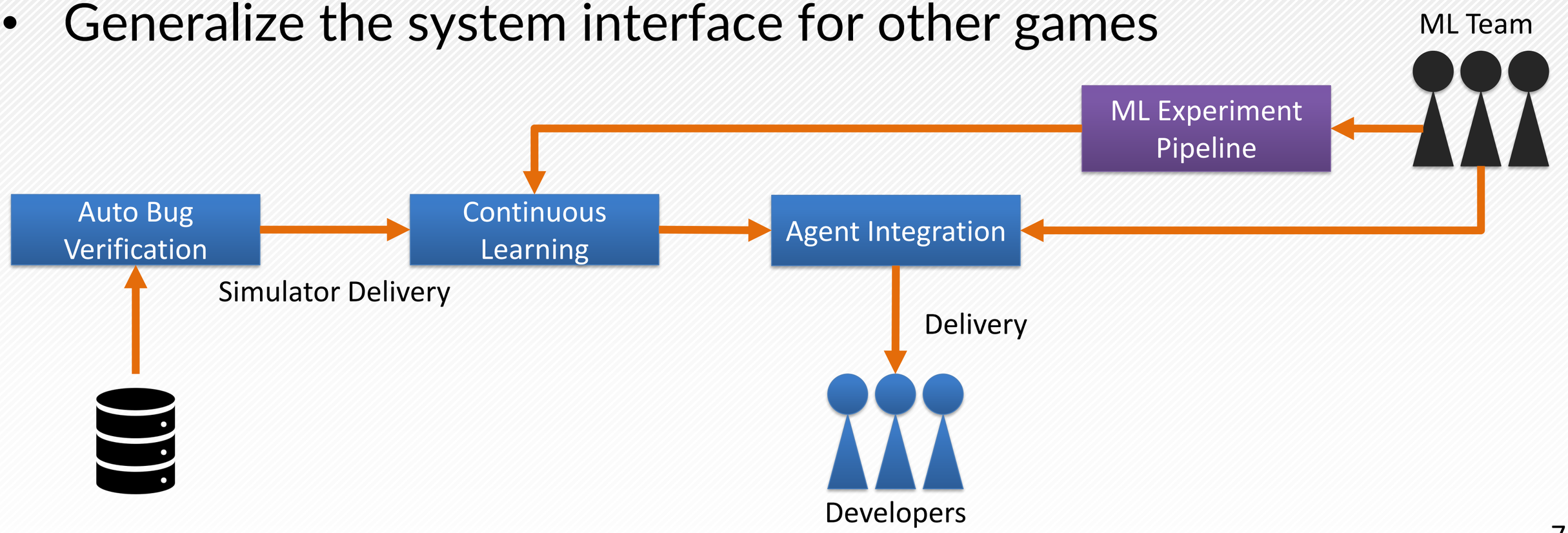


CURRENT ISSUES

- Not all observation bugs are detectable automatically
  - Data assignment problems
  - Visual data
- There is no general rule of thumb for the win-rate threshold
- 100% headless simulator is hard to be implemented
- System integration in the early stage of game development

THE FUTURE

- Expanding to RLOps (in-progress)
  - CL/CI/CD
- Generalize the system interface for other games



DISCUSSION W/ GAME DESIGNERS

Bug found

Core mechanics adjustments

New strategy discovery

Reward function design

AI strategy ≈ human (in some cases)

THANK YOU

Contact:
Email: hanedgar@square-enix.com
LinkedIn: edgar_handy
Twitter: @edgar_handy

AI Division. Founded on 2022.
Team Size: ~15 AI Experts
Research Focus:

- Academic & applied research on AI and Machine Learning for game experience and development

REFERENCES

Dota 2 is a trademark or registered trademark of Valve Corporation.

Wings of Liberty and StarCraft are trademarks or registered trademarks of Blizzard Entertainment, Inc. in the U.S. and/or other countries.

Gran Turismo and Gran Turismo Sophy are trademarks or registered trademarks of Sony Interactive Entertainment Inc.

All other trademarks are the property of their respective owners