

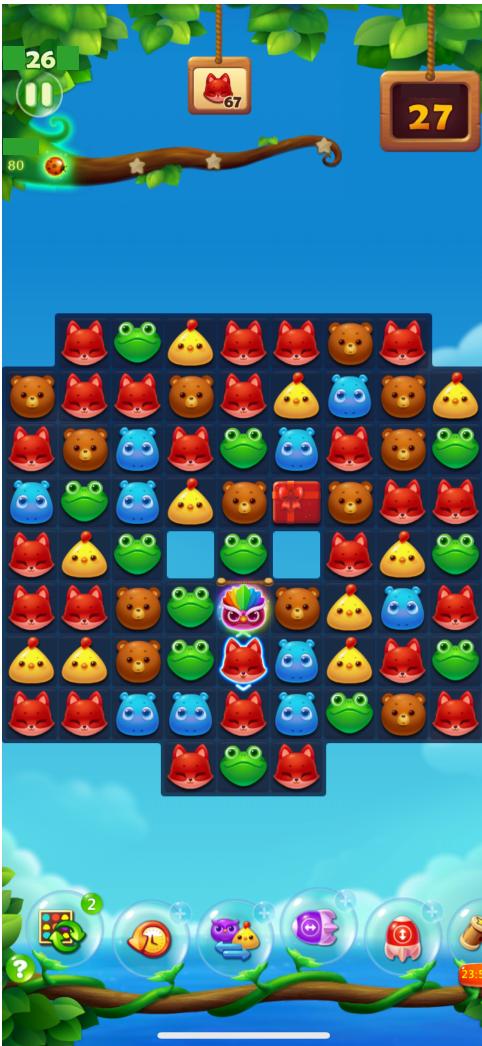


**MAKE
THE
WORLD
HAPPY**

Generating the Best Game Experience through AI

Rein Houthooft

Introduction

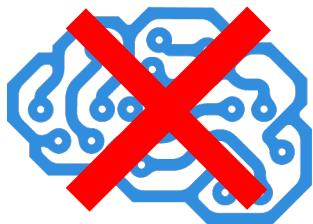


- Company focus: **casual** mobile games
- Main product: “**Anipop**” 
- Extremely popular (>**100M** users/month)
- Generates **TBs** of data each day
- **AI Lab** founded 1 year ago

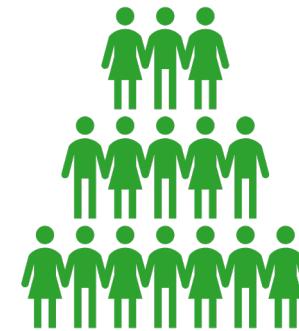


AI Lab: Goals & Strategy

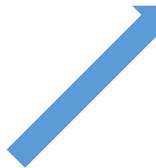
Open-loop to Closed-loop Game Design



Sparse feedback



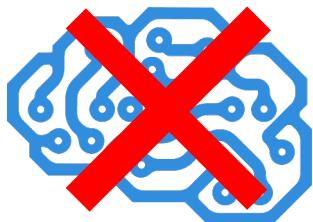
Little Optimization



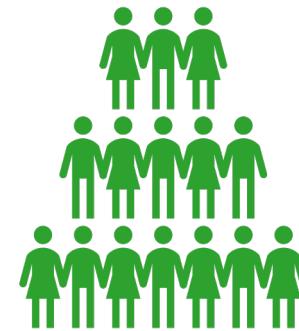
Open-loop to Closed-loop Game Design

- Product-player preferences mismatch
- Designers slow in adapting to changing player behavior
- Lower player satisfaction

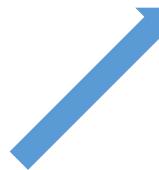
Open-loop to Closed-loop Game Design



Dense feedback



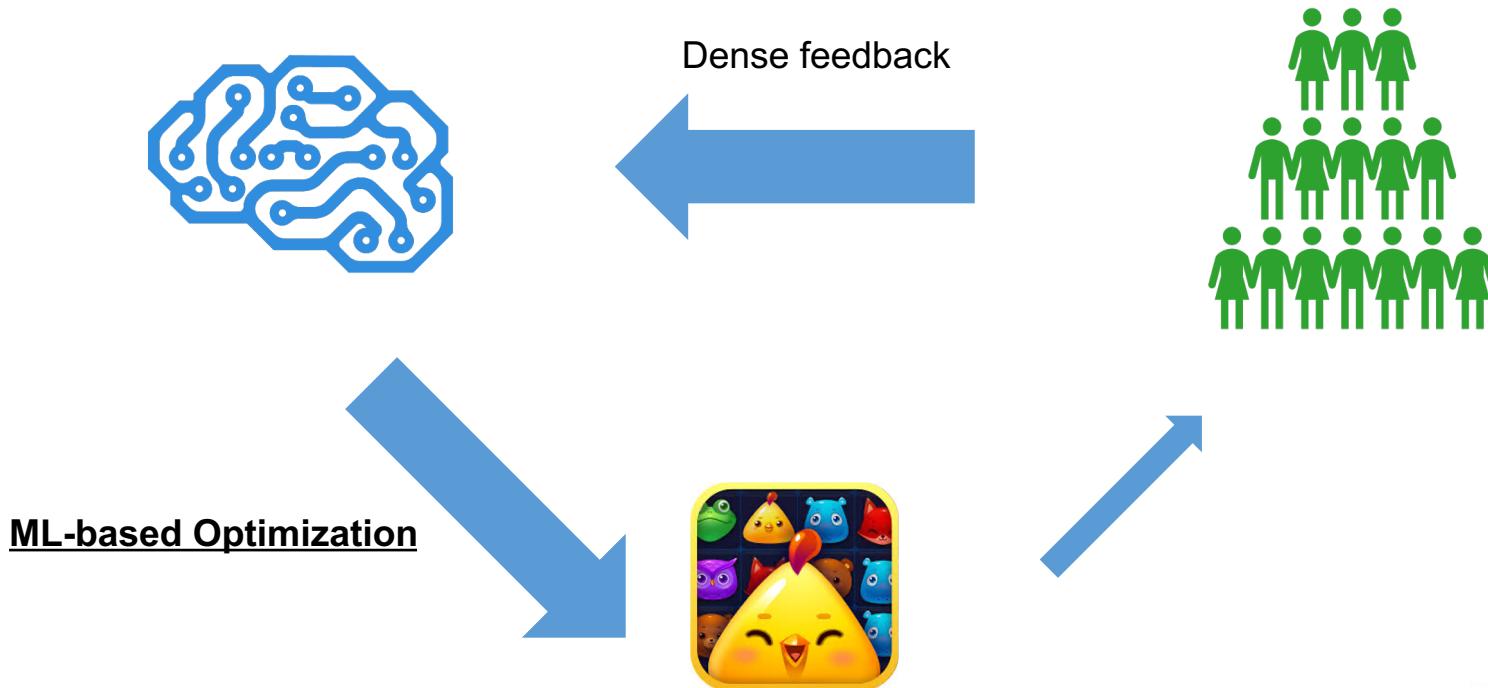
Rule-based Optimization



Open-loop to Closed-loop Game Design

- Some adaptation to changing player preferences
- Low granularity
- Hard to maintain over time

Open-loop to Closed-loop Game Design



Open-loop to Closed-loop Game Design

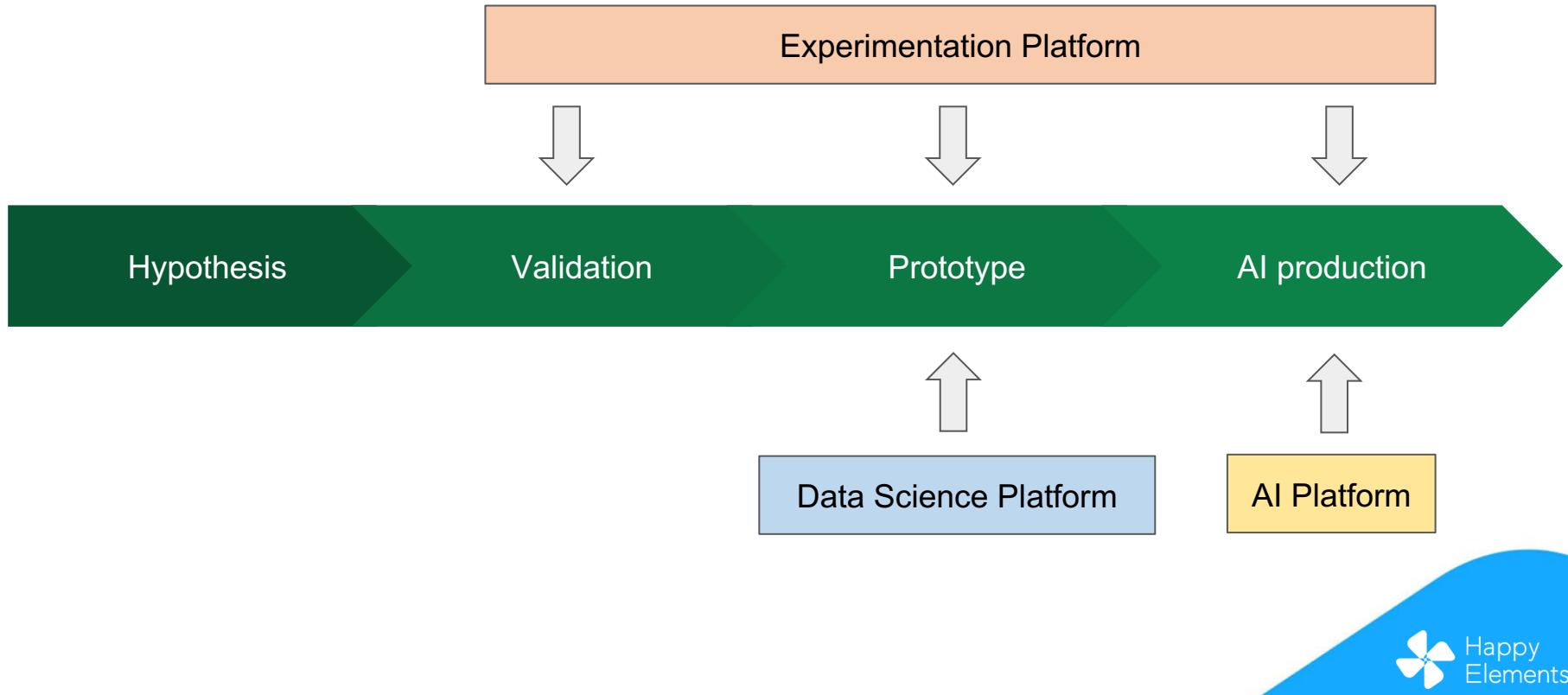
- Immediate adaptation to changing player preferences
- High granularity
- Maintains itself through objective function optimization



Use Case:

Deep Learning for Game Difficulty Adjustment

From Hypothesis to Production



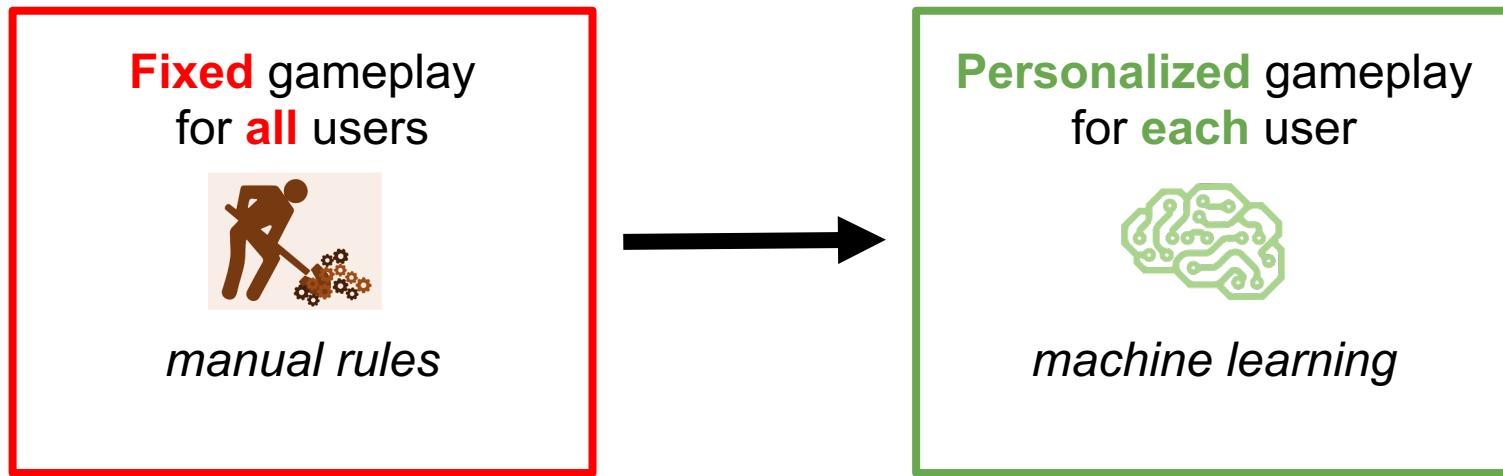
Game Difficulty Adjustment

- Hypothesis: preferred difficulty varies across users & time.
- Validation: difficulty correlates with LTV/retention.
- Prototype + Production: adjust difficulty dynamically via ML.

Problem Formulation

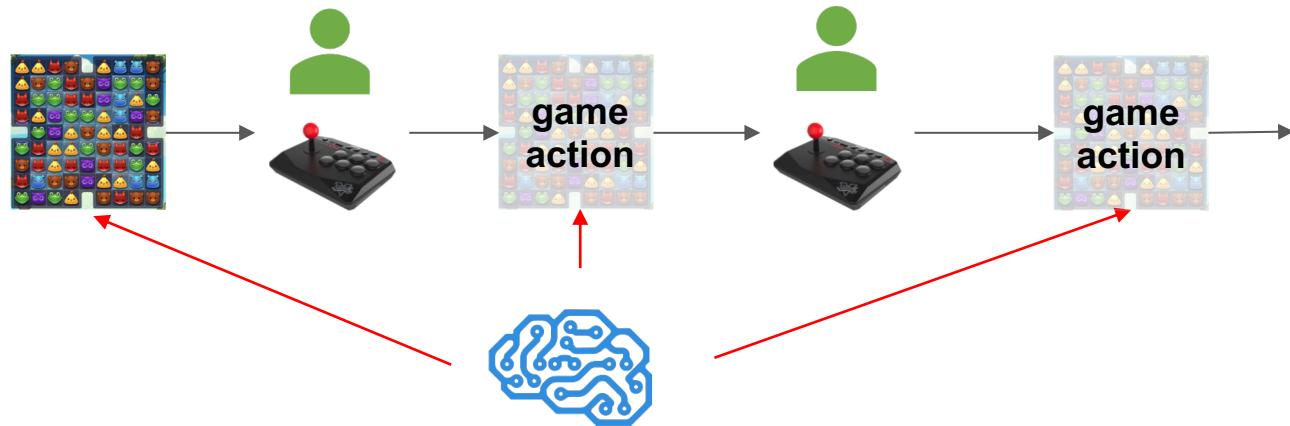


Problem Formulation



Problem Formulation

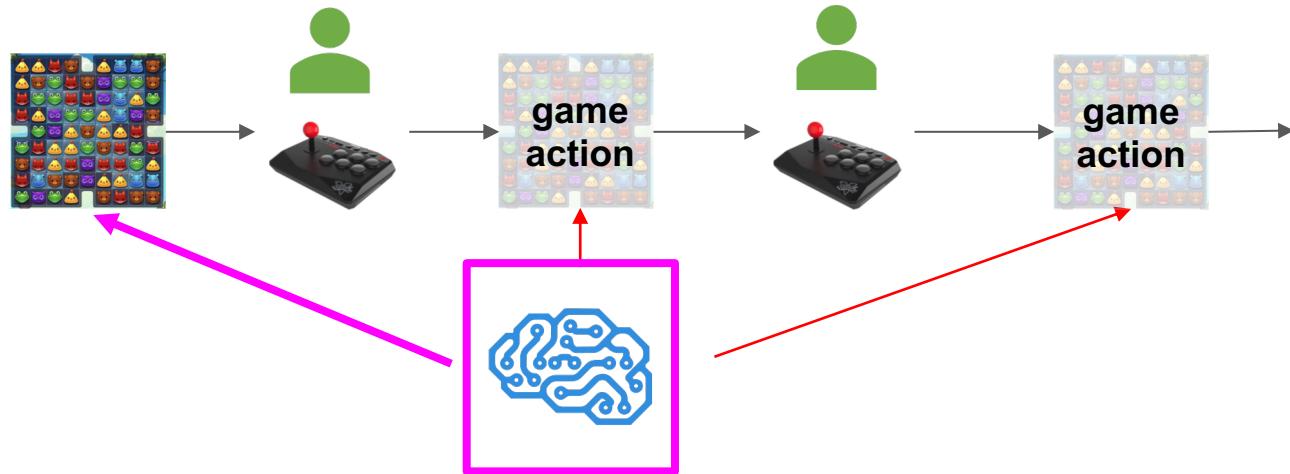
Gameplay modification: action sequences



Objective: Rewards = player LTV/retention

Problem Formulation

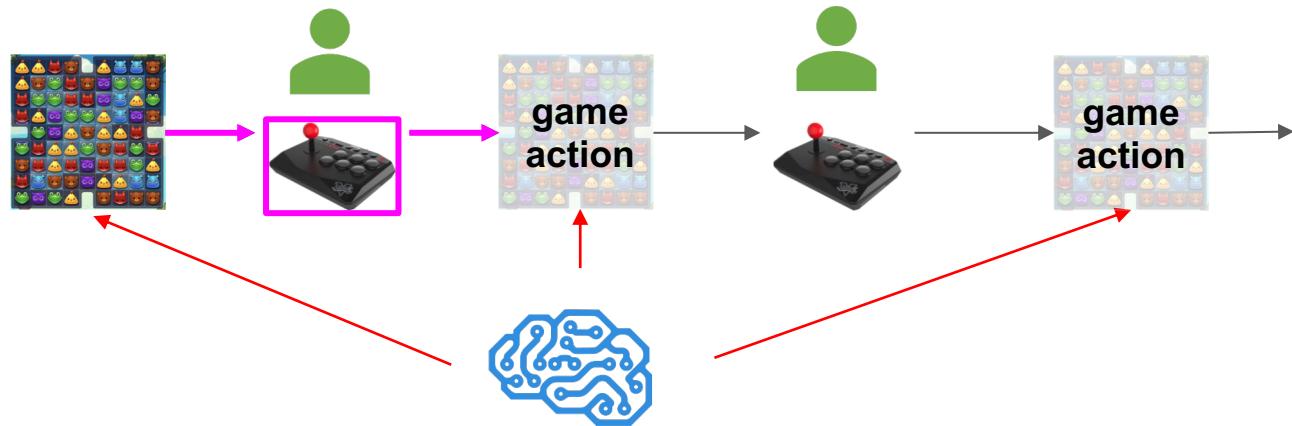
Gameplay modification: action sequences



Objective: Rewards = player LTV/retention

Problem Formulation

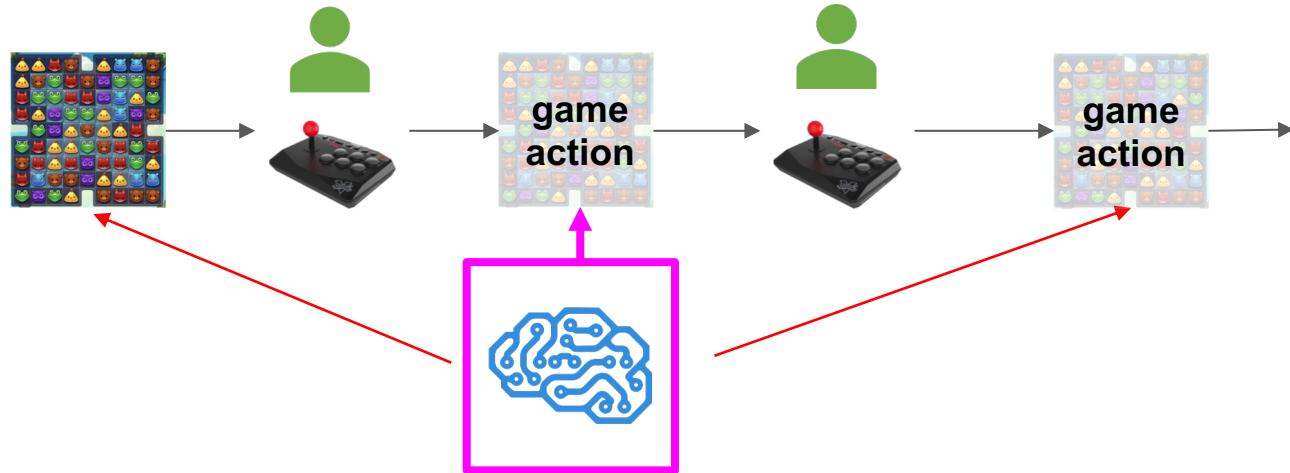
Gameplay modification: action sequences



Objective: Rewards = player LTV/retention

Problem Formulation

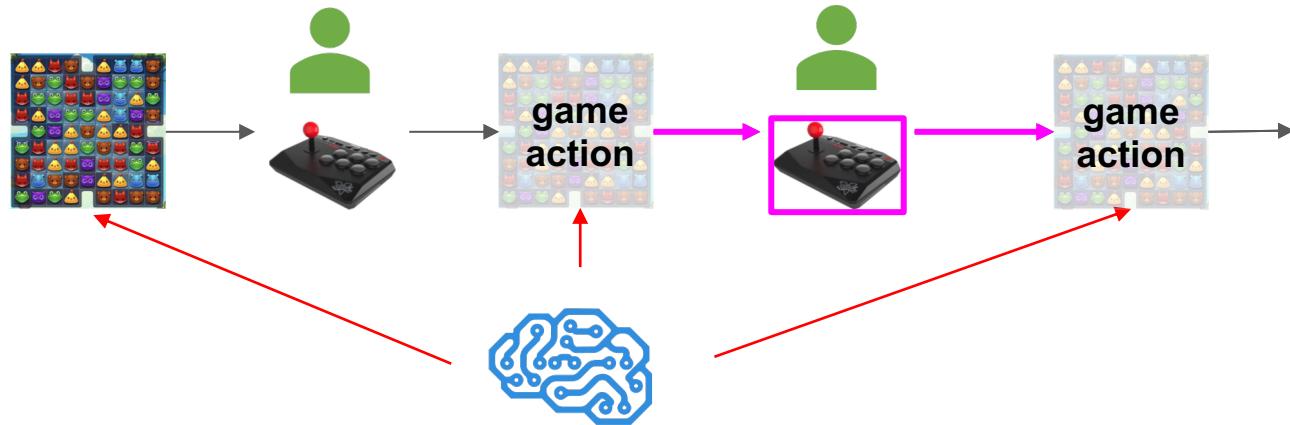
Gameplay modification: action sequences



Objective: Rewards = player LTV/retention

Problem Formulation

Gameplay modification: action sequences

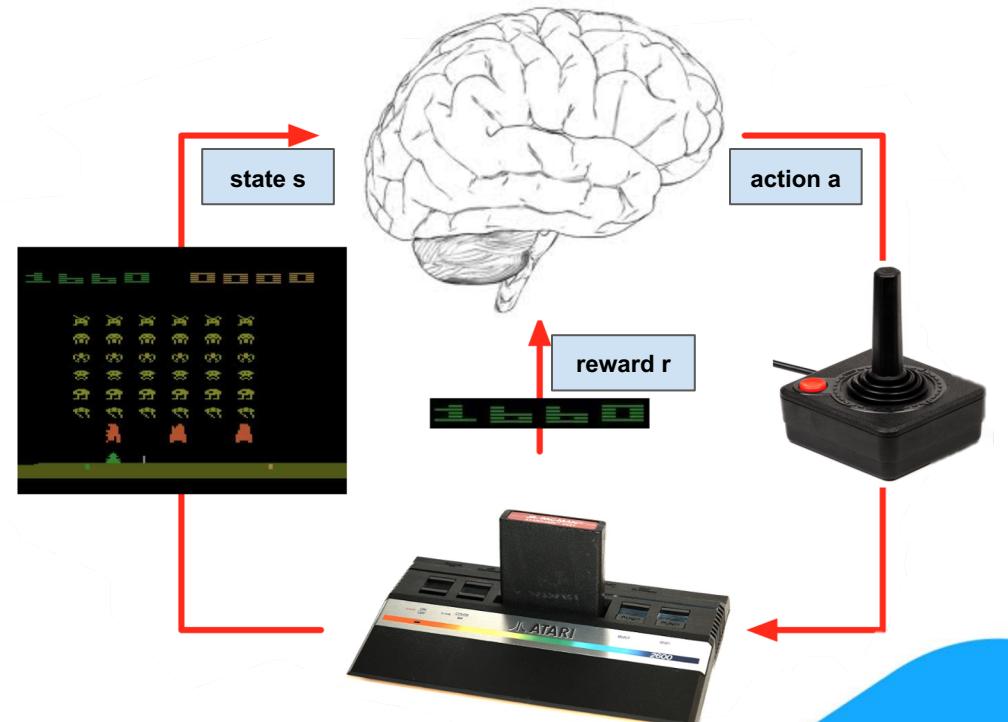


Objective: rewards = player LTV/retention

Problem Formulation

Reinforcement Learning:

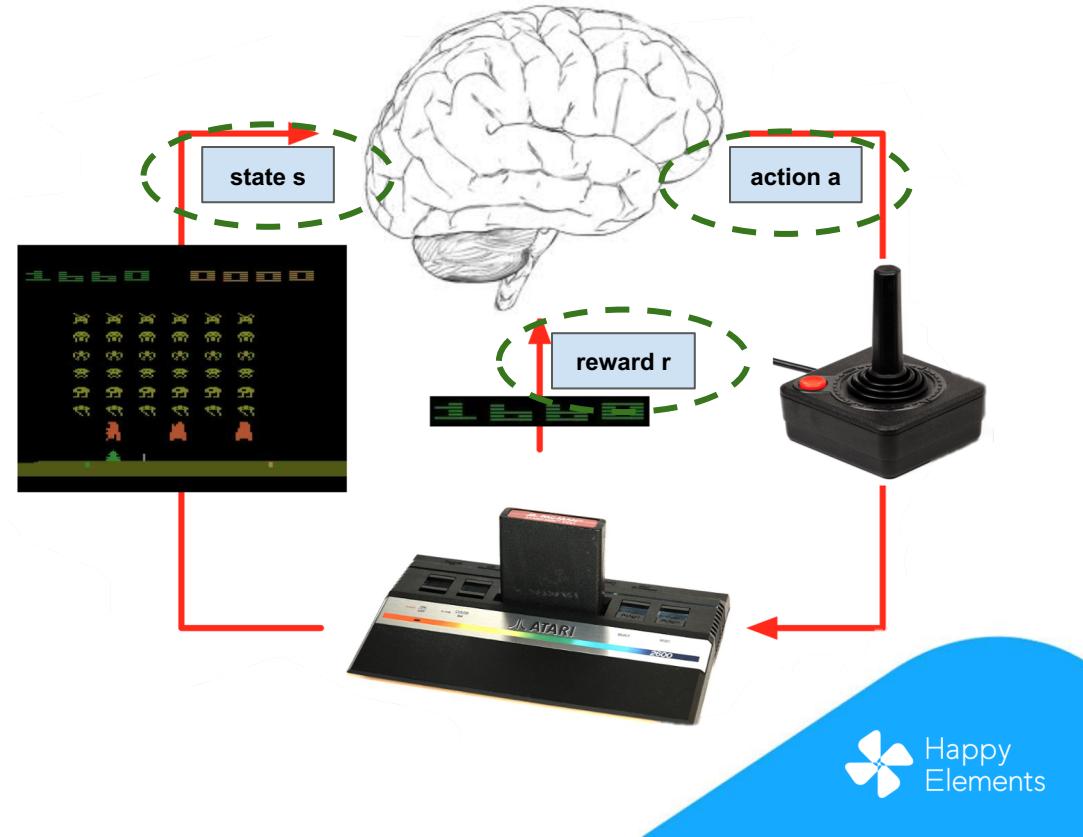
- What **action a** to take in **state s** to **optimize** the expected **reward $E[r]$** ?
- For example, video game:
 - state s = screen
 - action a = controller
 - reward r = score



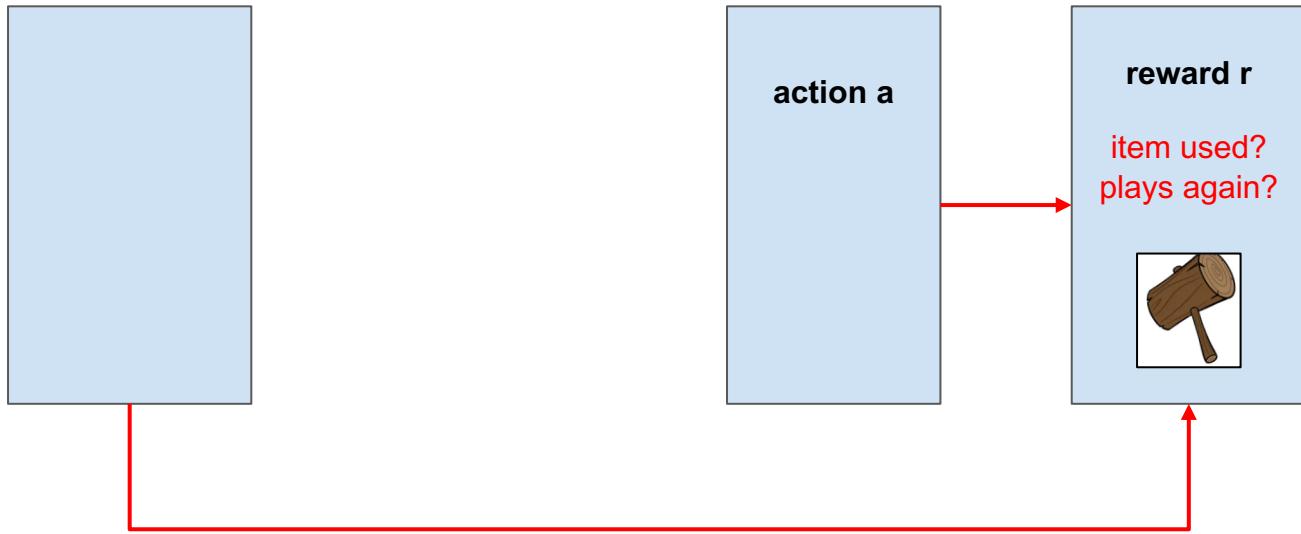
Problem Formulation

Reinforcement Learning:

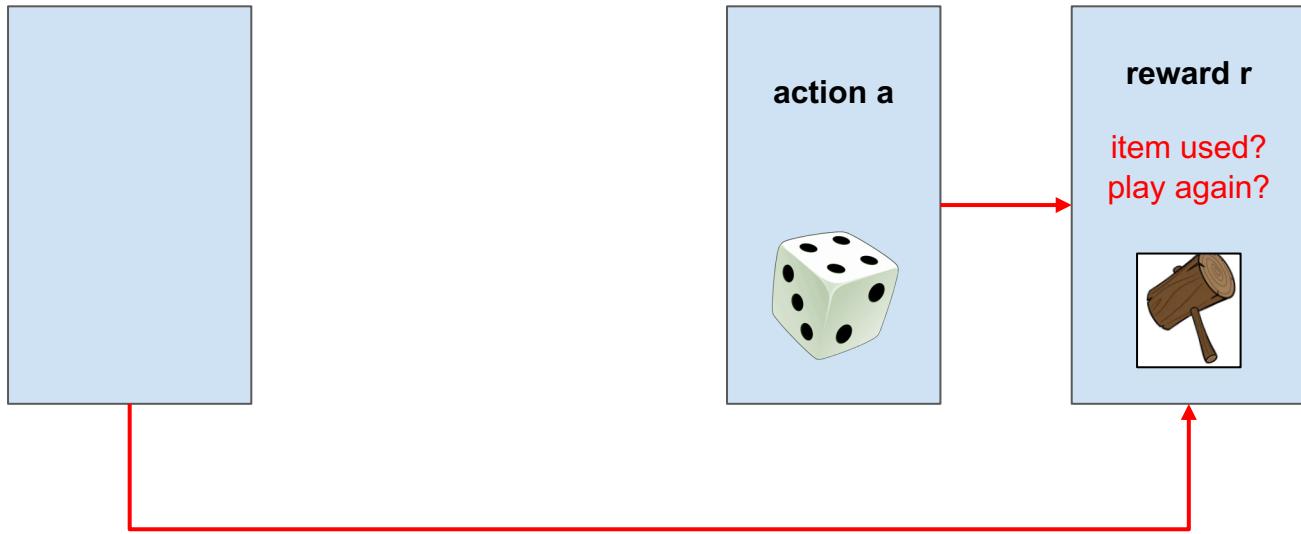
- What **action a** to take in **state s** to **optimize** the expected **reward $E[r]$** ?
- For example, video game:
 - state s = screen
 - action a = controller
 - reward r = score



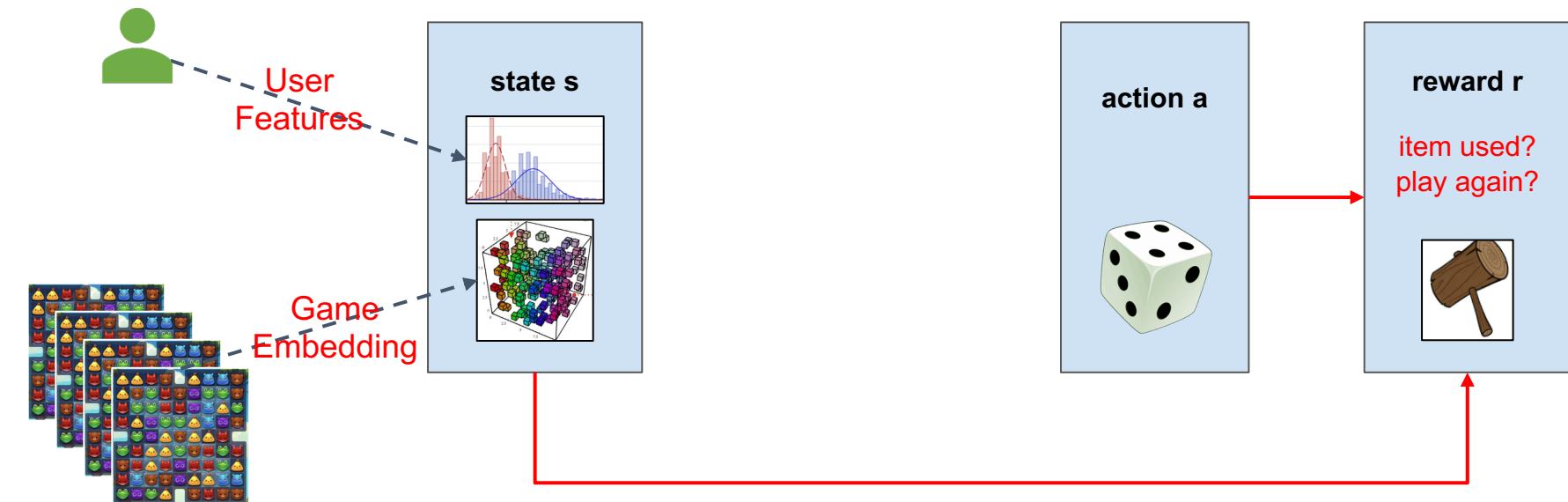
Problem Formulation



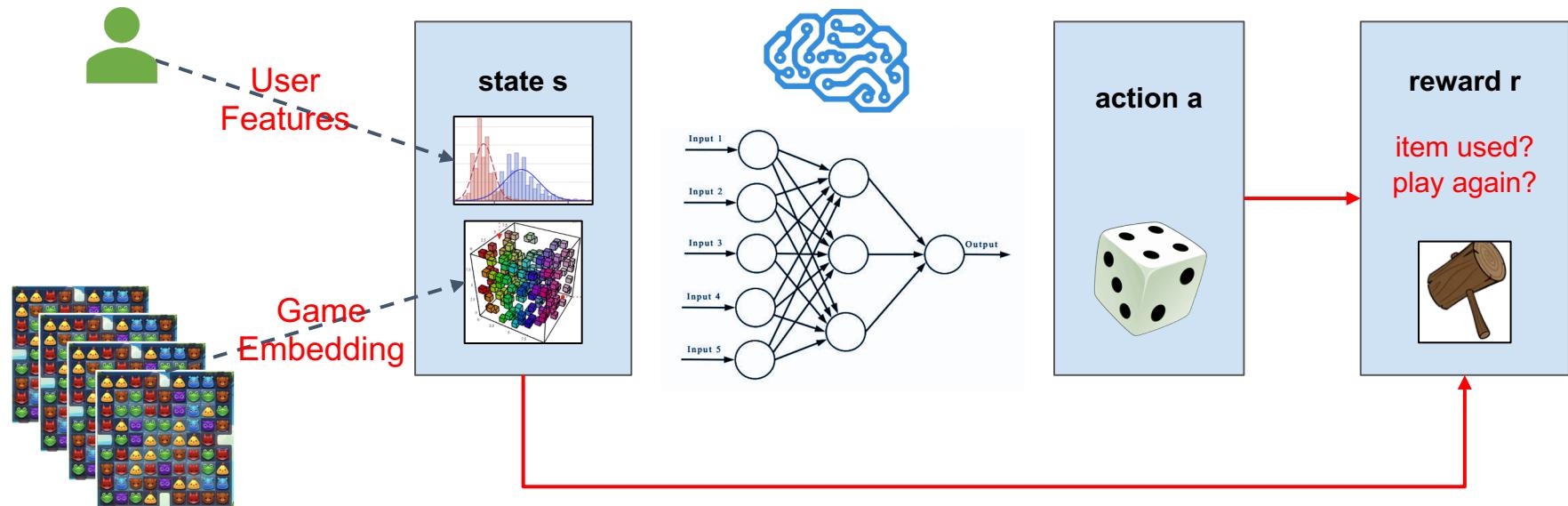
Problem Formulation



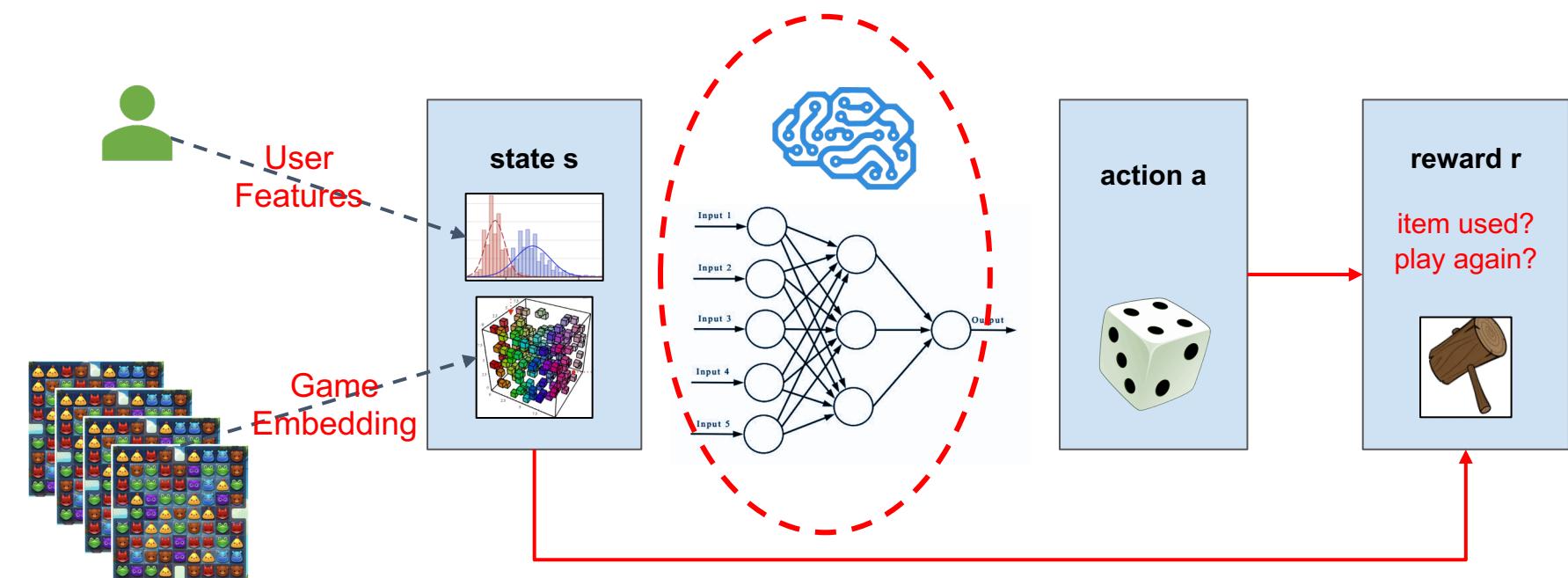
Problem Formulation



Problem Formulation

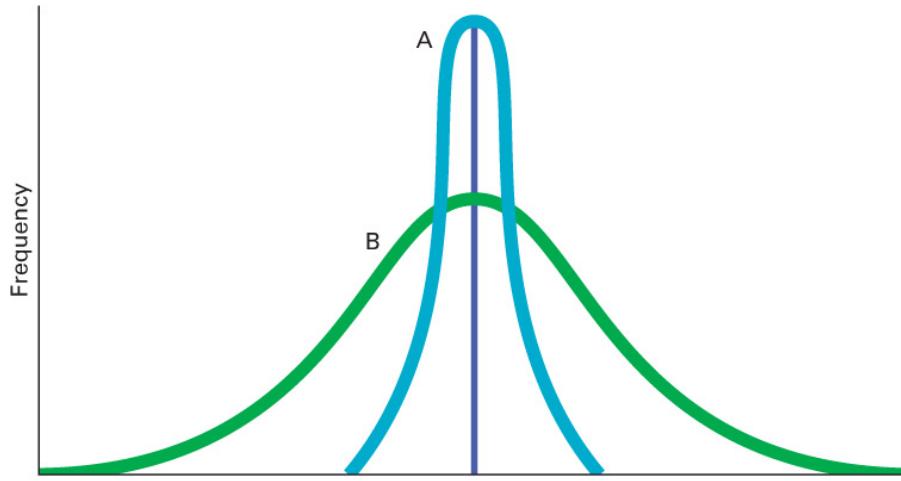


Problem Formulation

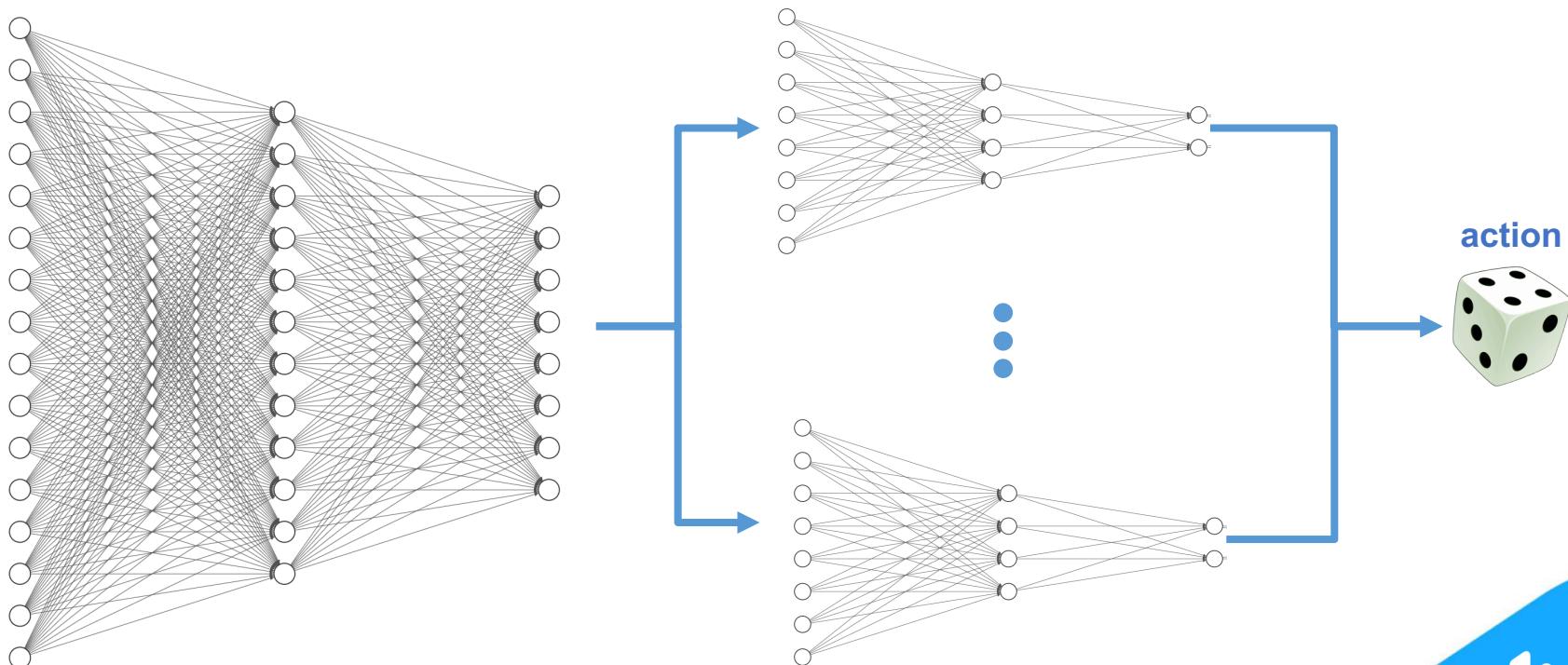


Model

- Based on deep RL method “Bootstrapped DQN” [Osband et al. 2016]
- Approximate Bayesian approach to capture model uncertainty.
- Solving exploitation (A) vs. exploration (B) problem.

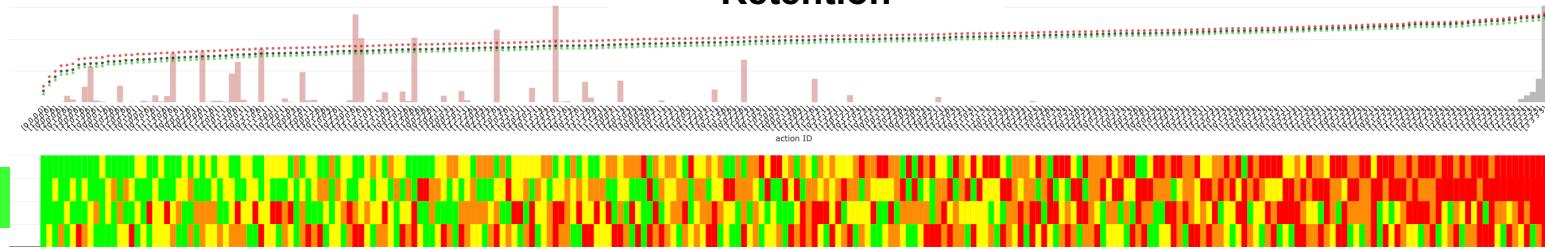


Model: Bootstrapped Contextual Bandits



Model: Patterns Learned

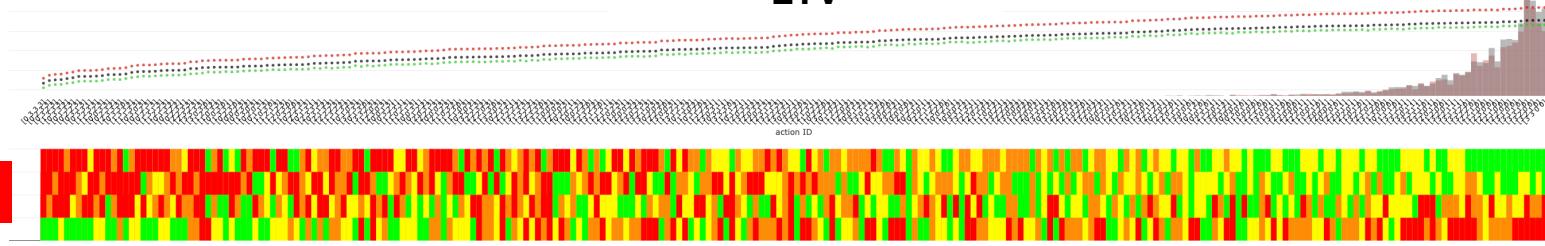
Retention



HARD

EASY

LTV

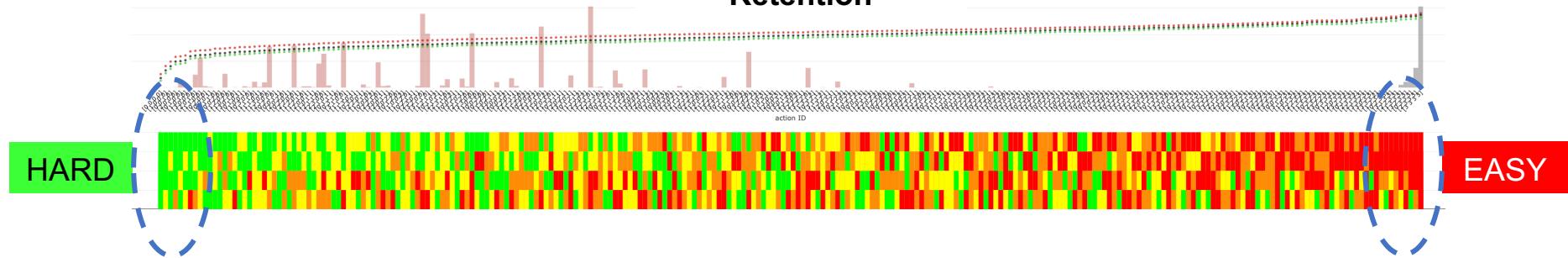


EASY

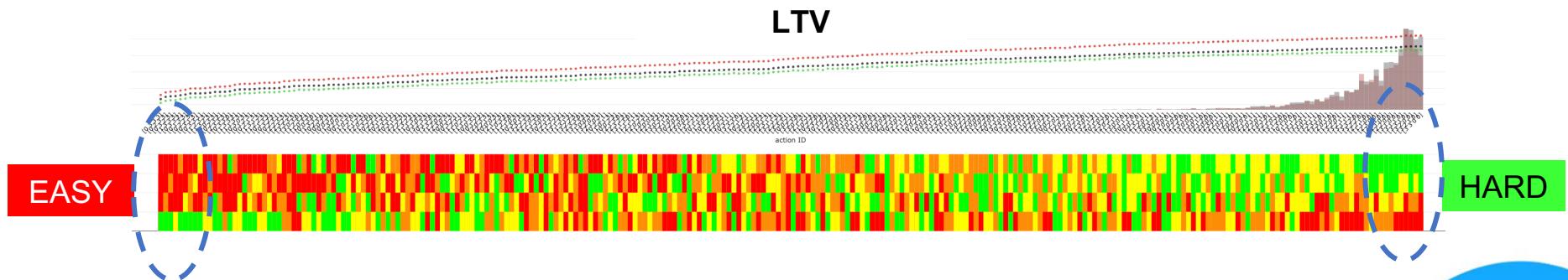
HARD

Model: Patterns Learned

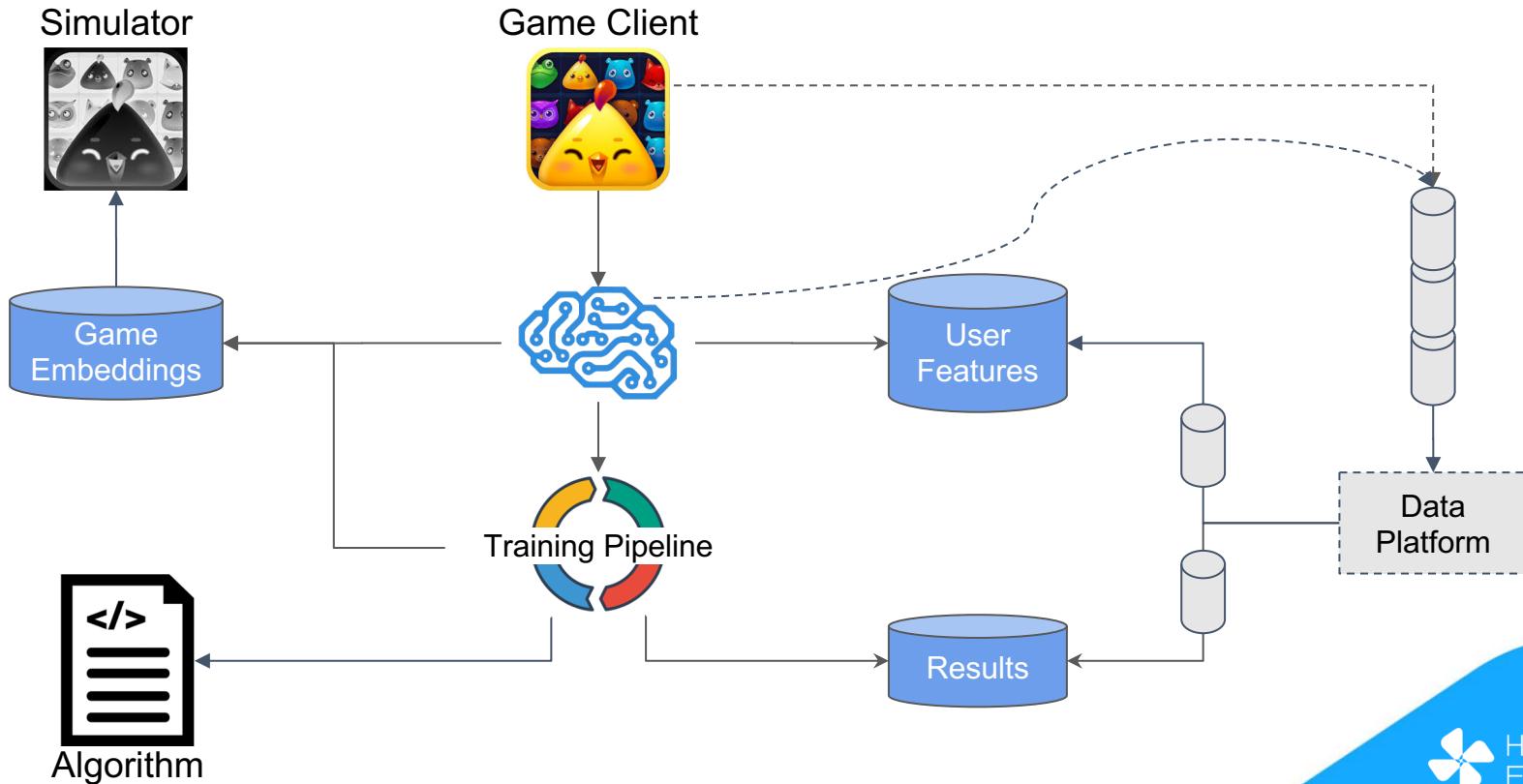
Retention



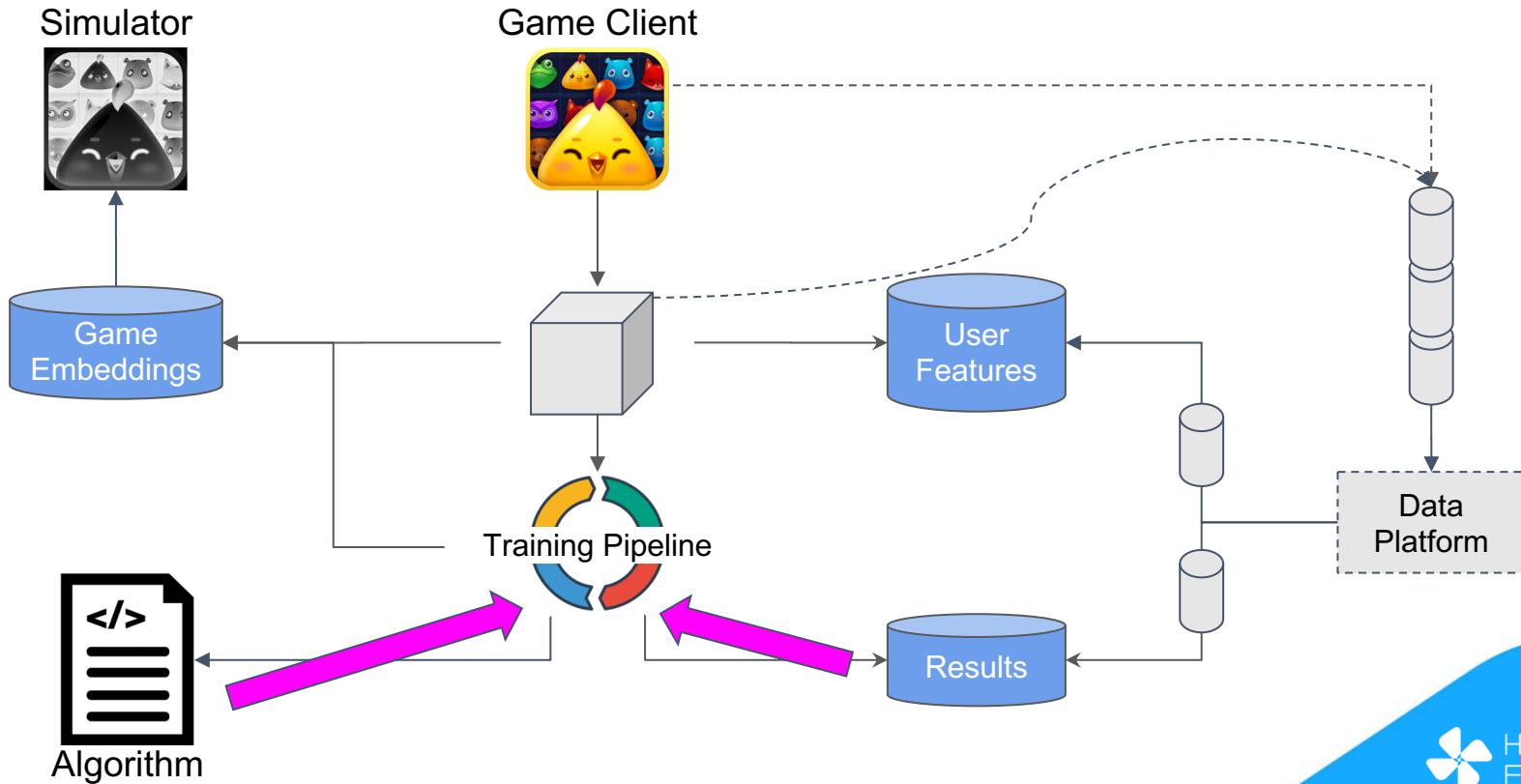
LTV



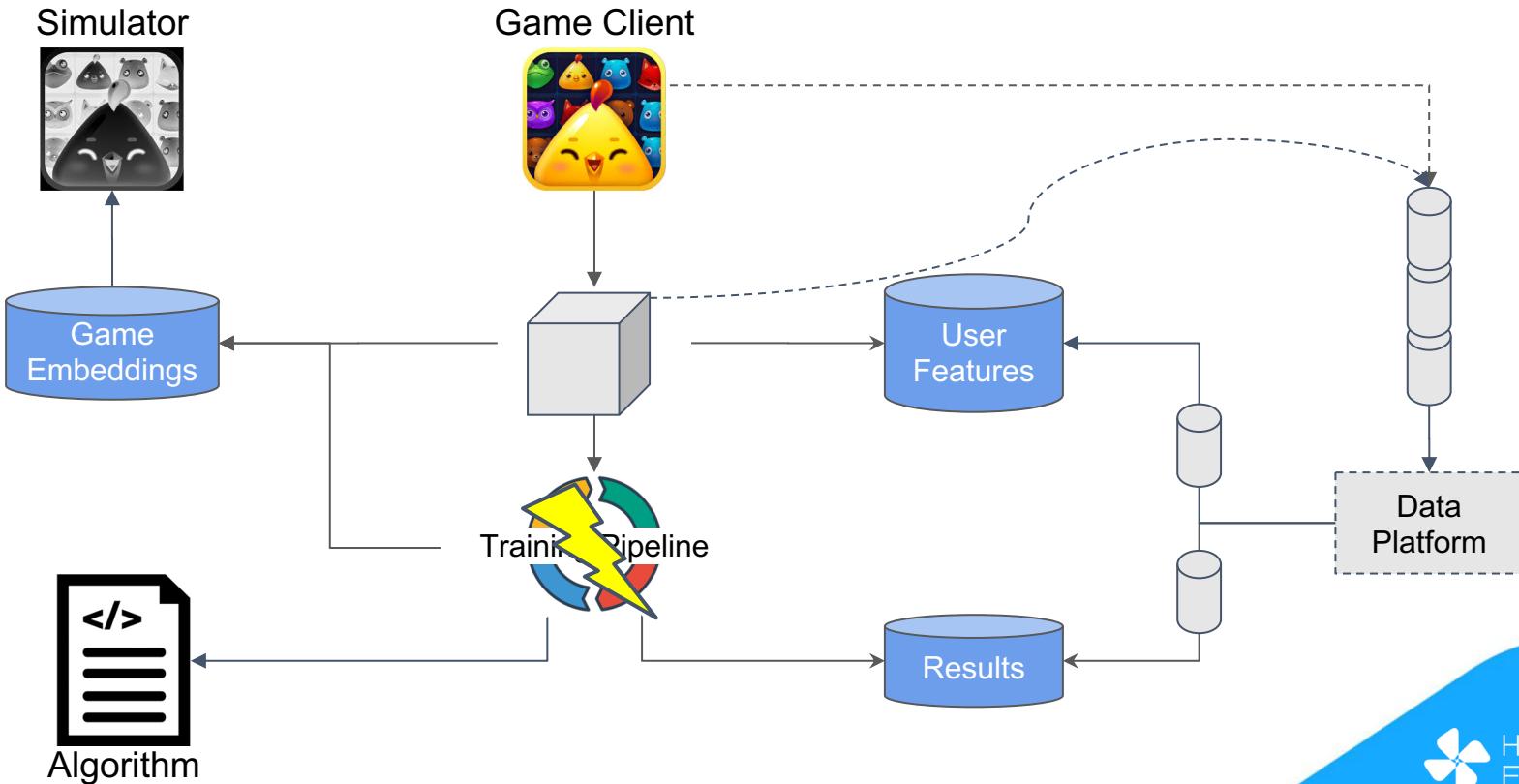
System Architecture



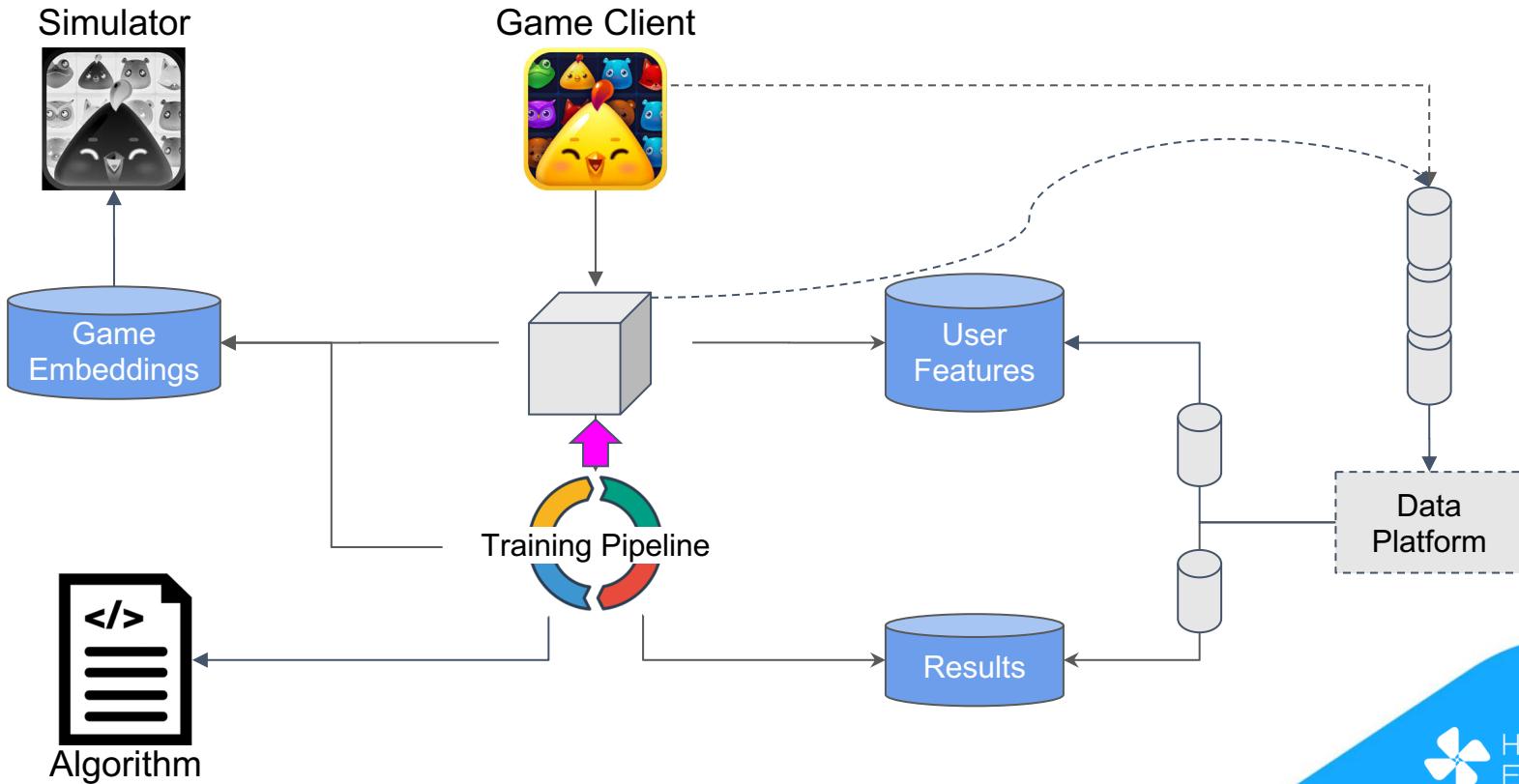
System Architecture



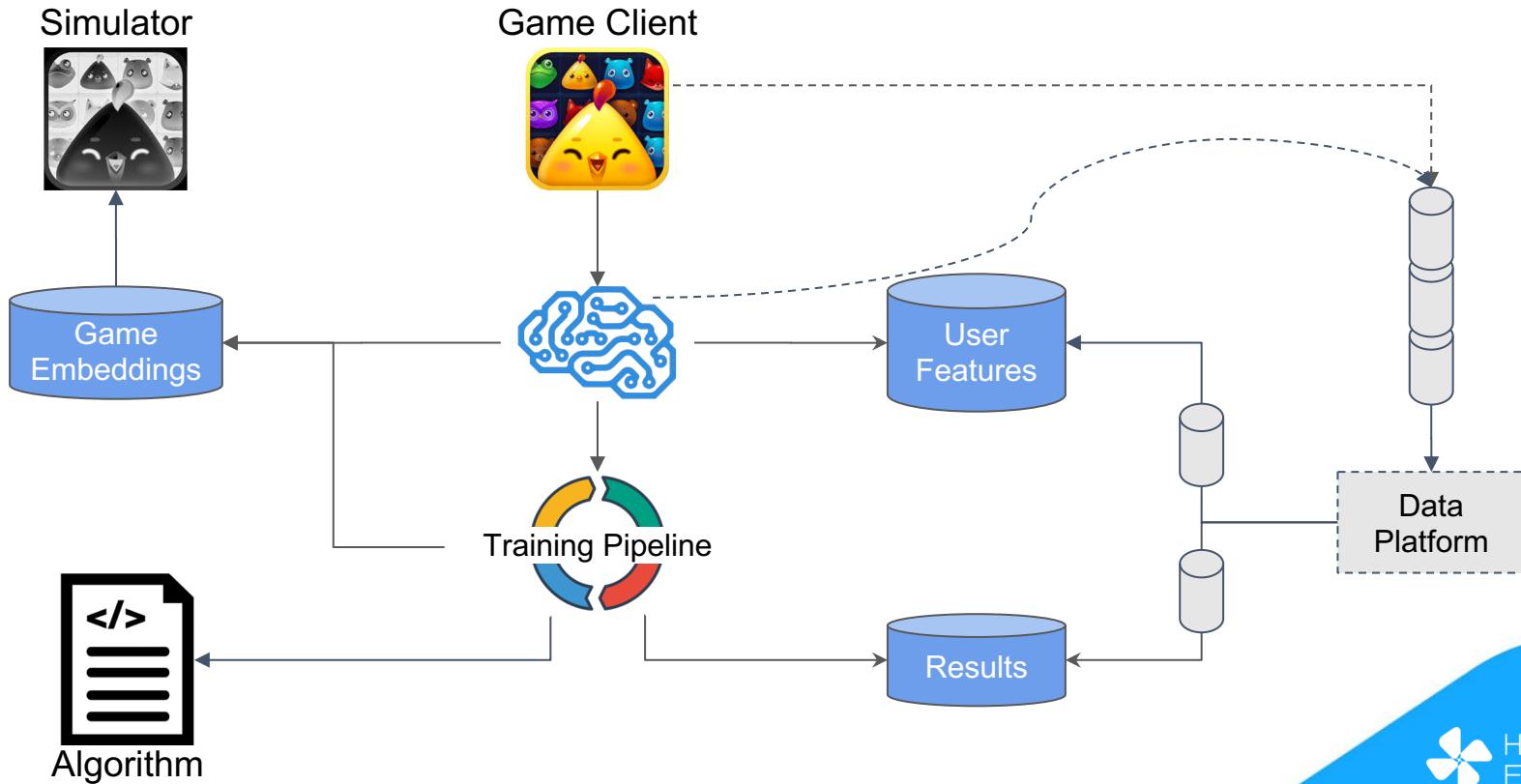
System Architecture



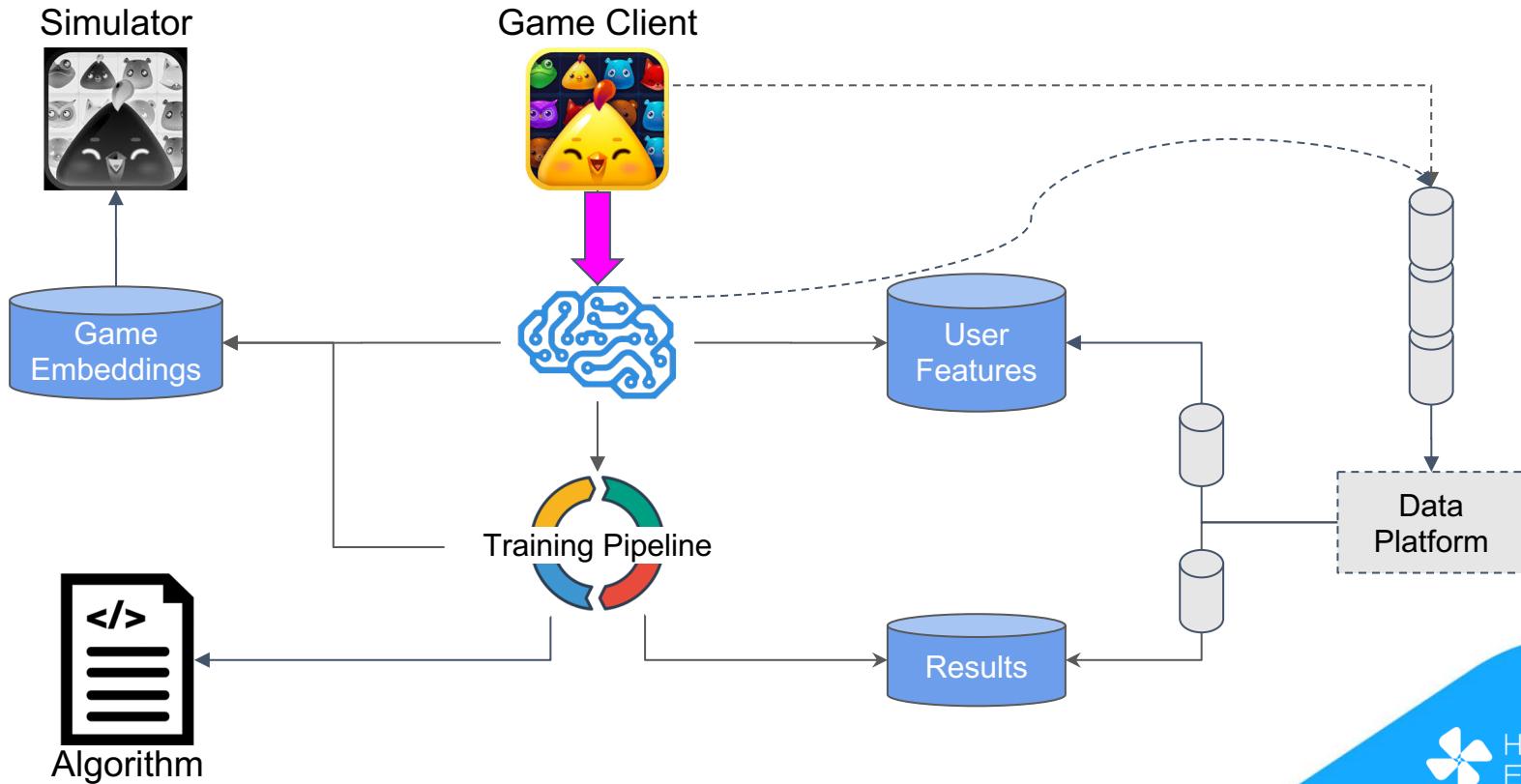
System Architecture



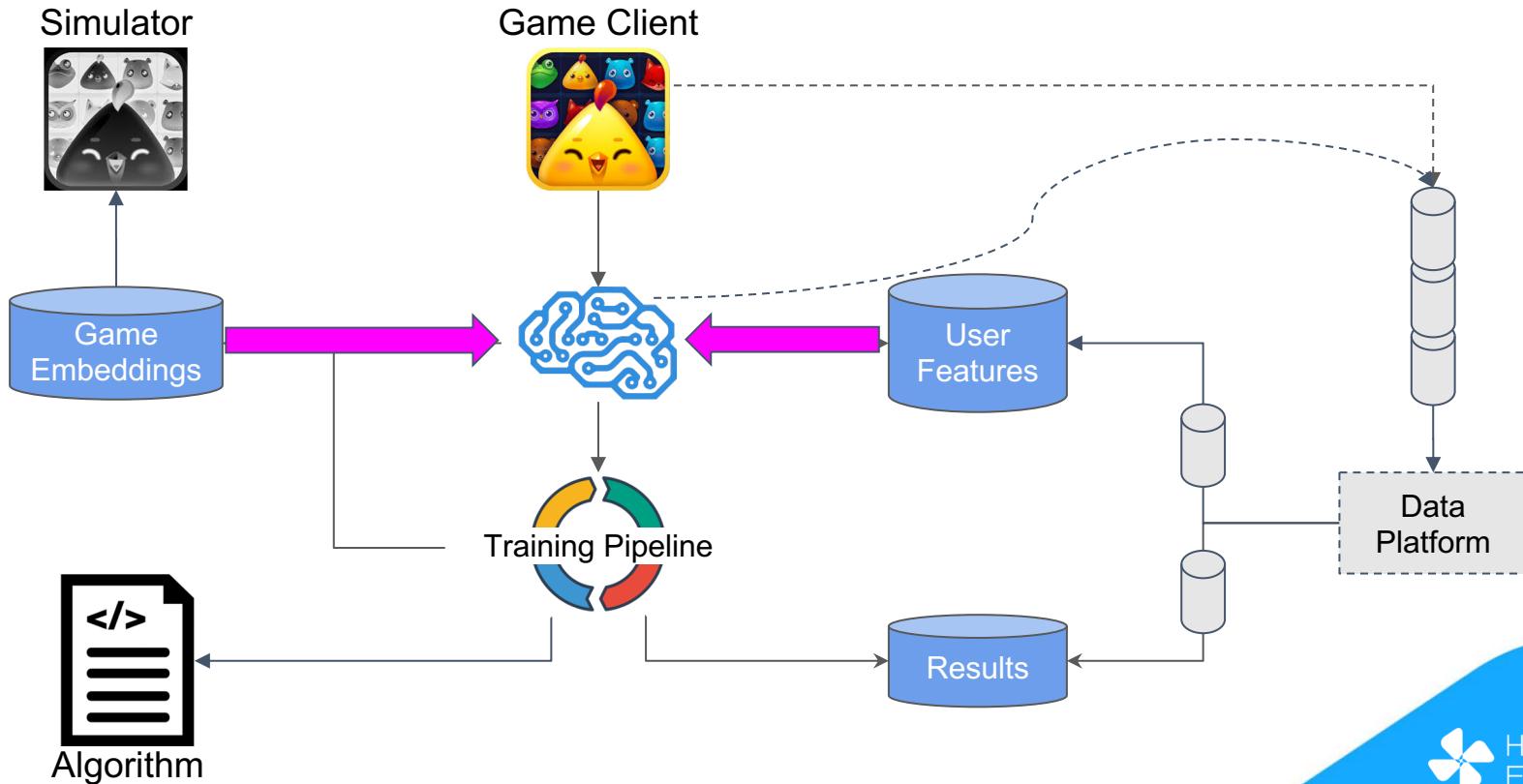
System Architecture



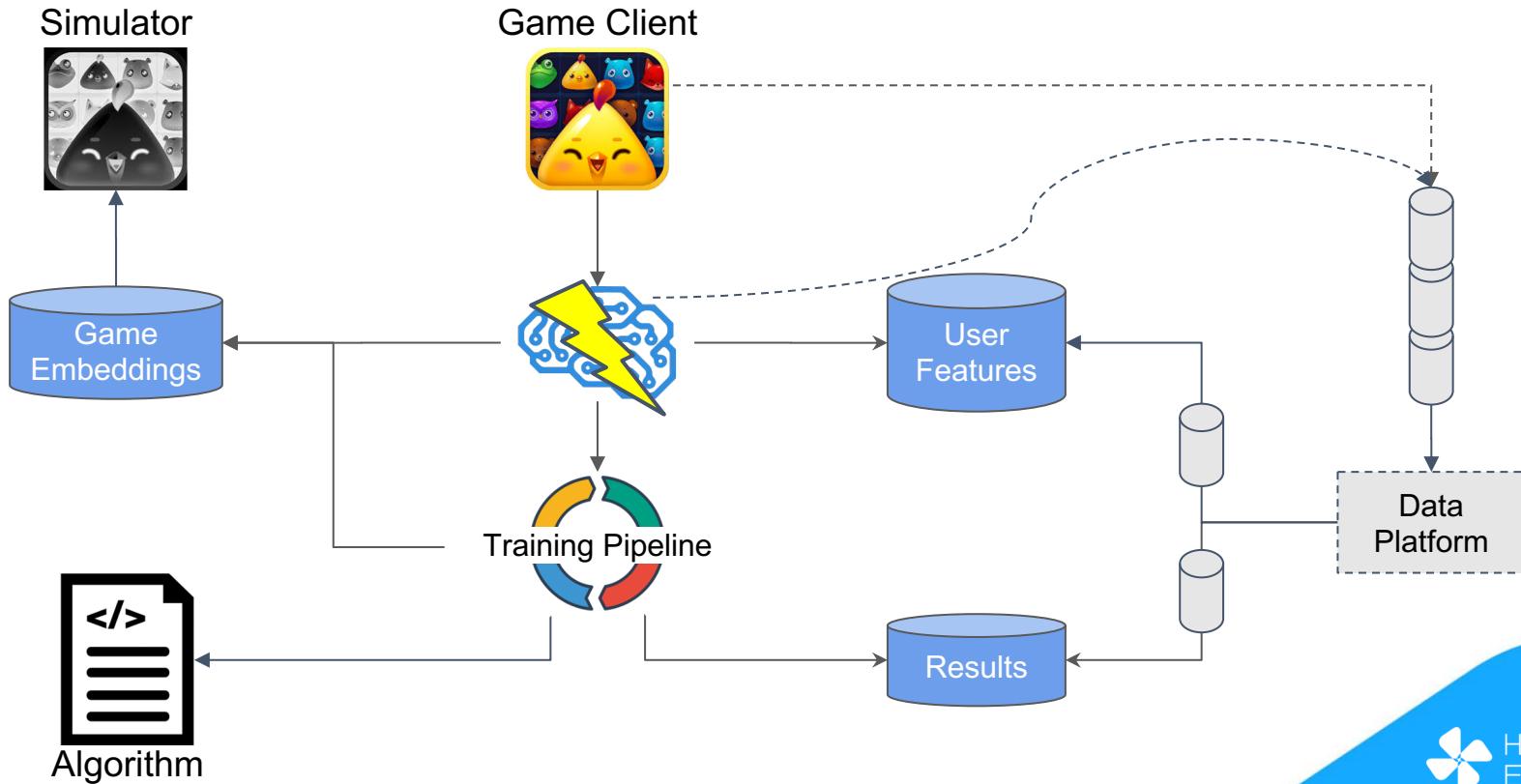
System Architecture



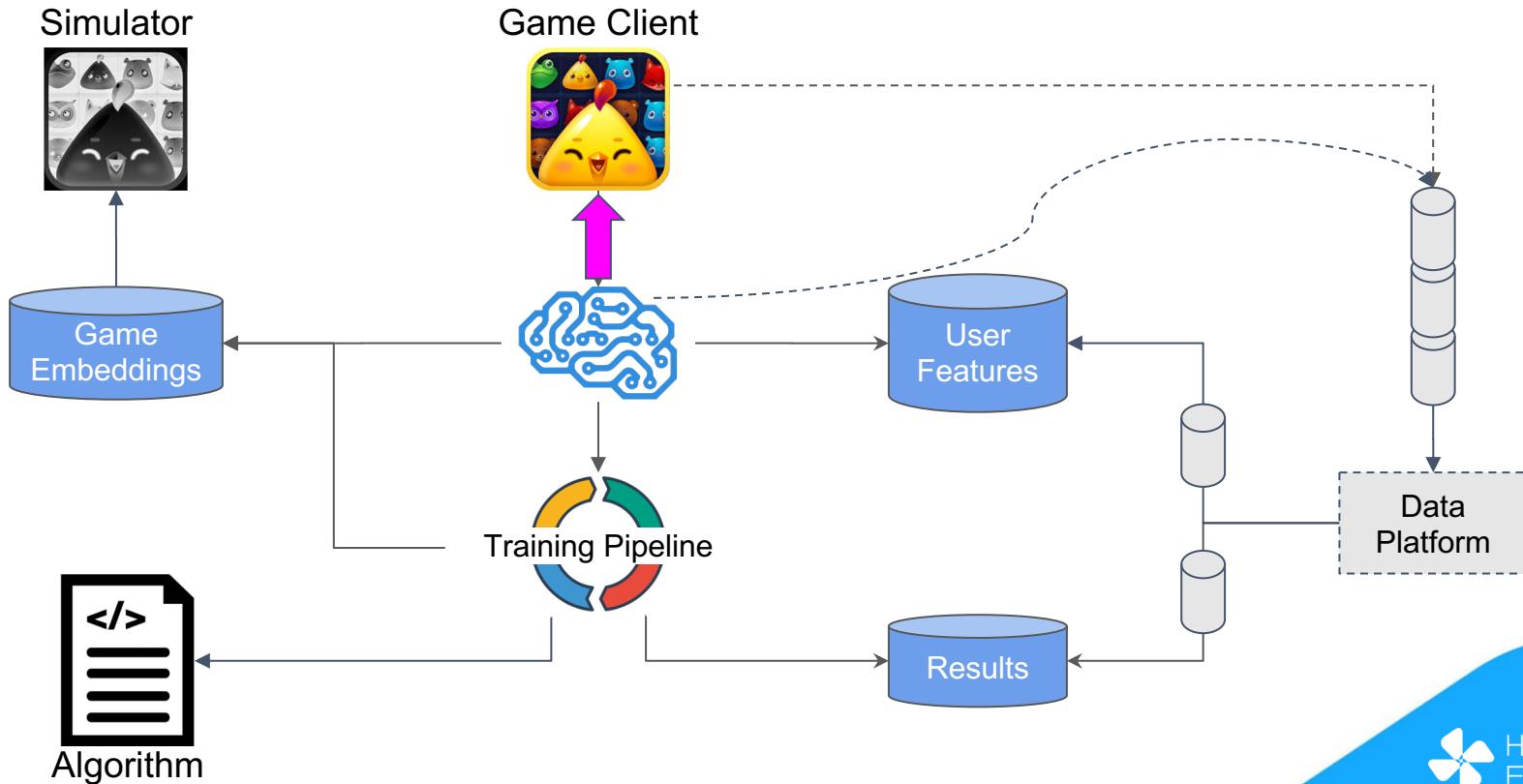
System Architecture



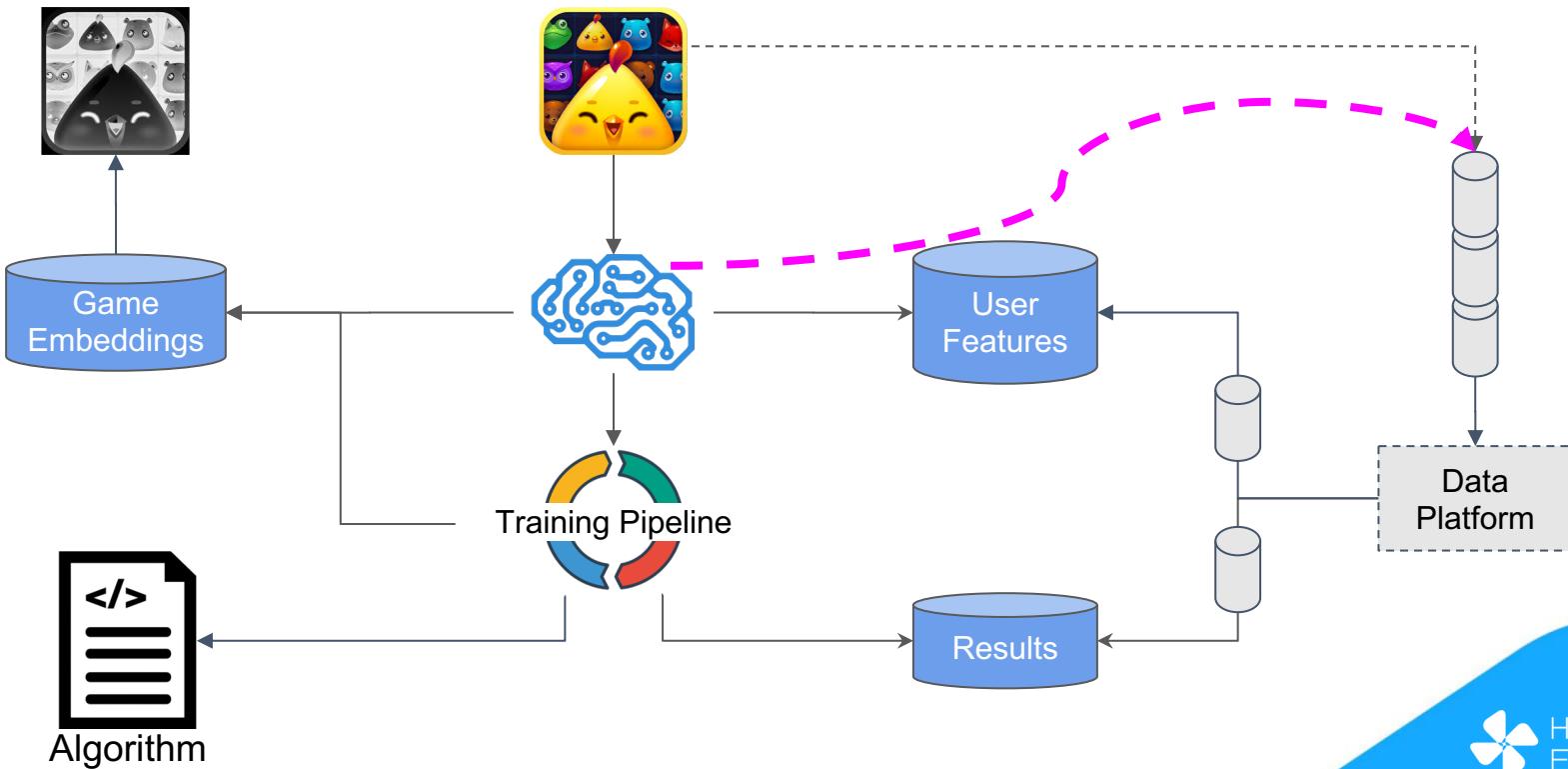
System Architecture



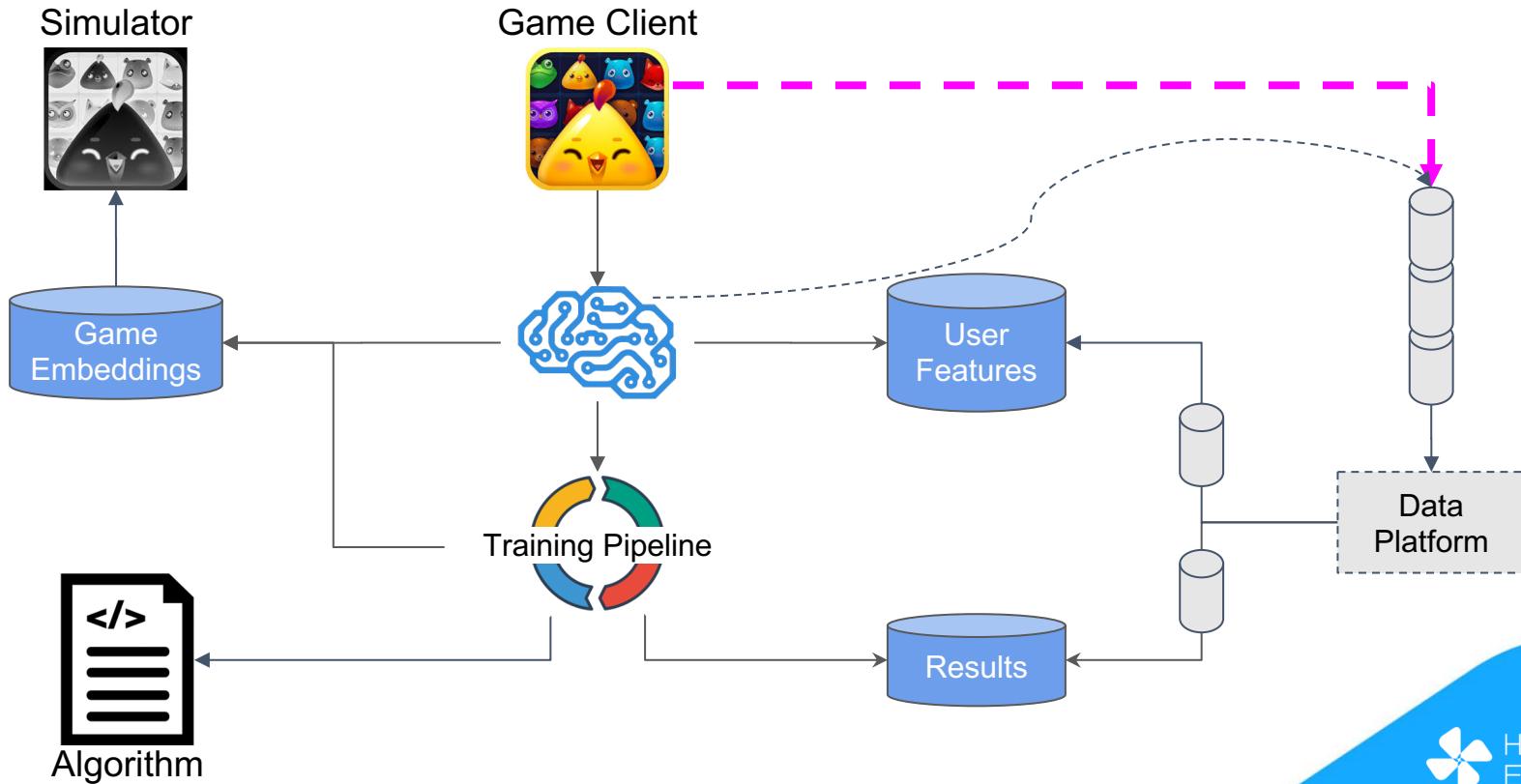
System Architecture



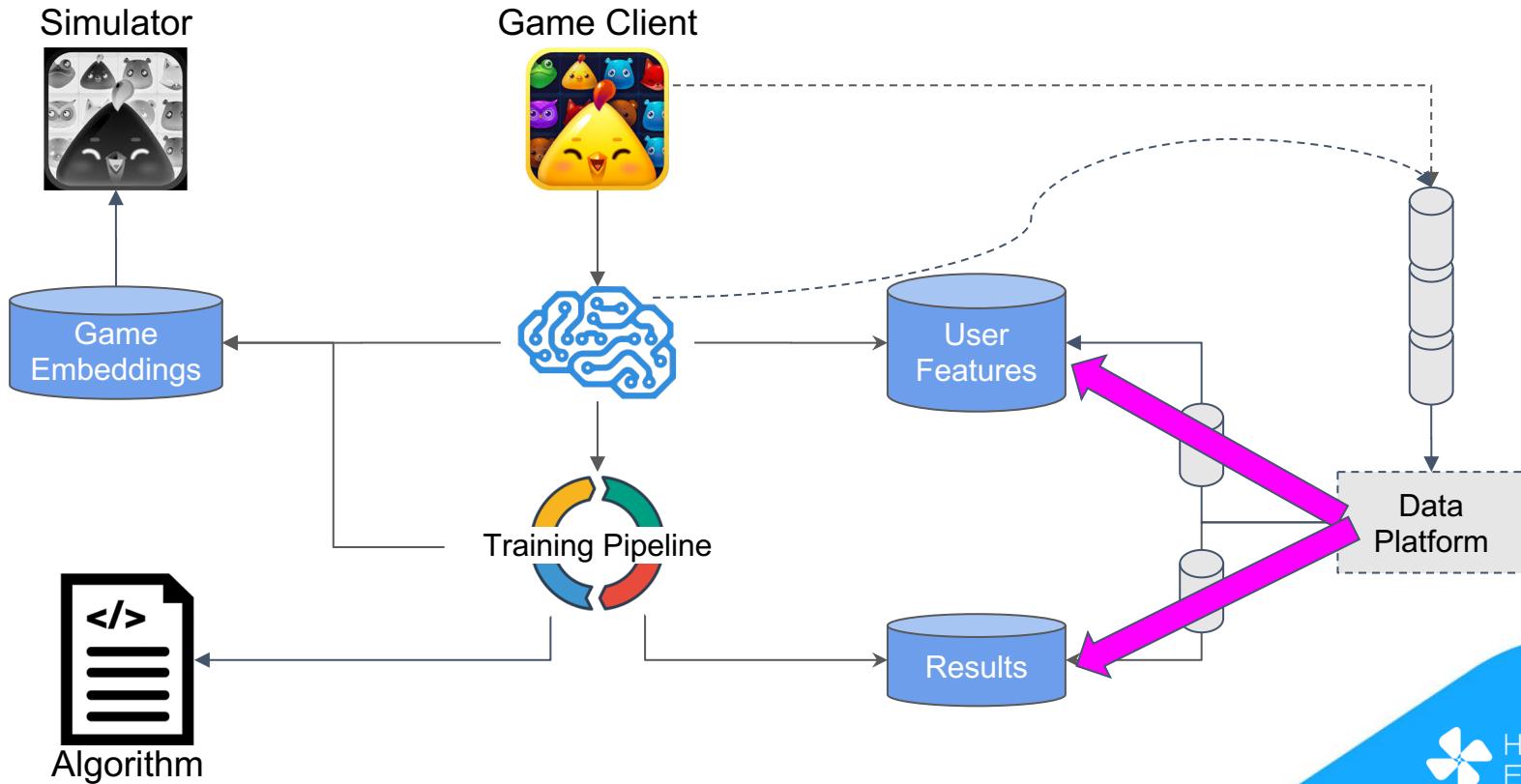
System Architecture



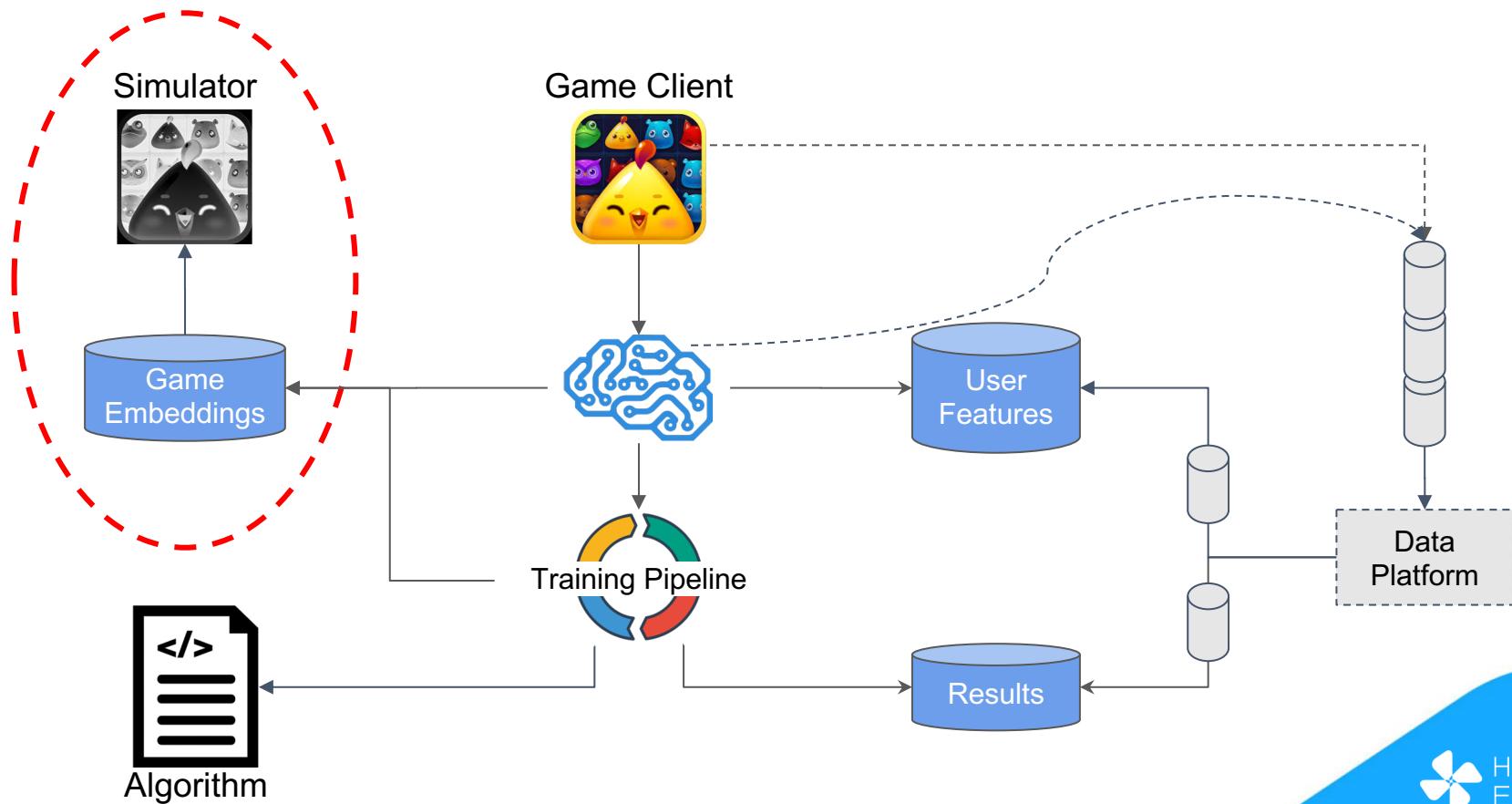
System Architecture



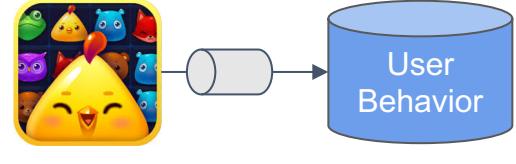
System Architecture



System Architecture

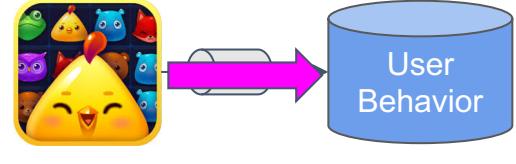


System Architecture: Gameplay Embeddings



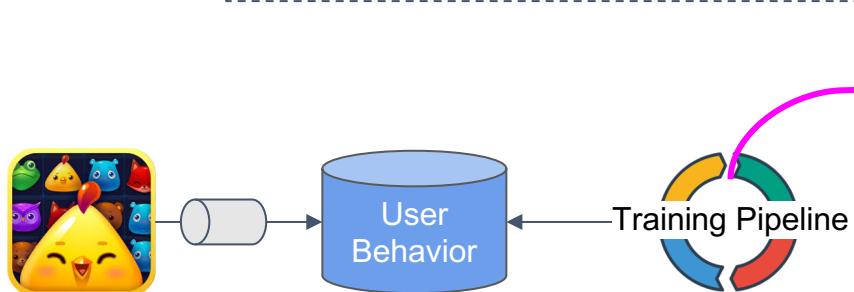
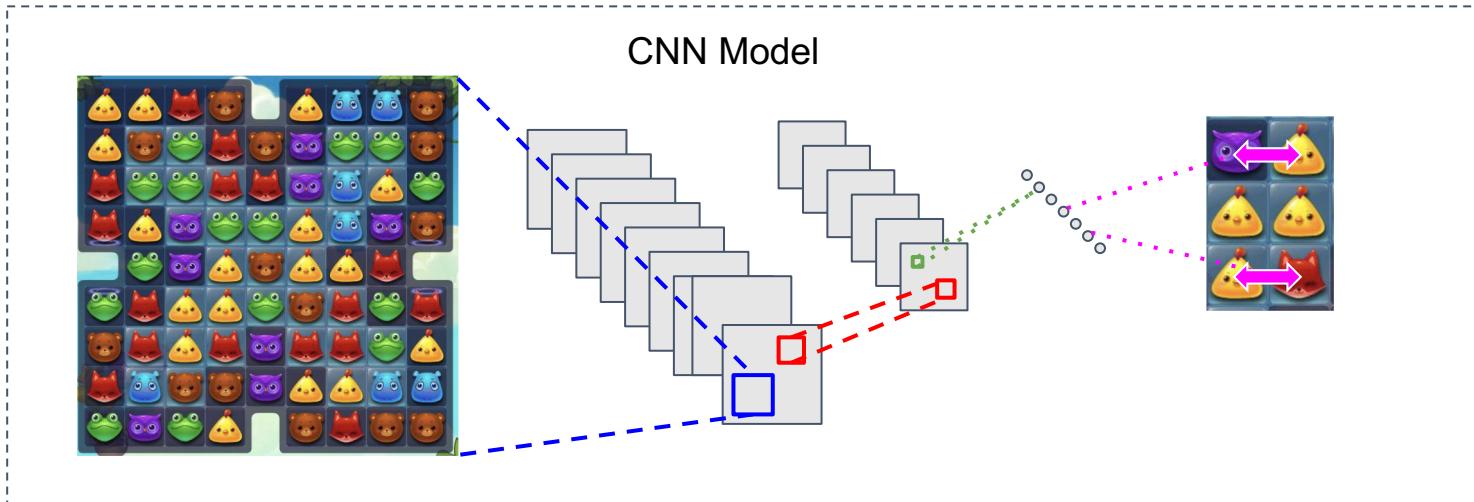
Game Client

System Architecture: Gameplay Embeddings

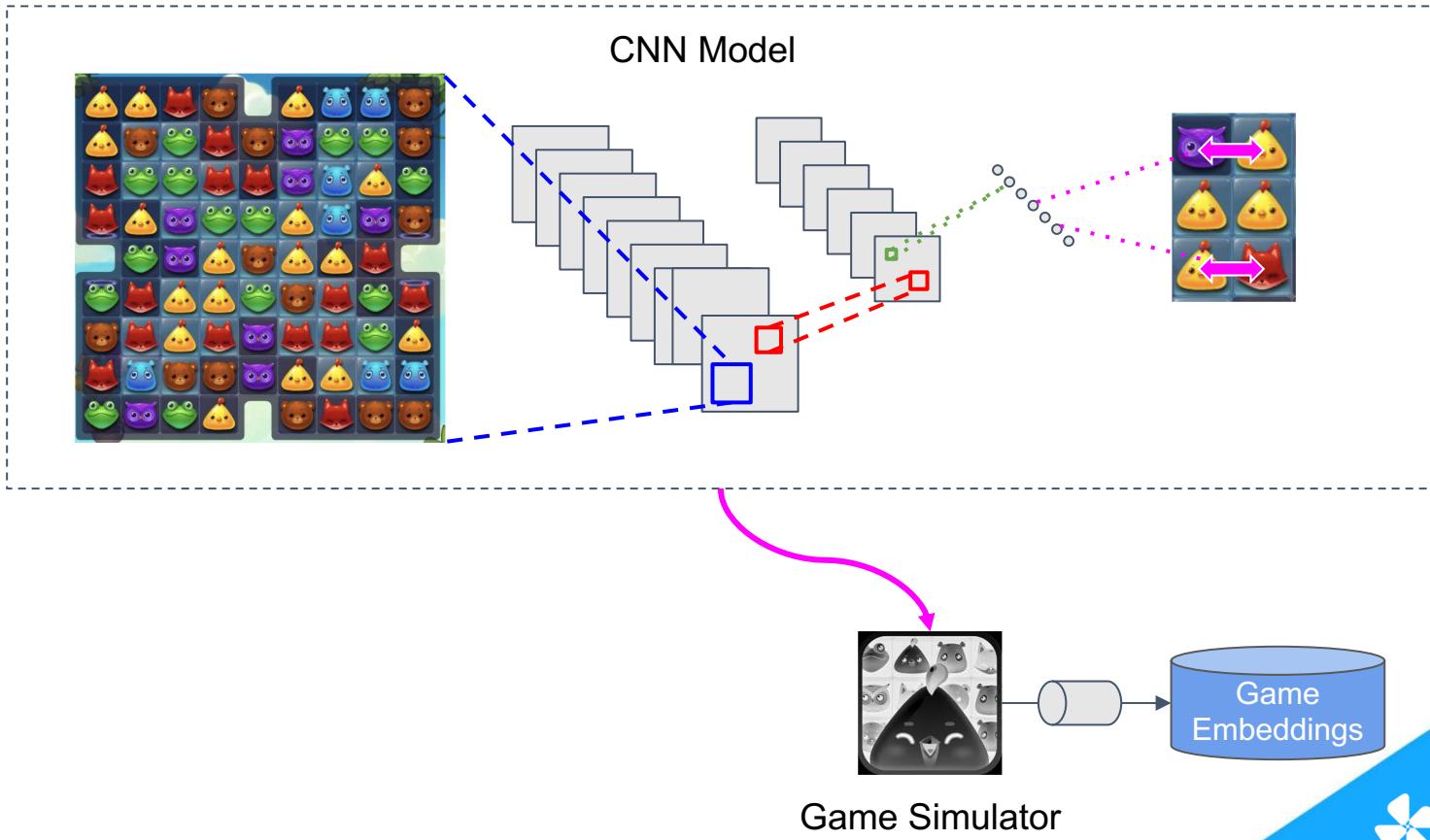


Game Client

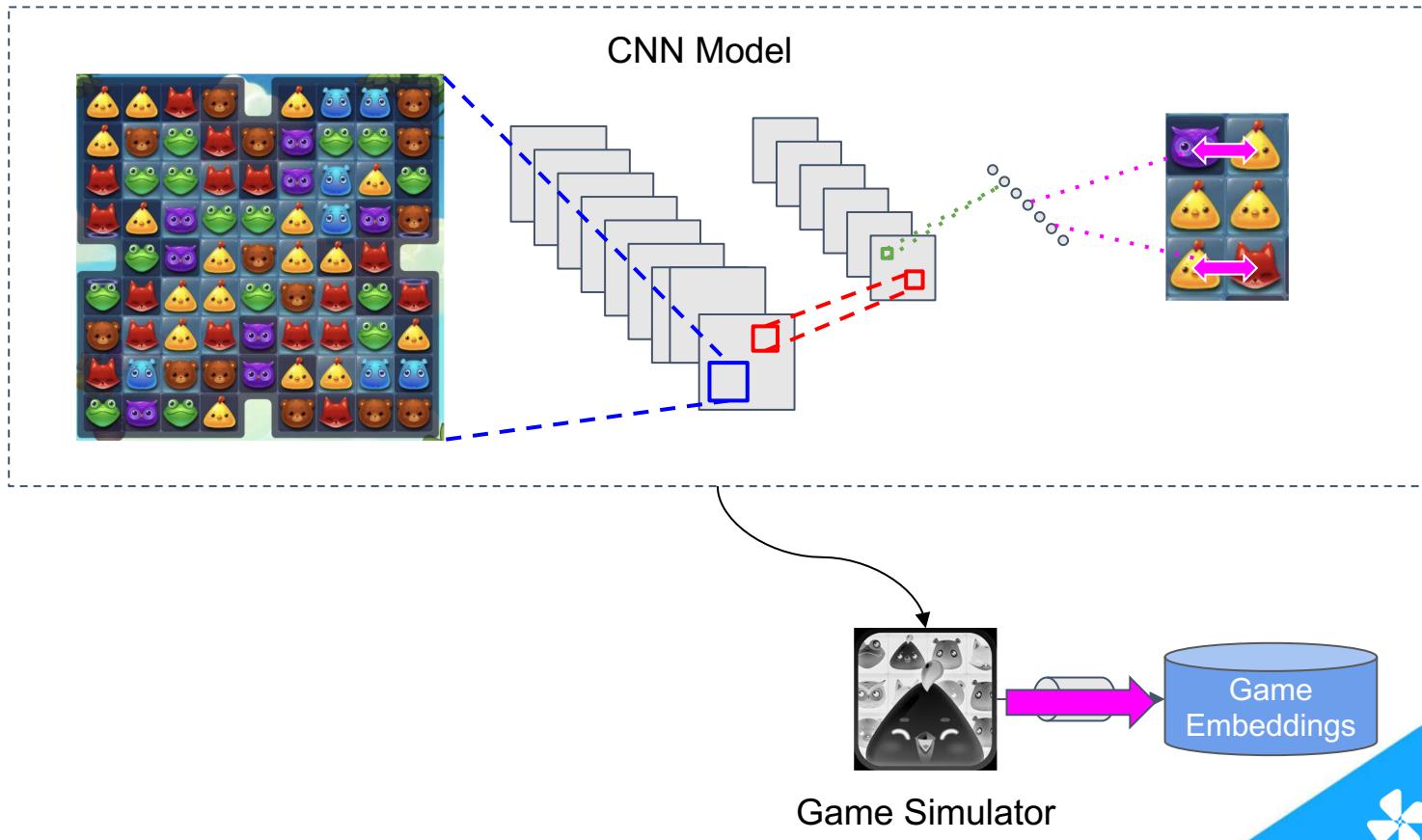
System Architecture: Gameplay Embeddings



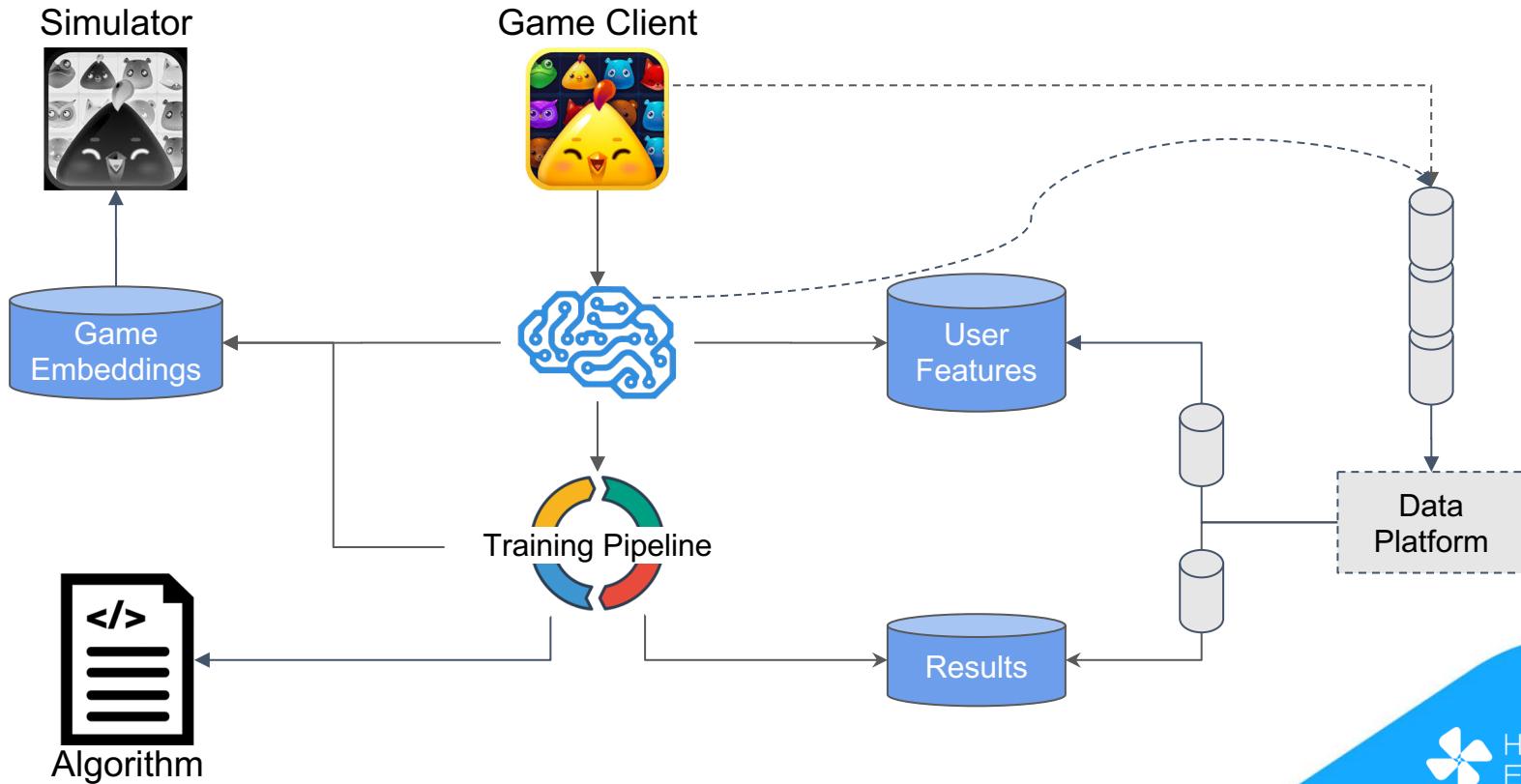
System Architecture: Gameplay Embeddings



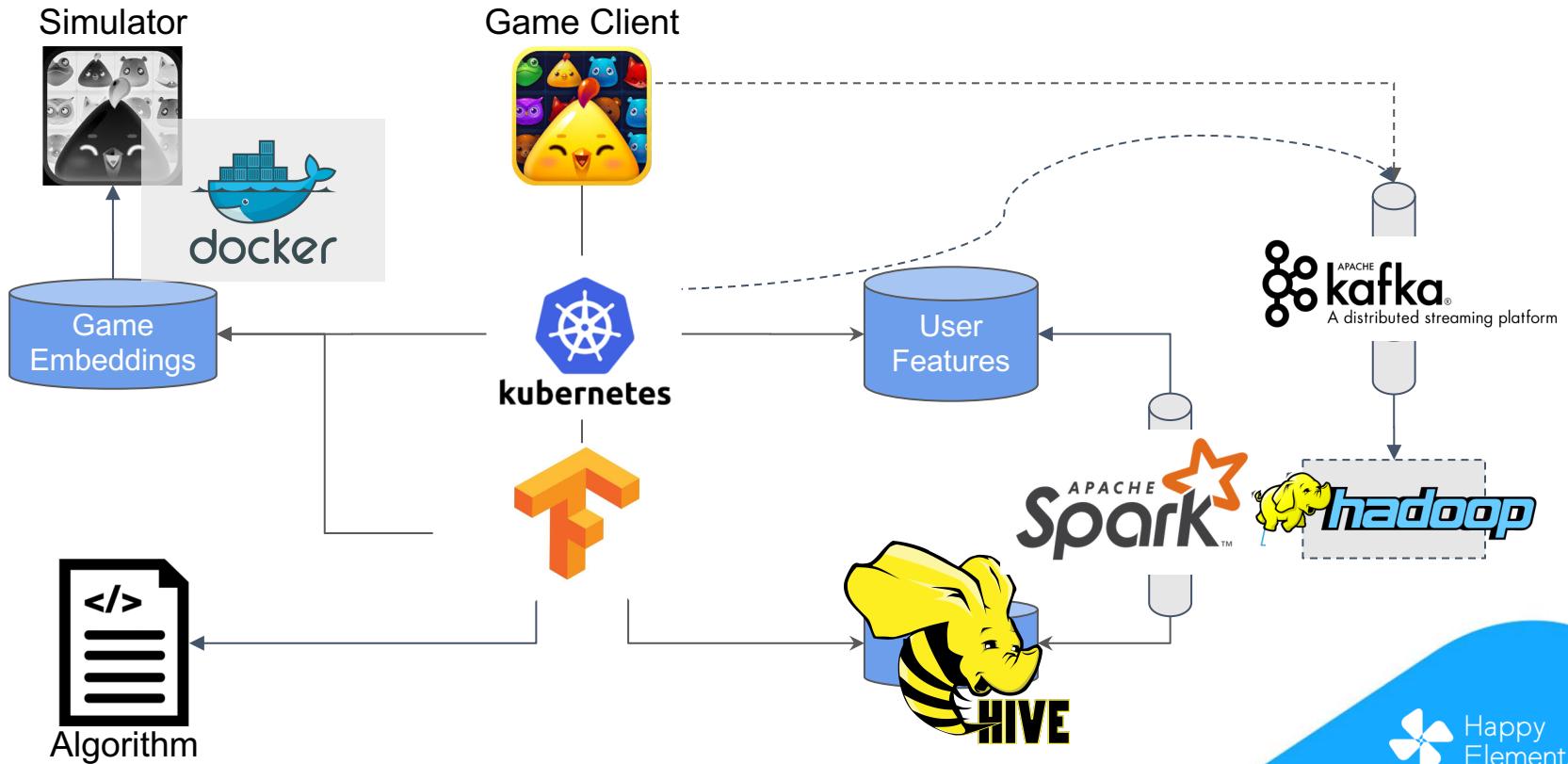
System Architecture: Gameplay Embeddings



System Architecture



Technology Stack Samples



Conclusions

- Production deployment of RL significantly impacts player LTV/retention
- ML effective in closing the loop in game development
- How to design games while taking into account AI capabilities?



THANKS

<http://en.happyelements.com/ai>