Dual-Objective Adaptive Adversary Design with Reinforcement Learning for Dynamic Difficulty Adjustment in a Turn-Based Tactical Game

Mustafa Khan, Roy Khoo

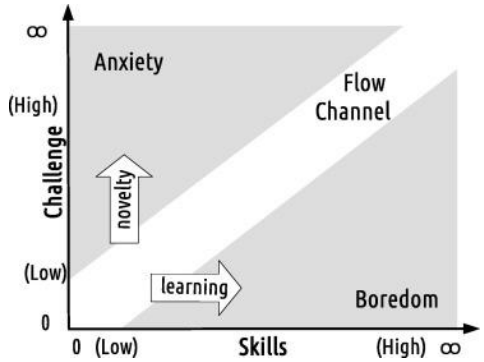
IOWA STATE UNIVERSITY

Problem Statement

- Most game AI is designed to maximize victory, not player engagement
- Traditional DDA often reduces difficulty to simple performance signals
- Player experience depends on tension, pressure, close calls, and recovery, not just outcomes
- Current approaches do not model these experiential qualities well in turn-based tactical games

Why?

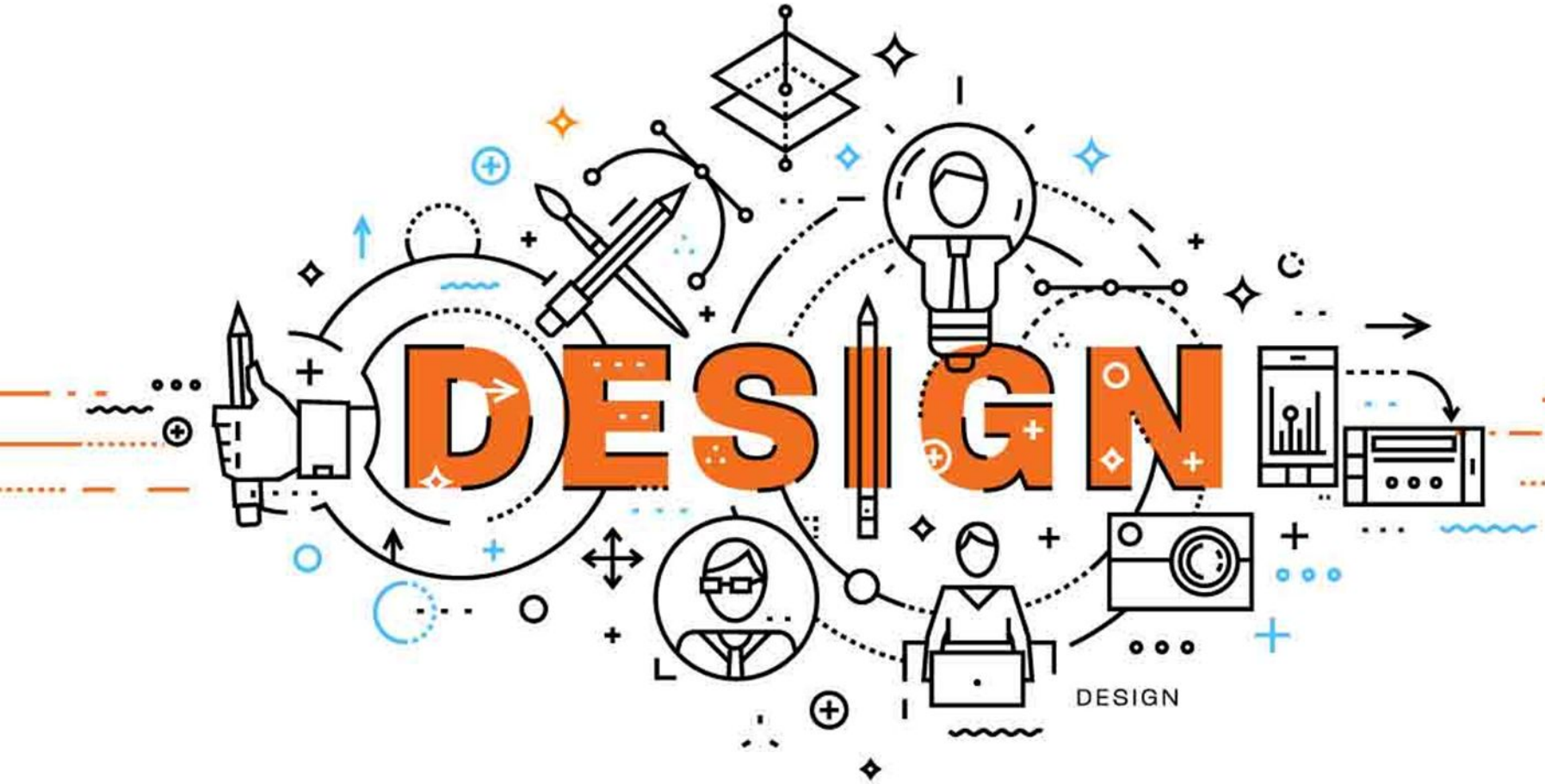
- Players lose interest when the level of challenge does not align with their skill
- Users in training simulations learn less effectively when they are disengaged
- The game industry invests major effort in hand-tuned difficulty that often does not generalize



Research Question

Can an RL agent learn to control enemy behavior and also optimize player engagement rather than simply increase difficulty, while adapting online to player state?

1. Which RL architecture performs best for this task?
2. Does explicitly separating challenge from engagement improve adaptation?
3. How can player engagement be measured without direct survey data?


DESIGN
DESIGN

Approach Overview

Dual-Objective System

- Instead of a single reward, we split the decision criteria: Challenge Policy (aims for tactical advantage) vs. Engagement Policy (aims for player satisfaction)
- Goal: An AI that pulls its punches or intensifies based on real-time player state

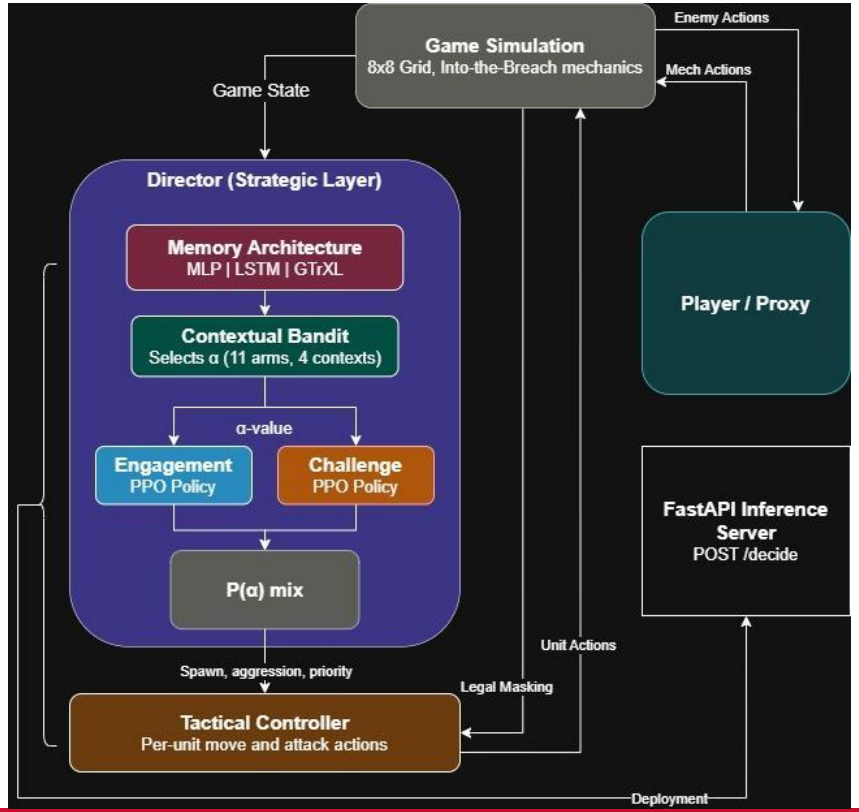
Hybrid 2-Layer Model

- Strategic Layer [The Director]: Operates on a macro-scale (spawn rates, objective shifts, unit types)
- Tactical Layer [Unit Controller]: Manages micro-movements and individual unit logic

Engagement Metric

- Implemented a stress/intensity tracker similar to other SOTA
- Monitors intensity cycles ensuring the player has moments of high adrenaline followed by breathing room to prevent burnout.

System Architecture



Contextual Bandit Meta-Controller

- The policy-mixing decision is handled by a **contextual bandit** rather than a fixed challenge–engagement ratio
- The controller observes the player's current health state and selects a mixing coefficient **α** at each turn
- 4 player contexts: **struggling** (< 30% HP), **challenged** (30-50%), **comfortable** (50-70%), **dominating** (> 70%) The action space contains **11 arms**, corresponding to **α values from 0.0 to 1.0** in steps of 0.1
- Arm selection uses **Upper Confidence Bound (UCB)**, balancing exploration of less-sampled arms with exploitation of strong-performing ones
- Each context maintains independent arm statistics, so the system learns different strategies for different player states

Director vs Tactical - Hybrid 2 Layer Model

Strategy Director

"Determines overall strategy to employ and what is on the board"

Inputs

- 8x8 grid state (terrain, occupied, etc)
- Player unit positions, HP, abilities used
- Building position and HP
- Player engagement metrics (flow, close calls, etc)
- Skill estimate (Player Modelling[!])

Outputs

- Spawn request: enemy type, position, count
- Param: target priority

Tactical Controller

"How pieces move and attack on a turn-to-turn basis"

Inputs

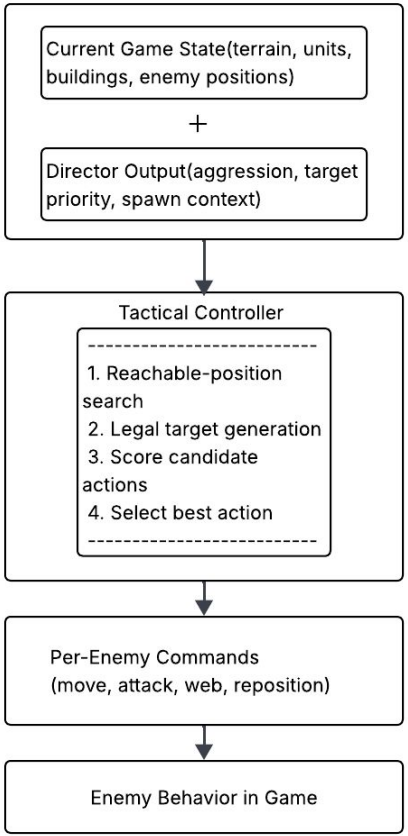
- Current unit position, HP, type, abilities
- Nearby allies, enemies, terrain
- Director high-level directive (spawn, priority)
- Legal actions, telegraphed attack zones
- Current phase context

Outputs

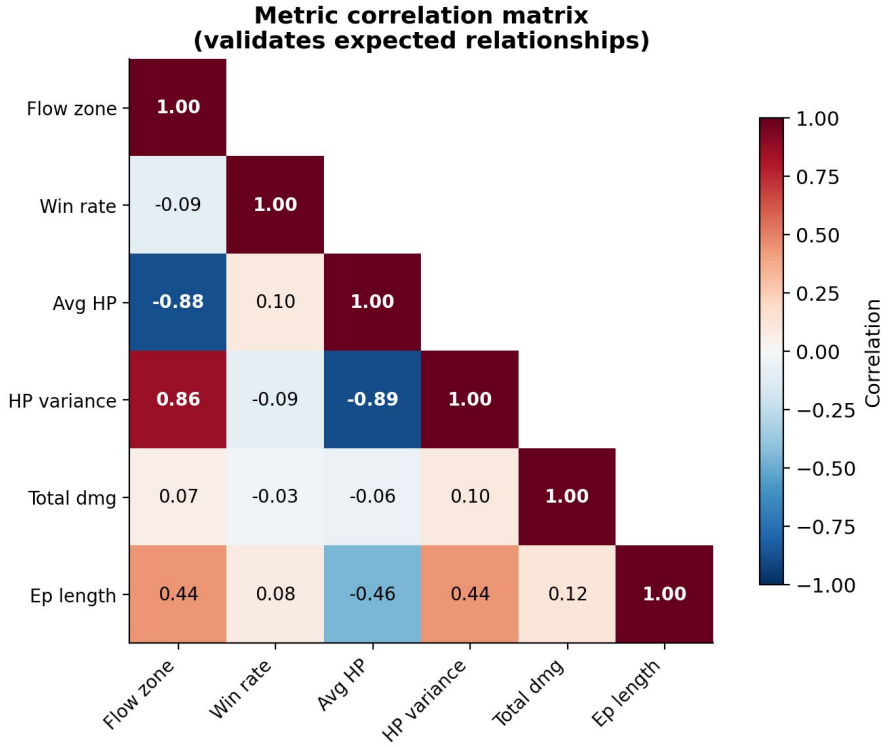
- Move command, direction and tile
- Attack action
- Special ability [?]

Tactical Component

- Translates the Director's high-level intent into per-enemy movement and attack decisions
- Operates on the full game state, including terrain, unit positions, and current directives
- Uses a search-and-score procedure to evaluate reachable positions and legal targets
- Combines local tactical opportunities with Director signals such as aggression and target priority
- Provides a clear separation between macro-level adaptation and micro-level execution



Reward Design



Reward Design *CONTINUED*

Flow Zone Rate Core

Percentage of episode spent in optimal challenge zone (HP between 30-70%)

```
flow_zone_rate = flow_zone_steps / episode_length
```

Close Call Rate Core

Near-death moments per step. Team-level (HP < 25%) + individual unit close calls.

```
close_call_rate = (close_calls + close_calls_individual) / episode_length
```

Combat Rate Core

Percentage of turns with active combat engagement.

```
combat_rate = combat_events / episode_length
```

Difficulty Score Derived

How hard the episode was. Combines HP loss and enemy pressure.

```
difficulty = ((1 - avg_hp) + min(avg_enemies/5, 1)) / 2
```

Grit Score Derived

Challenge faced and overcome. High grit = took damage but survived.

```
grit = (damage_taken / (damage_taken + 10)) × survival_bonus + close_call_bonus
```

Tension Score Derived

HP fluctuation measuring drama and swings between safe/danger states.

```
tension = mean(|diff(hp_history)|)
```

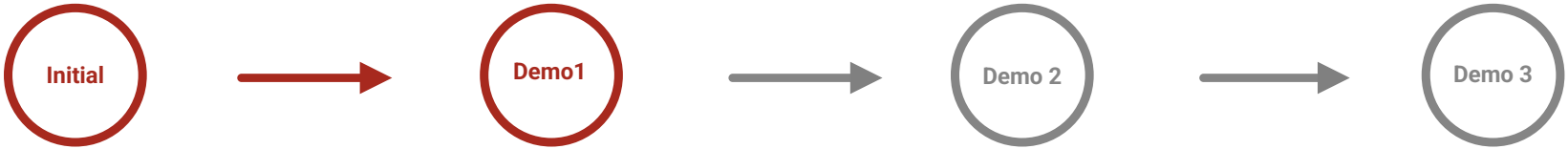
Engagement Score Composite

Primary optimization target. Weighted combination of core metrics measuring overall player engagement (0-1 scale).

```
engagement = (0.4 × flow_zone_rate) + (0.3 × min(close_call_rate × 10, 1)) + (0.2 × min(combat_rate × 5, 1)) + (0.1 × outcome_score)
```

40%	30%	20%	10%
Flow Zone	Close Calls	Combat	Outcome

Timeline



Research

- Project scoping with Lockheed Martin, defining engagement vs difficulty problem
- Literature review: Left 4 Dead Director, Agent57, flow theory, DDA research

Exploration

- Investigated player modeling approaches (IQL, clustering) before pivoting to dual-objective direction
- Early algorithm exploration across on-policy and off-policy methods with different memory architectures

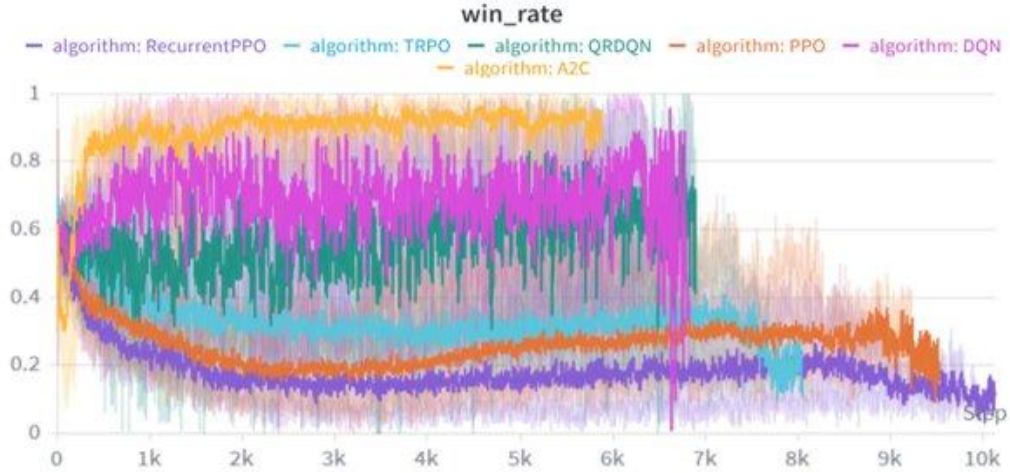
Implementation

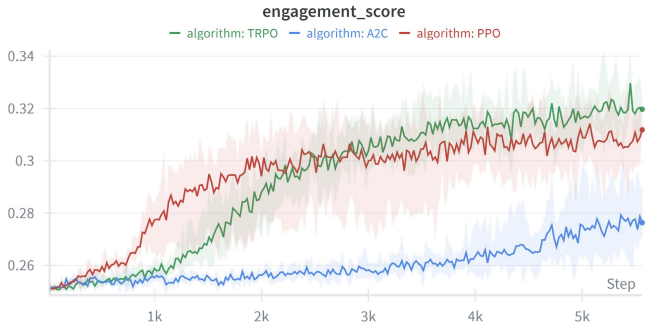
- Designed engagement and challenge reward formulations
- Implemented contextual bandit meta-controller with UCB arm selection
- Reward validation: saturation curves, sensitivity analysis, correlation analysis
- Tactical component integration with search-and-score policy

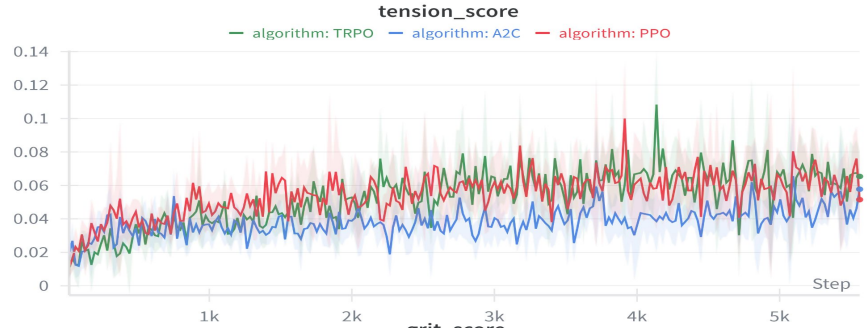
Integration

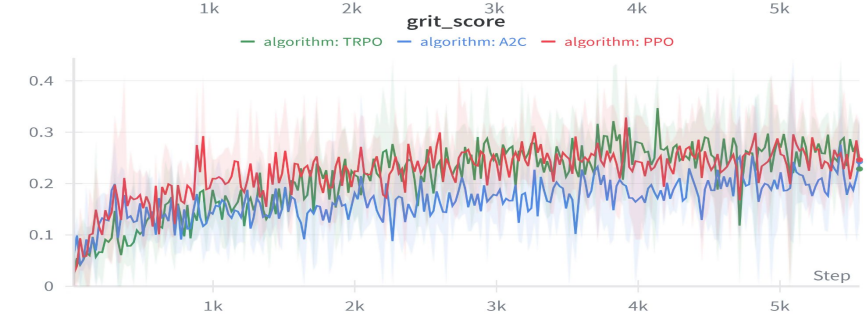
- FastAPI inference server deployed on university VM
- JSON API contract aligned with undergrad game team
- Full results analysis: architecture comparison, objective comparison, metric validation

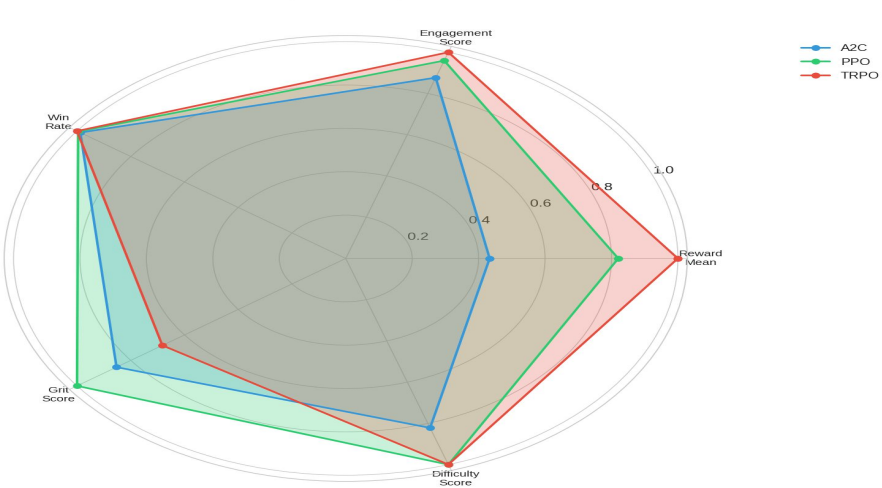

RESULTS


win_rate
algorithm: RecurrentPPO
algorithm: TRPO
algorithm: QRDQN
algorithm: PPO
algorithm: DQN
algorithm: A2C
Step


engagement_score
algorithm: TRPO
algorithm: A2C
algorithm: PPO
Step


tension_score
algorithm: TRPO
algorithm: A2C
algorithm: PPO
Step


grit_score
algorithm: TRPO
algorithm: A2C
algorithm: PPO
Step

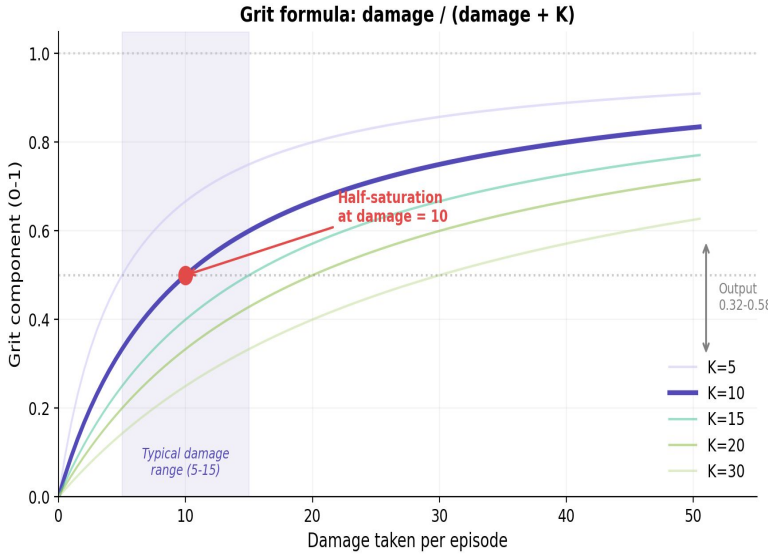

Algorithm Performance Profile
(Normalized)
Engagement Score
Win Rate
Reward Mean
Grit Score
Difficulty Score
A2C
PPO
TRPO

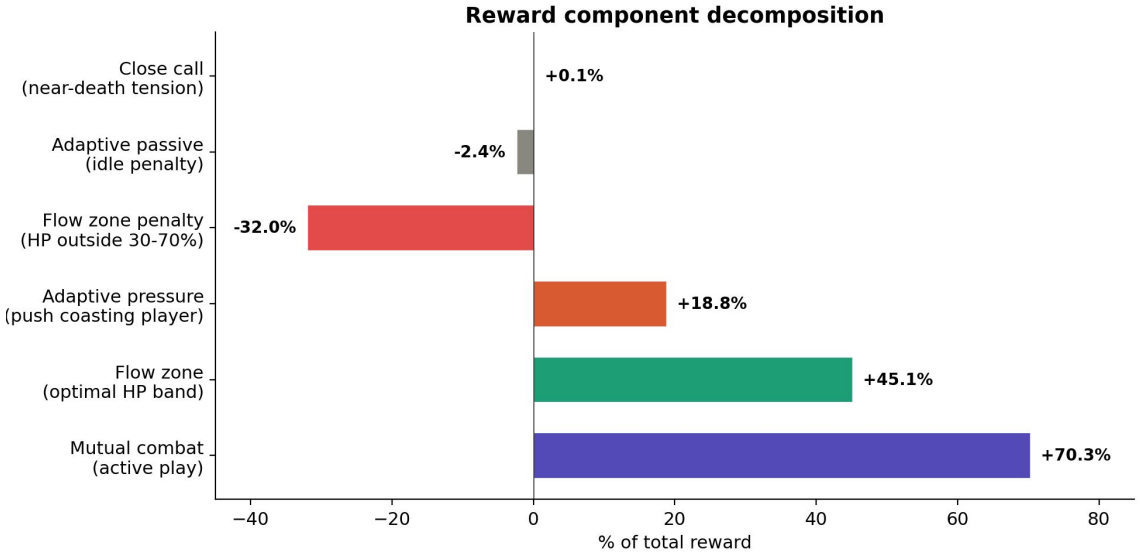
Memory Architecture and Policy Comparison

Architecture	Avg. Reward	Best Reward	Avg Time	Relative Speed
MLP	5.39	5.54	1.6h	1×
LSTM	4.51	4.60	3.2h	2× slower
GTrXL	4.58	4.99	10.3h	6.4× slower

Policy	Engagement	Flow Zone	Win Rate	Grit	Reward
Dual	**0.414**	0.373	**98.4%**	0.293	**5.45**
Engagement	0.391	**0.381**	97.7%	0.291	5.19
Challenge	0.342	0.378	97.8%	**0.295**	4.94

Metric Evaluation





Reflection

- **Engagement is harder to optimize than difficulty** because it depends on tension, recovery, pacing, and player state, not just whether the player wins or loses.
- **The dual-objective framing was useful**, but the reward definitions must be separated carefully so the challenge and engagement policies learn meaningfully different behaviors.
- **Simpler architectures can be stronger when the state is well designed**, as the MLP outperformed memory-based models because the observation already captured the key gameplay context.
- **The proxy player strongly shaped the results**, since its high win rate reduced close-call situations and limited the full use of the engagement reward.
- **Deployment exposed the importance of modular design**, especially separating the Director, Tactical Controller, simulator, and FastAPI integration.

Future Work

Human player studies: Validate engagement metrics against subjective player experience and adding richer player modeling to improve bandit context.

Reward diversification: Explore more separated policies and stronger policy-mixing behavior

Meta-progression and longer training: Add upgrades and evolving loadouts so memory architectures face real temporal dependencies.

Tactical improvements: Add MCTS-based lookahead and synergy-aware spawning to handle multi-step planning and nonlinear effects (e.g., push-into-water).

Difficulty reframing: Model difficulty by execution tolerance, counting viable action sequences, not just enemy strength, and validate engagement with a human player study.



Appendix

Related Works

L4D Director [Valve, 2008]

- Most recognized deployed in-game AI director, still referenced as foundation
- Stress-based spawning with hand-crafted pacing rules, no learning capability
- Similar product recently, Darktide (2022), Anacrusis (2021). No production game has shipped a learned director since
- Demonstrates the gap: industry relies on rule-based systems despite advances in RL

Related Works

Agent57 – Outperforming the Atari Human Benchmark [2020]

- First agent to surpass human performance on all 57 Atari games
- Introduces bandit meta-controller that dynamically selects between a family of policies (short-term vs long-term gain)
- Uses R2D2 (Recurrent Replay Distributed DQN), LSTM
- **Difference:** Optimize for score/win-rate as goal was to beat human benchmarks
- **Adaptation:** Our contextual bandit selects between engagement and challenge policies for player experience

Related Works

EnMatch – Matchmaking for Better Player Engagement [2024] Marvel Rivals

- Combination optimization problem, replaces traditional fair game matchmaking with a **learned engagement function**
- Uses GNN encoder, masked Transformer decoder and LNS heuristic operator trained with PPO
- More engagement = Higher collaboration (chat frequency, upvotes/downvotes) and lower churn
- Difference:
  - Engagement signals require real player behavourial data
  - Pre-game assignment and not in-game play

Related Works

Academic DDA – Hunicke [2004], Romero-Mendez [2023]

- Treats difficulty as a single continuous slider
- No distinction between challenge and engagement
- Parameter manipulation (fewer enemies, more health) rather than learned behavioral control
- Recent ML-based approaches only classify player skill to adjust a difficulty knob, not to shape experience quality

Game Environment

- **Grid:** 8x8 tile-based tactical combat
- Turn-based combat with mitigatable enemy attacks from displacement
- **Player Units:** 3 mechs (different stats and ability)
- **Objective:** Protect buildings with civilians from enemies
- **Win Condition:** Survive all enemy waves
- **Lose Condition:** All buildings destroyed or all allied unit destroyed





Buildings



Tiles



Ally Units



Enemy Units

Game Environment *CONTINUED*

Combat Mech



Dmg: 2 | Range: Adjacent
Hp: 3 | Push: 1
Mv: 3

Cannon Mech



Dmg: 1 | Range: Line
Hp: 3 | Push: 1
Mv: 3

Artillery Mech



Dmg: 1 | Range: Line
Hp: 2 | Radial: 1
Mv: 3

Ranged Vek



Dmg: 1 | Range: Line
Hp: 3 | Push: 0
Mv: 2 | Spe: None

Flying Vek



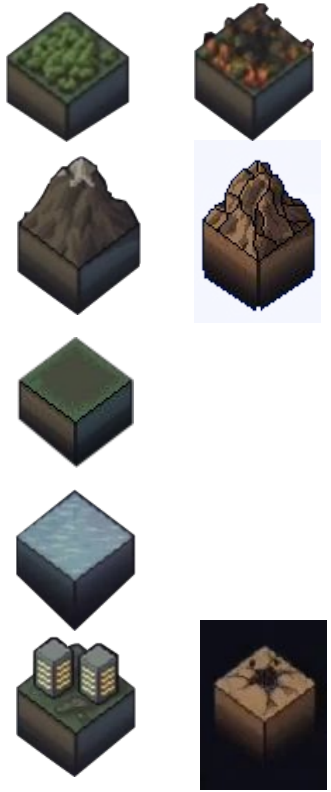
Dmg: 1 | Range: Adjacent
Hp: 2 | Push: 0
Mv: 5 | Spe: Fly

Webbing Vek



Dmg: 1 | Range: Adjacent
Hp: 3 | Push: 0
Mv: 2 | Spe: Web

Game Environment *CONTINUED*



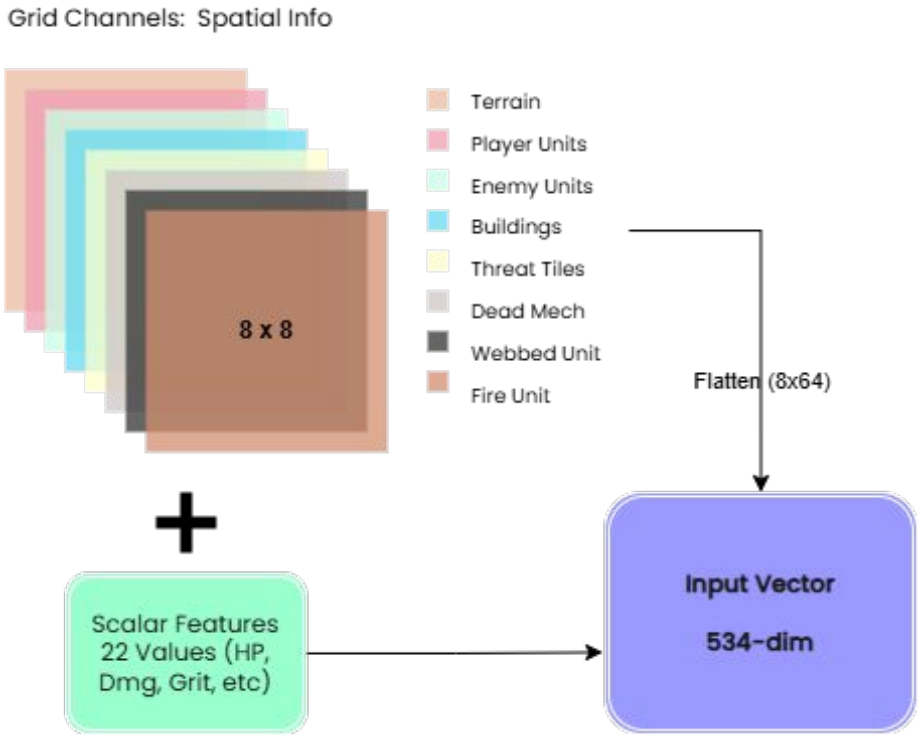
Mechanics

- Attacks by mech have knockback
- Push [Water] – Eliminates Ground Unit
- Push [Mountain] – 1 Dmg to Unit
- Push [Building] – 1 Dmg to Both
- Dead Mech(s) remain as obstacle





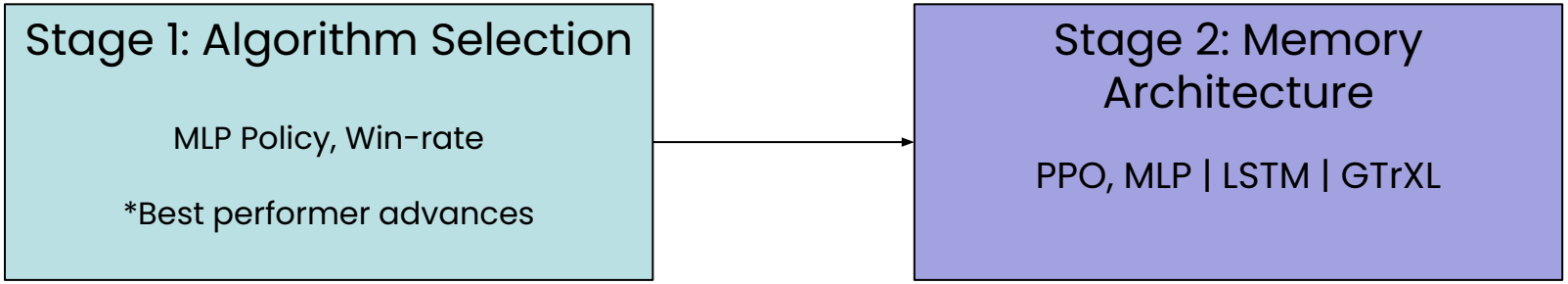
Observation Space



Proxy Player

- A simulated player was needed so the Director could be trained through repeated interaction
- The proxy player is a heuristic tactical agent with a tunable skill parameter in the range [0,1].
- Candidate actions are scored using tactical factors such as enemy damage, building protection, threat avoidance, and push interactions
- Action selection is skill-dependent: higher skill makes strong actions more likely, while lower skill increases randomness
- The default training setting used a proxy skill of **0.7**, representing a reasonably competent but non-optimal player

Training Pipeline



- **Goal:** identify the best RL algorithm–memory architecture combination
- Algorithms: **PPO, A2C, TRPO** , **DQN, QRDQN**
- Memory: **MLP** (SB3), **LSTM** (sb3-contrib), **GTrXL** (RLlib)
- PPO: 95% of TRPO reward at ~40% computational cost
- DQN/QRDQN: struggled with composite action space

Training Pipeline

CONTINUED

- **Goal:** to measure how both **memory design** and **training objective** affected performance, engagement, and efficiency
- After selecting PPO, the study also compared **three policy architectures**: **MLP, LSTM, and GTrXL**
- PPO fixed as algorithm, 2M timesteps, Optuna hyperparameter tuning(50 trials)
- Each architecture was trained under **three objective settings**:
  - **Challenge-only** (traditional DDA baseline)
  - **Engagement-only**
  - **Dual-objective** (Engagement and Challenge)
- This produced a total of **twelve training configurations** across architectures, objectives, and policy variants

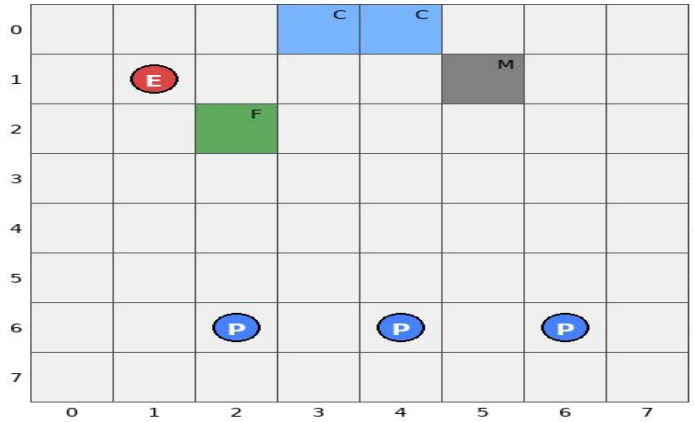
Training Pipeline

CONTINUED

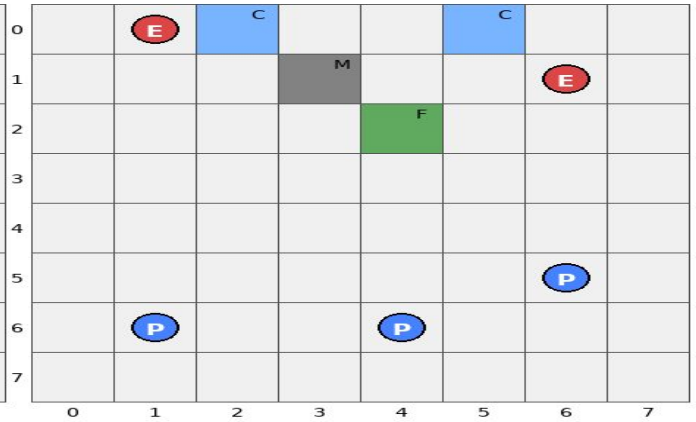
- **Goal:** Adapt Director and Tactical enemy behavior from player context to balance challenge with engagement.
- Offline training runs for 500 episodes with randomized seeds and four sampled scenarios.
- Each enemy turn encodes an 18-feature context, then LinUCB selects one of five alpha values.
- The director outputs pacing, priority, spawn plan, and aggression; tactics execute actions; turn rewards update the bandit.

Sample Scenario Layouts

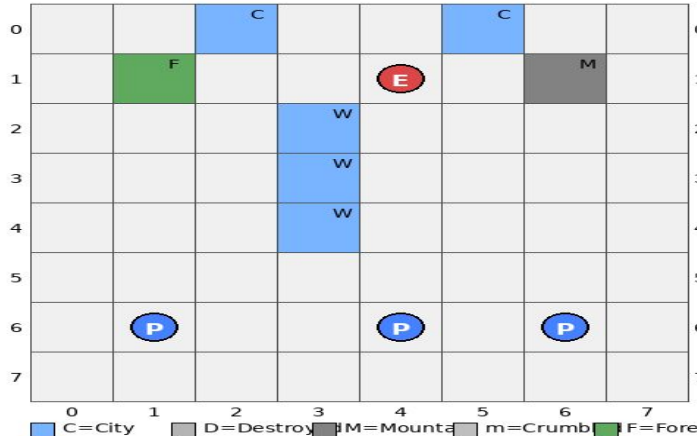
Central Town



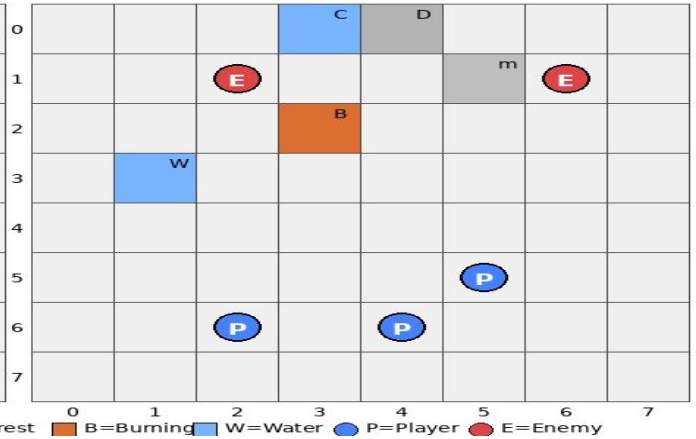
Split Town



River Crossing



Broken City



C=City D=Destroy M=Mounta m=Crumb F=Forest B=Burning W=Water P=Player E=Enemy

Algorithms Used and Their Core Update Rules

Policy Gradient Methods

1 PPO

$$L_{PPO}(\theta) = \mathbb{E}_t \left[\min\left(r_t(\theta)\hat{A}_t,\ \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\,\hat{A}_t \right) \right]$$

$$r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{old}}(a_t \mid s_t)}$$

2 TRPO

$$\max_\theta \ \mathbb{E}_t \left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{old}}(a_t \mid s_t)} \hat{A}_t \right]$$

$$\text{s.t.} \quad \mathbb{E}_t \left[D_{KL}\left(\pi_{\theta_{old}}(\cdot \mid s_t) \,\|\, \pi_\theta(\cdot \mid s_t) \right) \right] \le \delta$$

3 A2C

$$\nabla_\theta J(\theta) = \mathbb{E}_t \left[\nabla_\theta \log \pi_\theta(a_t \mid s_t)\, \hat{A}_t \right]$$

$$\hat{A}_t = R_t - V(s_t)$$

Value-Based / Bandit Methods

4 DQN

$$y_t = r_t + \gamma \max_{a'} Q_{\theta^-}(s_{t+1}, a')$$

$$L(\theta) = \mathbb{E}\left[\left(y_t - Q_\theta(s_t, a_t) \right)^2 \right]$$

5 QR-DQN

$$L_{QR} = \frac{1}{N} \sum_i \sum_j \rho_{\tau_i}^\kappa \left(y_j - \theta_i(s,a) \right)$$

$$\rho_\tau^\kappa(u) = |\tau - 1(u<0)|\, L_\kappa(u)$$

6 LinUCB

$$p_a(x) = x^\top \hat{\theta}_a + \alpha \sqrt{x^\top A_a^{-1} x}$$

$$\hat{\theta}_a = A_a^{-1} b_a$$

$$a_t = \arg\max_a p_a(x_t)$$

Symbol Guide: θ: model parameters $\pi_\theta(a|s)$: policy probability s_t, a_t, r_t: state, action, reward $\hat{A}_t$: advantage ϵ: PPO clip range D_{KL}: KL divergence δ: TRPO trust-region bound R_t: return $V(s_t)$: state value $Q_\theta(s,a)$: action-value θ^-: target network γ: discount factor τ_i: quantile level L_κ: Huber loss x: context vector A_a, b_a: arm statistics α: exploration weight $p_a(x)$: LinUCB score

PPO, TRPO, A2C, DQN, and QR-DQN were used for RL comparison, while LinUCB handled context-aware policy mixing.

Director Action Space: MultiDiscrete(4, 4, 4, 4, 4, 4)

Per turn: up to 3 spawn decisions, each with enemy type + target priority

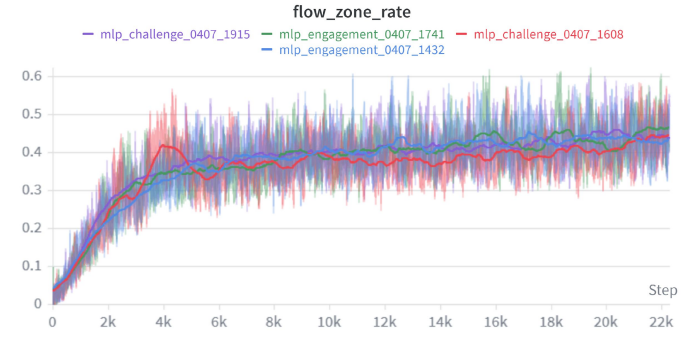
Spawn Slot 1	Spawn Slot 2	Spawn Slot 3
Enemy Type (4 options)	**Enemy Type (4 options)**	**Enemy Type (4 options)**
0: No-op	0: No-op	0: No-op
1: Ranged	1: Ranged	1: Ranged
2: Web	2: Web	2: Web
3: Flying	3: Flying	3: Flying
Target Priority (4 options)	**Target Priority (4 options)**	**Target Priority (4 options)**
0: Nearest	0: Nearest	0: Nearest
1: Player	1: Player	1: Player
2: Building	2: Building	2: Building
3: Weakest	3: Weakest	3: Weakest

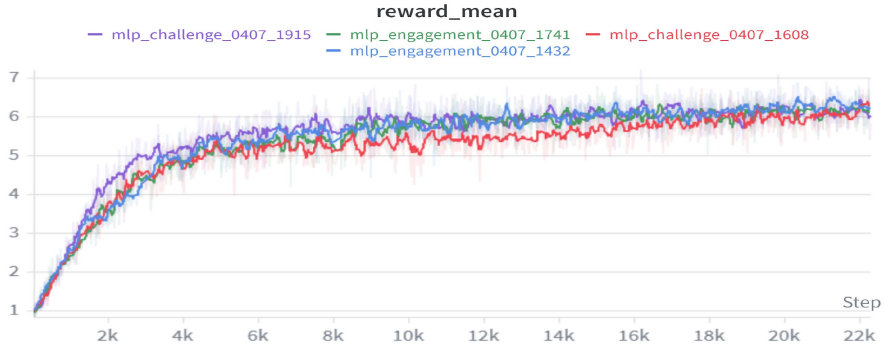
- Total: 4^6 = 4096 combinations per turn.
- Network outputs 6 categorical logit heads
- (softmax per head, sampled during training, argmax at inference)

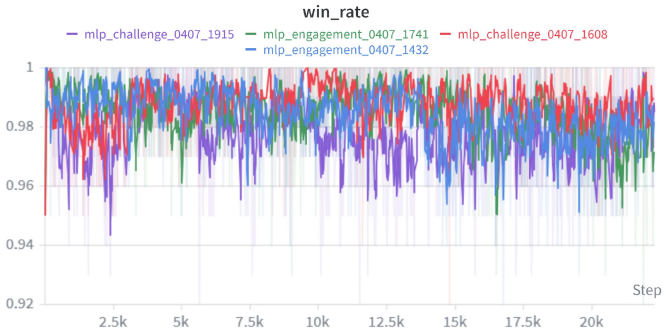
Optuna Hyperparameter Tuning (50 Trials, TPE Sampler)

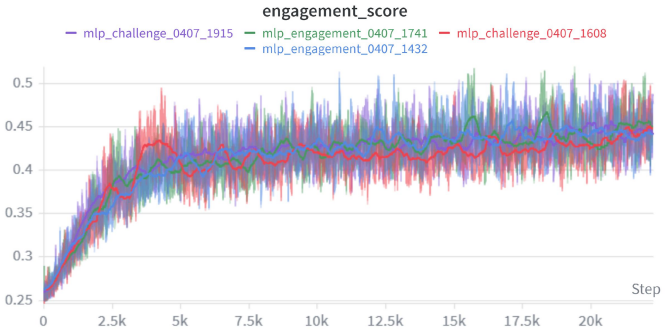
Parameter	Optuna Best	SB3 Default	Why It Matters
Hidden layers	**[128, 256]**	[64, 64]	*534-dim obs needs more capacity*
Learning rate	3.0e-4	3.0e-4	*Matched default (already optimal)*
Batch size	**32**	64	*Smaller batches, more frequent updates*
Rollout (n_steps)	**256**	2,048	*Short episodes (5-10 turns) need fast feedback*
PPO epochs	**7**	10	*Fewer passes avoids overfitting small batches*
Discount (gamma)	**0.9713**	0.99	*Short episodes, prioritize immediate reward*
Clip range	**0.1685**	0.20	*Conservative updates, multi-component reward*
Entropy coeff	**0.0080**	0.0	*Prevents premature convergence*
GAE lambda	0.95	0.95	*Matched default*
Activation	tanh	tanh	*Matched default*

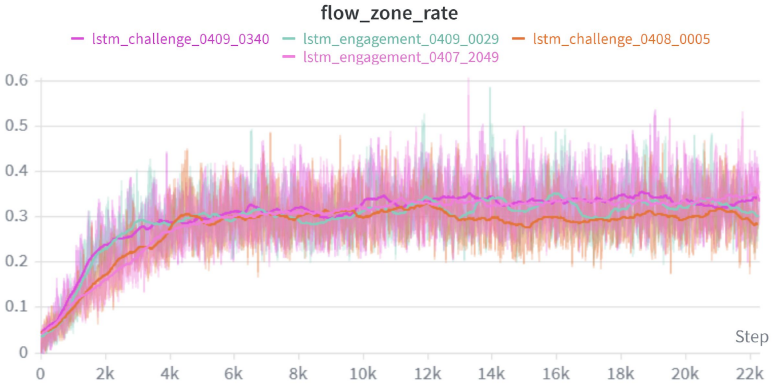
Training Dynamics

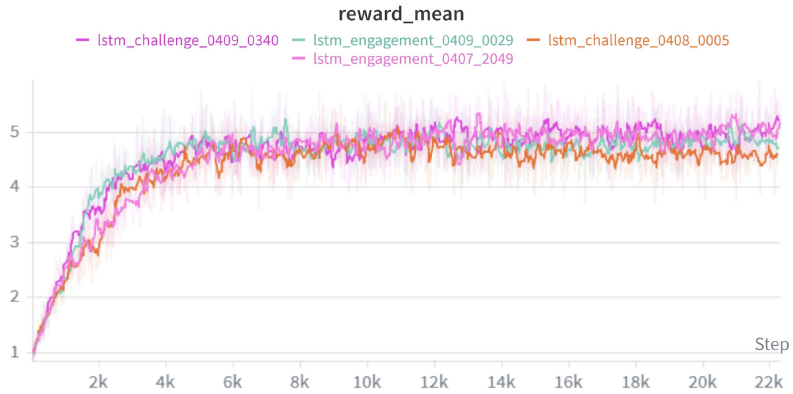


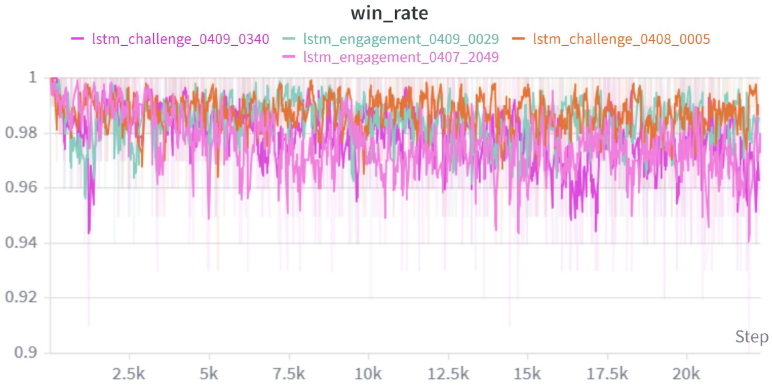


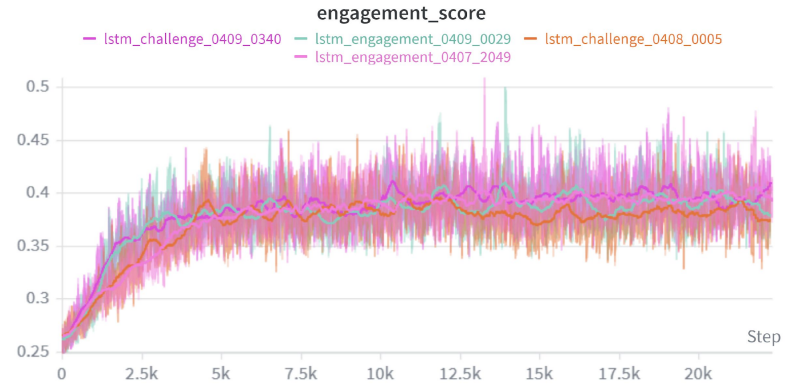





flow_zone_rate
lstm_challenge_0409_0340
lstm_engagement_0409_0029
lstm_challenge_0408_0005
lstm_engagement_0407_2049
0.6
0.5
0.4
0.3
0.2
0.1
0
Step
0
2k
4k
6k
8k
10k
12k
14k
16k
18k
20k
22k


reward_mean
lstm_challenge_0409_0340
lstm_engagement_0409_0029
lstm_challenge_0408_0005
lstm_engagement_0407_2049
5
4
3
2
1
Step
2k
4k
6k
8k
10k
12k
14k
16k
18k
20k
22k


win_rate
lstm_challenge_0409_0340
lstm_engagement_0409_0029
lstm_challenge_0408_0005
lstm_engagement_0407_2049
1
0.98
0.96
0.94
0.92
0.9
Step
2.5k
5k
7.5k
10k
12.5k
15k
17.5k
20k


engagement_score
lstm_challenge_0409_0340
lstm_engagement_0409_0029
lstm_challenge_0408_0005
lstm_engagement_0407_2049
0.5
0.45
0.4
0.35
0.3
0.25
Step
0
2.5k
5k
7.5k
10k
12.5k
15k
17.5k
20k

Limitations

- **Simulation-only results:** Evaluation was performed in the Gymnasium training environment rather than the live game engine, so a sim-to-real gap remains.
- **Proxy player too strong:** Its high win rate suppressed close-call events, limiting the practical ceiling of engagement.
- **No human validation:** Engagement was validated mathematically, not against subjective player experience.
- **Prior work:** Existing DDA research lacked open-source environments, datasets, and baselines, so much of the system had to be built from scratch.
- **Game dependency:** The evolving undergraduate game API and mechanics required repeated redesigns to keep the environment aligned.


thank you