Lab 4 Report
Elijah Yoo, Ethan Wong

How We Split Up the Work

Eli focused on writing the core logic and implementing key features, such as the skeleton for the tic tac toe game as well as the implementing the STM, and the learning process for the tic tac toe "AI"

Ethan worked on mainly the user interface as well as organization of the code. Worked on debugging the code and fixing up the structure of the connection between the STM and the game, allowing the game to run properly with user interactions. Finally ensuring that it was efficient, and organized.

We both worked together to write the report and README files. Throughout the process we worked together at the same time to code up the project so we really did a little bit of everything.

Report

This report presents our observations of how our AI agent behaved and evolved when trained using a Collective Learning System (CLS) under various combinations of reward and punishment settings. After confirming that the program functioned as expected, we tested several combinations of betaReward and betaPunish values to evaluate how well the AI could learn through gameplay.

Over multiple sessions, we tested five different beta configurations and ranked them based on how effectively the AI learned to play Tic Tac Toe:

1. **Mixed Beta Values** – R: 0.5, P: 0.25
2. **Initial Testing** – R: 0.5, P: 0.5
3. **Reward Only** – R: 0.5, P: 0
4. **Low Beta Values** – R: 0.25, P: 0.25
5. **High Beta Values** – R: 0.75, P: 0.75

Below are detailed notes from each trial.

Initial Testing

betaReward = 0.5, betaPunish = 0.5

From the beginning, we challenged the AI with full effort, making no concessions in gameplay. As expected, the AI lost the first few matches easily. However, once it successfully blocked a winning move and received positive reinforcement, its performance improved rapidly. It began consistently selecting the center as its opening move and became difficult to beat. When using standard strategies, the best we could do was force a draw.

Despite this progress, we noticed significant weaknesses. The AI struggled when we introduced unpredictable patterns. For example, when we allowed it an easy win scenario it hadn't previously seen, it failed to capitalize on it. This showed that the AI was heavily reliant on prior exposure and lacked the generalization needed to adapt to unfamiliar setups. Interestingly, unconventional play such as intentionally suboptimal moves often threw it off entirely.

Low Beta Values

betaReward = 0.25, betaPunish = 0.25

With lower learning rates, training became noticeably slower and more frustrating. It took the AI 17 games to achieve a single draw, and progress remained sluggish even after that. Occasionally, it stumbled into a workable strategy, but often it would stray from effective plays.

A core limitation was that the AI evaluated games as all-or-nothing. A win meant all moves were considered good; a loss meant they were all bad without distinguishing specific decisions. We briefly considered letting the AI win when it made a smart move to accelerate learning, but doing so would have introduced supervision, undermining the core premise of CLS. We stuck to playing optimally and observed how the AI would adapt under consistent pressure.

High Beta Values

betaReward = 0.75, betaPunish = 0.75

These aggressive values made the AI hypersensitive to results. If it found success with one approach, such as starting on a side space and forcing a draw, it would quickly commit to that path. But after a single loss, it would immediately abandon the strategy and move on to something entirely different.

This led to a kind of frantic cycling constantly switching tactics without staying long enough to refine any of them. The AI couldn’t form lasting patterns because punishment undid its progress too quickly. Ultimately, it got stuck in a loop of unproductive experimentation, and we ended the trial when it became clear the system couldn’t retain meaningful strategies under such volatile conditions.

Reward-Only Strategy

betaReward = 0.5, betaPunish = 0

We tested whether positive reinforcement alone could produce useful learning. At first, the AI made random moves and didn’t learn from failure. However, it eventually stumbled into a win by starting with a side move and began focusing exclusively on that pattern.

Without punishment to guide it away from poor outcomes, it clung to that one win as if it were the only viable strategy. Even repeated losses didn’t deter it. Over time, the AI learned to respond effectively to variations of that opening, becoming surprisingly competent within that narrow scope.

This behavior was both impressive and concerning. The AI developed deep confidence in a single tactic but lacked flexibility. Even when that tactic failed, it wouldn’t let it go. The result was an inflexible but persistent learner able to improve within one lane but incapable of generalizing across the full game space.

Mixed Beta Values

betaReward = 0.5, betaPunish = 0.25

This configuration delivered the strongest learning results. It mirrored the balanced approach in some ways but incorporated a slightly reduced penalty for failure, which made a notable difference.

After around ten games, the AI consistently chose a center-first strategy and learned to handle many follow-up scenarios with competence. Even when we disrupted its approach or intentionally confused it, the AI recovered quickly and didn’t throw away successful patterns.

This configuration struck a productive balance: reward reinforced progress, while punishment provided useful feedback without completely erasing learned behavior. The AI became increasingly effective and was difficult to beat even when we used well-practiced strategies. In the end, this model demonstrated the best mix of adaptability, resilience, and long-term learning.

Reflection

This lab challenged how we think about Tic Tac Toe as a game. Normally, it’s seen as a solved system, but when facing a learning agent like this, strategy becomes fluid. We couldn’t rely solely on standard tactics we had to constantly adapt based on how the AI evolved.

Over time, we weren’t just playing Tic Tac Toe; we were also studying the AI’s behavior and identifying its blind spots. It became a battle of adaptation: the AI learned to counter our moves, and we, in turn, learned to exploit the limits of its current understanding.

Eventually, the AI became remarkably strong, gradually eliminating every strategy we used to beat it. At a certain point, it felt like the only way to stop it from becoming unbeatable was to stop teaching it altogether.