

Scalable CLA Strategies in Snafu

Michael Cloppert (*cloppemj@gwu.edu*)
The George Washington University

December 19, 2007

1 Abstract

This paper investigates scalable collective learning automaton strategies in the game *Snafu*. It is shown that training an automaton on a small playing space can yield significant improvements over an identical, untrained automaton on a larger playing space. Results are achieved through reduction of selected features to unit vectors, using a cosine similarity function to guess the best move absent any identical state transition matrix match.

2 Introduction

Steven Lisberger’s 1982 sci-fi thriller *Tron* brought to life a variety of 8-bit video games popular at the time by immersing the protagonist in the game itself, and anthropomorphizing the computer opponent as the antagonist. In one fast-paced scene, an early video game is realized as players riding “Light Cycles;” motorcycles that leave a wall in their trail. The game winner is the player who stays alive longest without hitting the perimeter wall, or the trailing walls from other cycles (Fig. 1).

This scene was inspired by a number of similar games available for early 8-bit systems. Common names included *Snakes*, or *Snafu*. This paper investigates training a variety of collective learning automata to compete in *Snafu* against a “pseudo-random” opponent representative of those provided by the original video games.

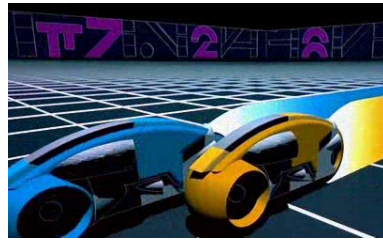


Figure 1: Light Cycle scene from the movie *Tron*

For the purpose of this experiment, the following properties and rules are enforced in the *Snafu* environment state machine (ESM):

- The game is played on an $M \times N$ -sized board.
- The game consists of two players.
- Both players begin the game at random locations, facing a random direction.
- Valid directions of travel are 0° , 90° , 180° , and 270° relative to the orientation of the board.
- Each round, players may choose to continue progressing in the same direction (straight), or turn 90° left or right.
- Each player advances in the selected direction at the same speed: 1 square per round.
- Both players know the state of the board, locations of walls, and locations of the opponent.
- Players do *not* know the direction their opponent selects for the next round until the round advances.

- When advancing rounds, a wall is created in the last position each player assumed.
- The game ends when one player hits a wall or goes out of the $M \times N$ playing space.

As M and N increase, the CLA’s state-transition matrix (STM) grows quickly, decreasing the frequency of identical matches between the automaton’s state and the STM. This significantly hinders learning. This paper explores CLA strategies that will enable robust learning and STM population on a smaller playing surface which can then be applied to a larger playing surface to realize a meaningful win-rate over the opponent.

3 Hypothesis

Creating a scalable STM necessitates update and selection policies that will apply regardless of the $M \times N$ dimensions of the playing surface. A normalized update policy is achieved by reducing all feature measurements to unit vectors where possible. A generalized selection policy is accomplished by computing the cosine similarity (Equation 1) between the automaton’s present, unit-vector-reduced state and each STM entry, selecting the STM entry to which the automaton’s state has the highest similarity measure.

$$csim(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{||\vec{x}|| * ||\vec{y}||} \quad (1)$$

With carefully-selected features, update strategies, and parameters, it can be shown that narrowly-winning strategies learned on small playing surfaces will yield statistically-significant, otherwise unachievable win margins when applied to larger surfaces.

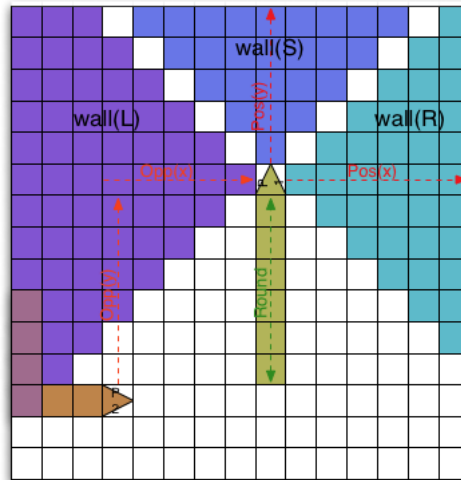


Figure 2: Illustration of features considered in this experiment

4 Experiment Design

4.1 The Opponent

The opponent used in this experiment is a pseudo-random, "mindless" opponent. The opponent adheres to the following two rules when deciding when and which direction to turn:

1. Keep going the same direction if no wall is straight ahead
2. If a wall is straight ahead, randomly choose amongst the directions that will not lead to another wall.

This is representative of the typical opponent provided in early implementations of *Snafu*.

4.2 The Collective Learning Automaton

The following four features, in various combinations, are considered for the CLA (Fig. 2):

1. **Automaton Position (P):** The position of the automaton relative to perimeter walls straight ahead, and to the right, of the present heading. Each is divided by its corresponding board dimension to normalize the measure as a unit vector.
2. **Opponent Position (O):** The x and y position of the opponent relative to the automaton’s heading, with positive x measured as the distance 90° to the automaton’s right, and positive y measured as the distance straight ahead. In the example Figure 2, both x and y are negative for this measure. Each is divided by its corresponding board dimension to normalize the measure as a unit vector.
3. **Wall Saturation (W):** The percent saturation of the board ahead, 90° to the left, and 90° to the right of the automaton’s heading, as illustrated in Figure 2.
4. **Round Number (R):** The round number within the game. As the final number of rounds cannot be determined during the game, this is not normalized to a unit vector.

The update function and reward policies are relatively straightforward compared to the other CLA components. The contents of each STM entry are integer values associated with each of the three decisions that can be made from any position: *continue straight*, *turn right*, and *turn left*. The primary consideration for reward selection in this experiment was *what is the relative value of a tie to a win or loss*. There are two components to this: first, the ratio of win-value to tie-value, and second, should a tie be rewarded or discouraged (positive or negative value). To this end, two reward policies were considered, as illustrated in Table 1. This value is directly applied by the update function to the STM upon completion of each game, based on the game’s outcome. No interim updates are made while the game was in progress.

Table 1: Reward Policies Considered

Policy	Win	Lose	Tie
1	+3	-3	+1
2	+4	-4	-1

4.3 Selection Function

The selection function is the critical component that enables scalability of learned strategies to larger boards. Each round, the selection function compares the current feature measurements of the automaton’s environment to every entry in the STM using a cosine similarity function (Eqn 1). To reduce false selection of dissimilar states in a sparse STM, a minimum cosine similarity must be met to consider the CLA’s state a “match.” Further, in order to control confidence in the STM entry selected by the cosine similarity function, a threshold must be met by the direction selected from the STM with respect to other directions in the same entry. Therefore, a match and subsequent action is taken based on the following two parameters:

1. A minimum c -value. The cosine similarity measure must meet or exceed this value to be considered a “match.”
2. A minimum t -value. This is the threshold by which one of the decisions in the selected STM entry must exceed the other two decisions if it is to be used.

If no STM entry exists whose similarity to the current environment exceeds c , or if no decision in the closest cosine similarity measure is more than t greater than both other decisions, the automaton functions identically to its pseudo-random opponent.

The architecture of this CLA is illustrated in Figure 3. From various combinations of reward policies, selection policy parameters, and feature combinations, a set of candidate strategies were identified for study that provided modest win percentage gains on a small, 15x15 board, over 1000 games. Those strategies are listed in Table

2.

Table 2: CLA Strategies Tested

Policy Set	t	c	Reward (W,L,T)
WR	12	0.5	+4, -4, -1
PR	3	0.5	+3, -3, +1
PR	9	0.5	+3, -3, +1
PO	4	0.5	+4, -4, -1
POWR	3	0.5	+3, -3, +1
POWR	9	0.5	+3, -3, +1

To evaluate scalability, a strategy is executed on a 15x15 board for 1,000 games. The resultant STM is stored, and then loaded into an identically-configured CLA on a 30x30 board. This trained CLA is played against the pseudo-random opponent another 500 games. The effectiveness of the trained CLA is established by comparing its success to that of an independent, identical CLA without the training STM played on the same sized board. "Success" is measured as the comparison of the CLA loss frequency to the opponent loss frequency (Eqn 2). The implication of this measure is that a game outcome of "tie" neither hurts nor helps either player.

$$success = \frac{losses_{opponent}}{games} - \frac{losses_{CLA}}{games} \quad (2)$$

This test procedure and success measurement is repeated 100 times for each strategy. From these iterations, a mean (μ) and standard deviation (σ) are computed. The success of the strategy is plotted as a function of the expected win percentage after a certain number of consecutive games are played.

Success of a strategy's scaling from a small to large board can be claimed if the trained CLA's asymptote exceeds that of the untrained CLA on the 30x30 board with a confidence of greater than 95%; i.e., $\alpha = 0.05$ [1, 2], or the critical probability $p^* = 1 - \alpha/2 = 0.9750$ [3]. The equivalent normal Z-score between the two strategies, as computed by Equation 3, must therefore exceed 1.96 [3]. If the z score does not exceed this value, no statistically-significant conclusion can

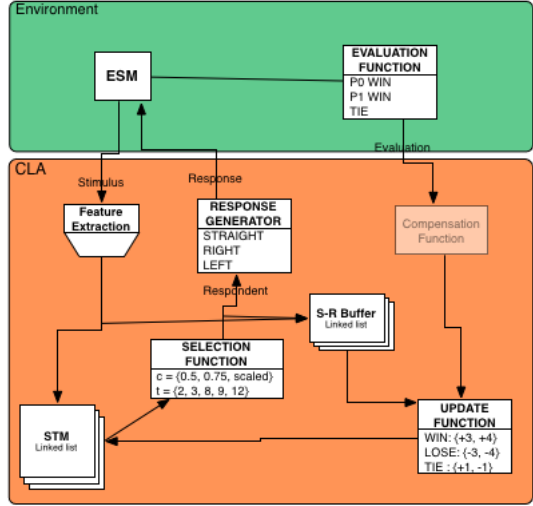


Figure 3: *Snafu* Architecture

be drawn and the strategies are said to be equivalent.

$$z = \frac{p_1 - p_2}{\sqrt{\frac{p_1 q_1}{n_1} + \frac{p_2 q_2}{n_2}}} \quad (3)$$

5 Results

5.1 Wall-Round Feature Set

This strategy set can be described as one based on wall saturation around the automaton as the game progresses. Figure 4 illustrates the success of the CLA training on a 15x15 board with $c = 0.5$, $t = 12$, and negative reward for tie. While the win ratio of the CLA asymptotes to just over 2%, it does so with a high standard deviation over the 100 iterations and therefore is winning, but with low confidence. The STM growth rate has slowed by the 1000th game, implying frequent updates to existing STM entries and a robust knowledge base.

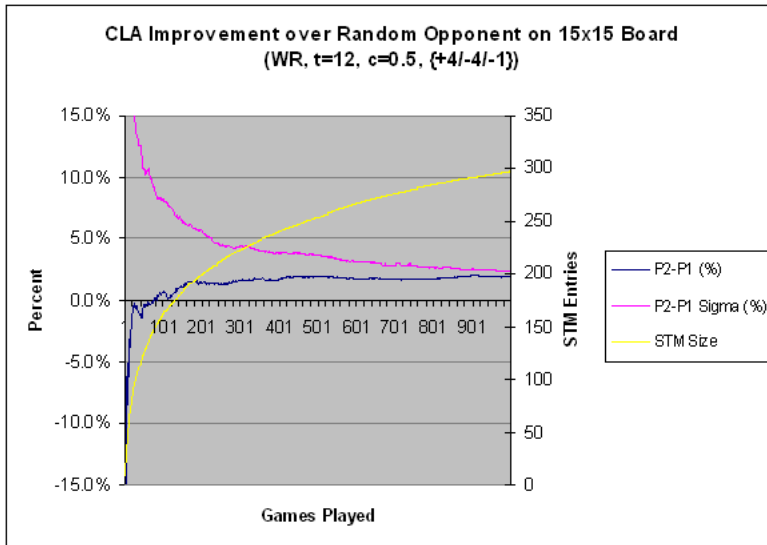


Figure 4: Wall and Round features, $c=0.5$, $t=12$, negative reward for tie

Figure 5 shows the success of the trained CLA against an untrained, identically-equivalent CLA on the 30x30 board. While the success of the CLA on the small board was modest and unpredictable, the trained automaton performs far better on the 30x30 board than its untrained counterpart with a high level of confidence.

5.2 Position-Round Feature Set

This strategy can be described as one based on board position of the automaton as the game progresses. Figure 6 illustrates the success of the CLA training on a 15x15 board with $c = 0.5$, $t = 3$, and positive reward for tie. While the win ratio of the CLA asymptotes to just over 2%, as with the prior strategy, it does so with a high standard deviation over the 100 iterations and therefore is winning, but with low confidence. The STM growth rate has slowed by the 1000th game, again implying frequent updates to existing STM entries and a robust knowledge base.

Figure 7 shows the success of the trained CLA

against an untrained, identically-equivalent CLA on the 30x30 board. While the success of the CLA on the small board was modest and unpredictable, the trained automaton again performs far better on the 30x30 board than its untrained counterpart with a high level of confidence.

The results achieved when adjusting $t = 9$ closely mirrored those for $c = 3$. The resultant trained-untrained plot is provided in the appendix (Figure 12).

5.3 Position-Opponent Feature Set

This strategy set can be described as one based on automaton position and distance from opponent, regardless of round or wall saturation. Figure 8 illustrates the success of the CLA training on a 15x15 board with $c = 0.5$, $t = 4$, and negative reward for tie. Once again the win ratio of the CLA asymptotes to just over 2% with a high standard deviation over the 100 iterations and therefore is winning, but with low confidence. Note that, in contrast with previous ex-

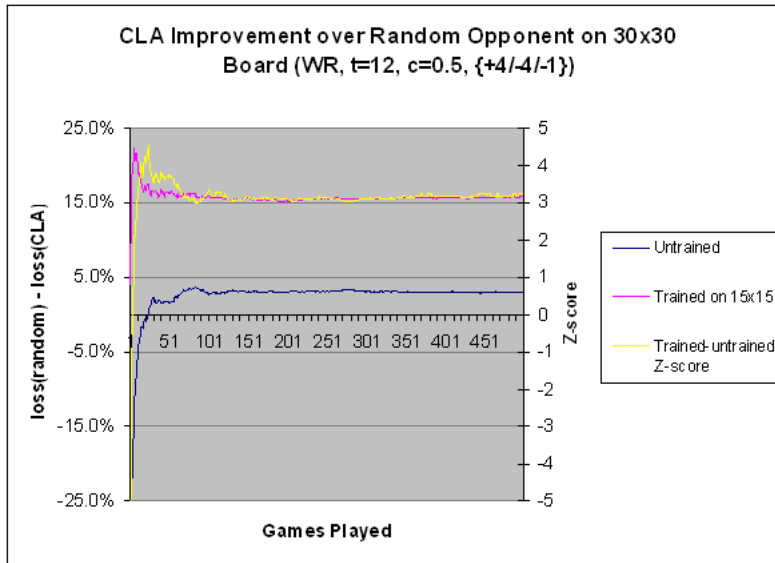


Figure 5: Success of CLA trained from 15x15 board, WR feature set, $c=0.5$, $t=12$.

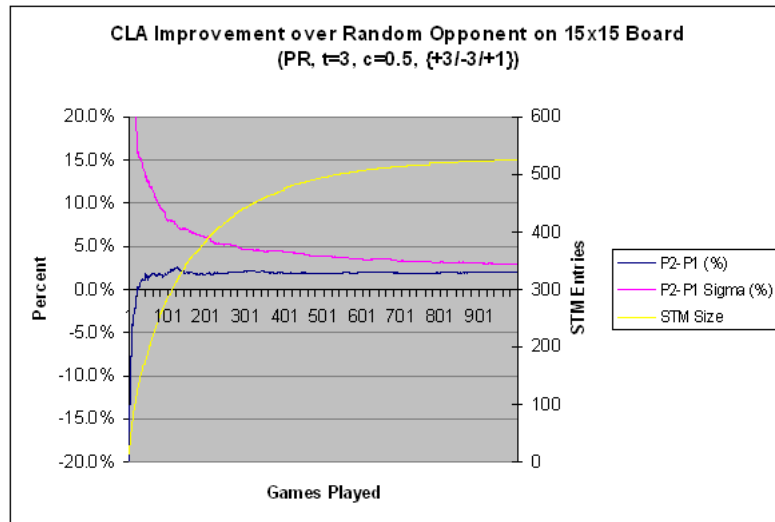


Figure 6: Position and Round features, $c=0.5$, $t=3$, positive reward for tie

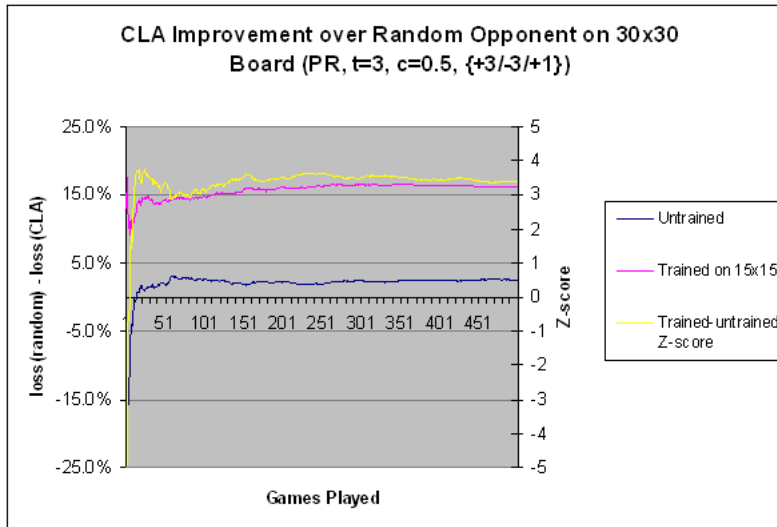


Figure 7: Success of CLA trained from 15x15 board, PR feature set, $c=0.5$, $t=3$.

periments, the STM continues to grow almost linearly at the 1000th game, indicating the STM is not saturated.

Figure 9 shows the success of the trained CLA against an untrained, identically-equivalent CLA on the 30x30 board. Despite the observation that the STM on the smaller board had not neared saturation by the 1000th game, and that the success of the CLA on the small board was modest and unpredictable, the trained automaton again performs far better on the 30x30 board than its untrained counterpart with a high level of confidence.

5.4 Position-Opponent-Wall-Round Feature Set

This strategy can be described as one taking into account all noted features. Figure 10 illustrates the success of the CLA training on a 15x15 board with $c = 0.5$, $t = 3$, and positive reward for tie. The win ratio of the CLA asymptotes to zero with a high standard deviation. In this case, it is hard to tell whether the strategy is useful on

the smaller board or not. The STM growth rate has not slowed by the 1000th game, indicating the STM is not saturated. By observation of this data, the combination of features and parameters appears to be suboptimal.

Figure 11 shows the success of the trained CLA against an untrained, identically-equivalent CLA on the 30x30 board. While the success of the CLA on the small board was dubious, the trained automaton still performs far better on the 30x30 board than its untrained counterpart with a high level of confidence.

The results achieved when adjusting $t = 9$ closely mirrored those for $c = 3$. The resultant trained-untrained plot is provided in the appendix (Figure 13).

6 Conclusion

Small board dimensions, it was discovered, make for a difficult learning environment for the automaton, and low statistical significance in winning measures. However, when scaled to a larger

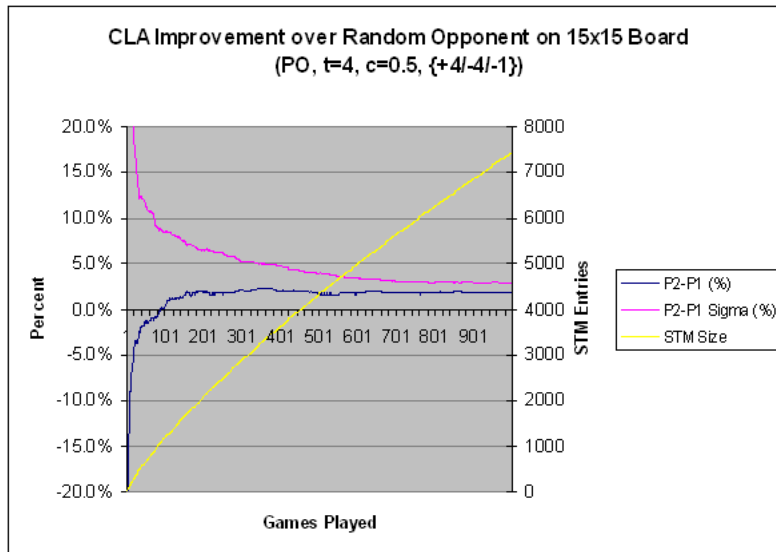


Figure 8: Position and Opponent features, $c=0.5$, $t=4$, negative reward for tie

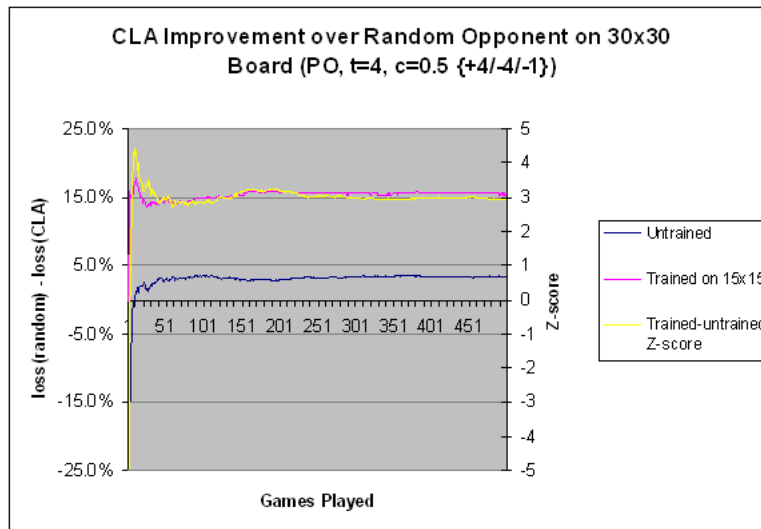


Figure 9: Success of CLA trained from 15x15 board, PO feature set, $c=0.5$, $t=4$.

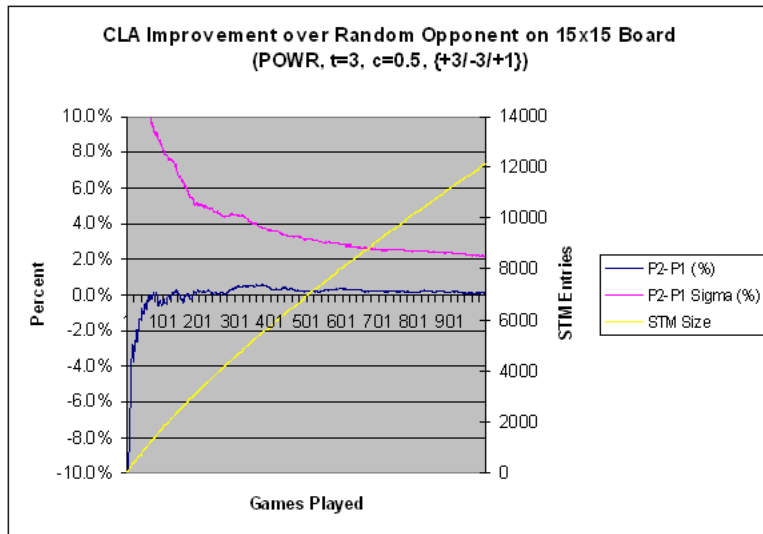


Figure 10: All features, $c=0.5$, $t=3$, positive reward for tie

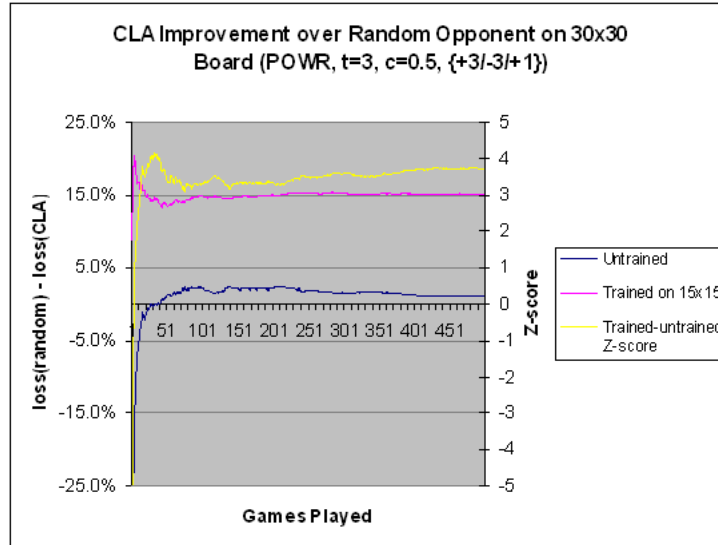


Figure 11: Success of CLA trained from 15x15 board, entire feature set, $c=0.5$, $t=3$.

30x30 board, the strategy learned proves highly effective. This holds true for knowledge bases that approached saturation, as well as those that do not, implying that even sparsely-populated knowledge bases can provide a significant advantage when given to CLA's playing *Snafu* on much larger boards against pseudo-random opponents. An interesting follow-up to this study could explore the lower bound of this theory.

References

- [1] Mark Happel. Various class notes, 2007.
- [2] Peter Bock. *The Emergence of Artificial Cognition*. World Scientific Publishing Co., 1993.
- [3] Stat Trek Inc. Statistics tutorial: Confidence interval for difference between means. <http://stattrek.com/AP-Statistics-4/Difference-Means.aspx?Tutorial=AP>, 2007.

Appendices

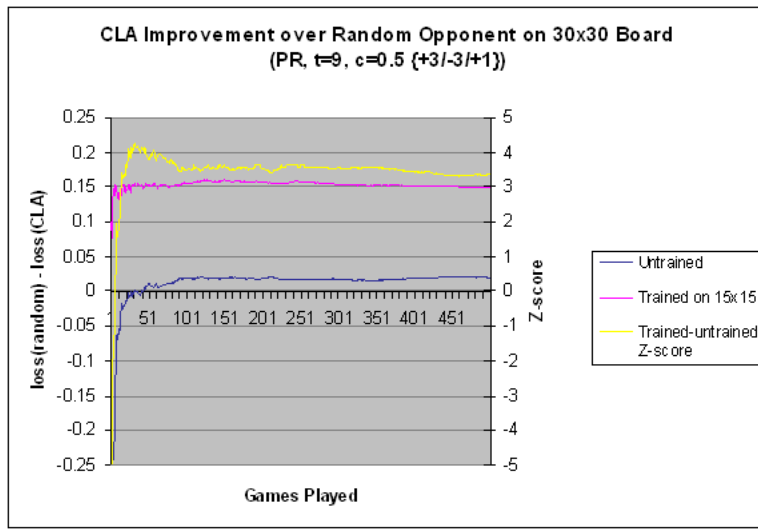


Figure 12: Success of CLA trained from 15x15 board, PR feature set, $c=0.5$, $t=9$.

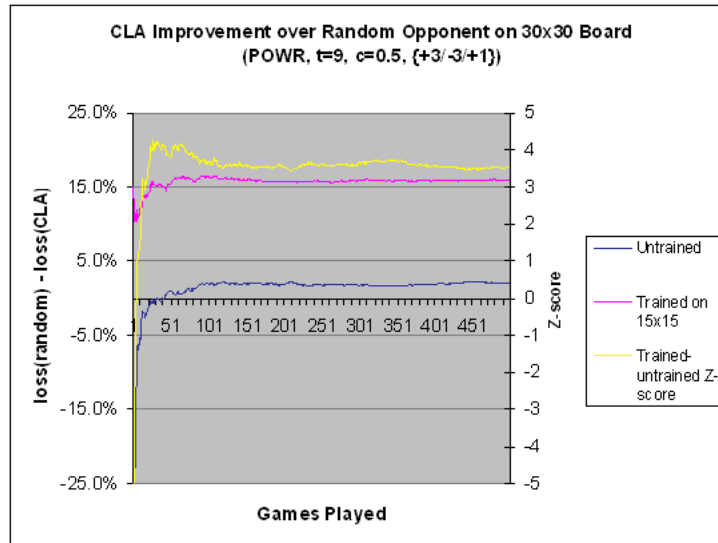


Figure 13: Success of CLA trained from 15x15 board, entire feature set, $c=0.5$, $t=3$.