Optimising Deep Learning and Search for Imperfect-Information Games in General Game Playing

PRESENTED BY: JOEL WEST (Z5311058)

SUPERVISOR: MICHAEL THIELSCHER

ASSESSOR: ALAN BLAIR

Executive Summary

- A previous student, Zachary Partridge, proposed a framework for a general imperfect information game player
- My aim is to
  - Optimise the frameworks speed to search games more extensively and play larger games
  - Improve reliability and documentation for future students

Outline

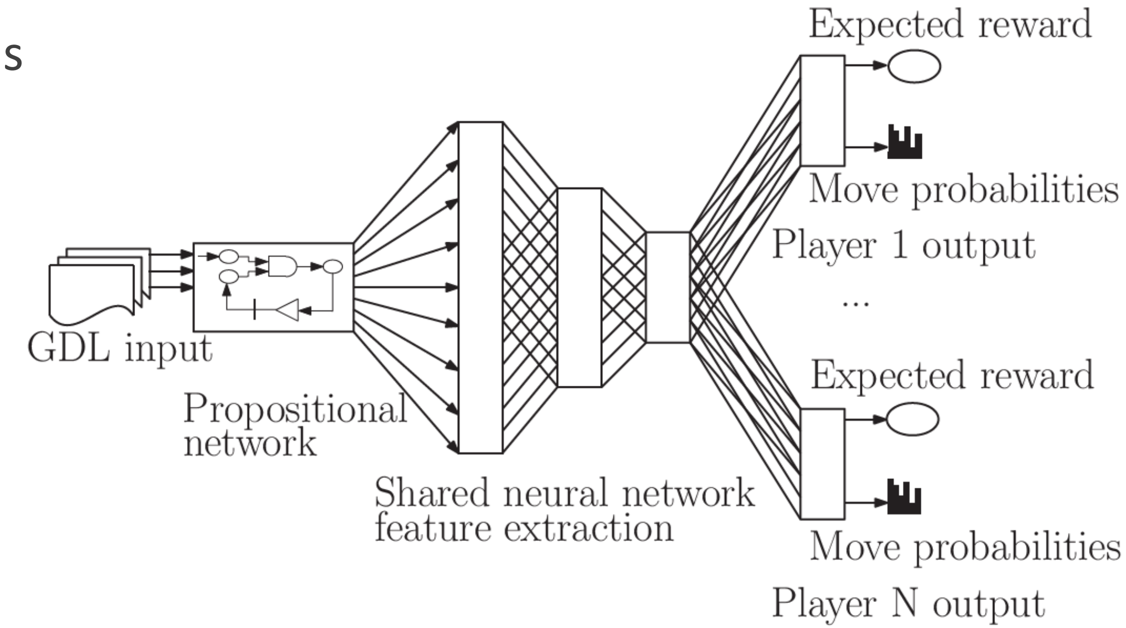
- Background
- Methodology
- Results
- Conclusion

Outline

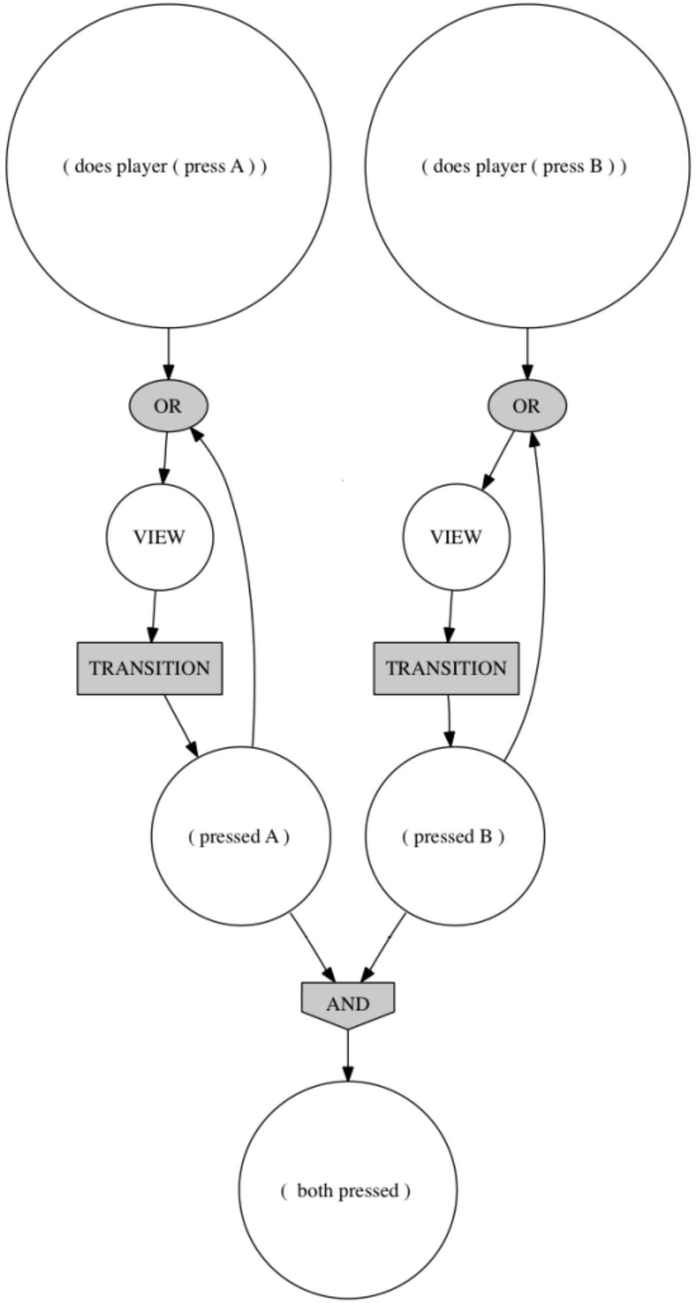
- Background
  - Generalized AlphaZero
  - Counterfactual Regret Minimisation
  - ReBeL
  - Previous Framework
- Methodology
- Results
- Conclusion

Generalised AlphaZero

- AlphaZero performs a depth limited MCTS and uses a neural network to estimate the values of leaf nodes
- AlphaZero operates under a set of assumptions that don't work for GGP
- Generalised AlphaZero extends AlphaZero for any perfect information game



Shared network from Genesereth & Thielscher



Simple example of a game where two buttons can be pressed, but never unpressed from Cox et al.

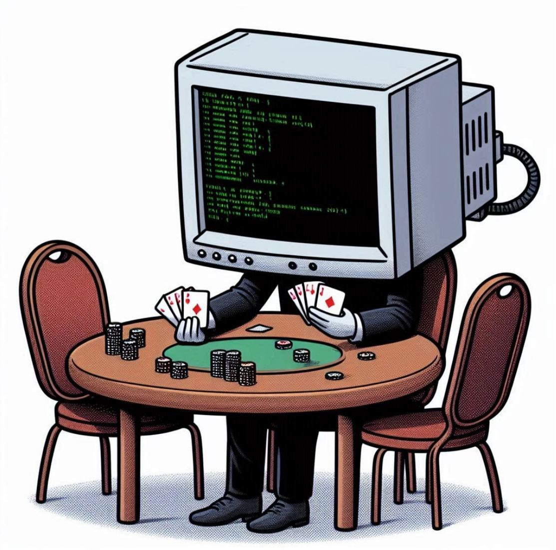
- Propositional networks provide an interface for general game players
- Describes games via a network of Boolean logic

Counterfactual Regret Minimisation (CFR)

- CFR is an iterative stochastic descent algorithm for imperfect information games
- The counterfactual regret of an action is the expected utility lost in hindsight by playing a strategy as opposed to always playing said action
- On each iteration, CFR calculates regrets and strategies for each information set
- The average strategy played converges to a Nash equilibrium
- There are many other variants of CFR
  - CFR-Decomposition (CFR-D)
  - Monte Carlo CFR (MCCFR, of which there are many further subvariants…)

ReBeL

- Extremely strong HUNL poker and liar's dice framework from Brown et al.
- Requires no expert knowledge
- A public belief state is maintained using publicly available information (assumed to include opponent actions) from which states are sampled from
- A depth limited CFR-D search is conducted on each sample
- At the depth limit, leaf node policies are estimated using a neural network
- Provably converges to a Nash equilibrium for 2 player zero sum games



Previous Framework by Zachary Partridge

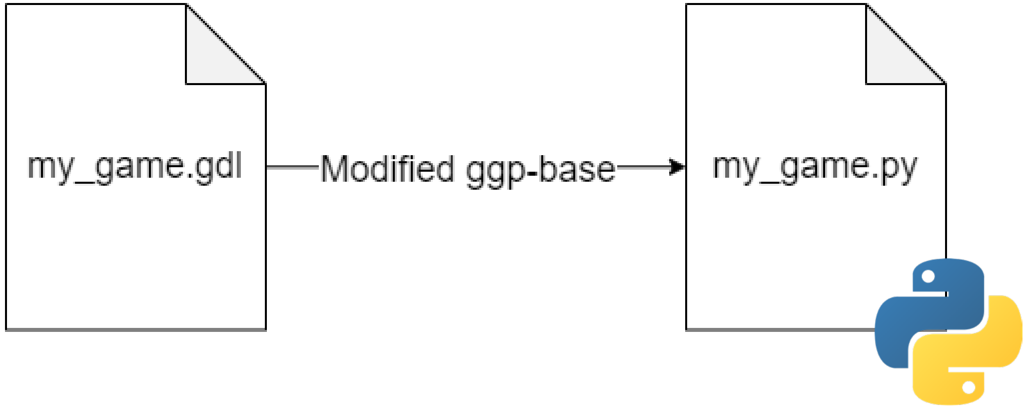
- Combines concepts from both ReBeL and Generalised AlphaZero
- Implemented in Python
- Extended previous work `ggp-base` to create a propositional network for GDL-II games
- Outlines methodology of sampling game states from a history of actions/observations
  - Games are randomly played from the initial state
  - “Invalid” states are saved to an LRU cache to avoid unnecessary computation
- Uses a depth limited Vanilla CFR to search sampled states
- A neural network
  - Estimates the value of states at depth limit
  - Provides the initial policy to the CFR search

Outline

- Background
- Methodology
  - Propositional Network Reformatting
  - Reimplementation of Framework in C++
  - Use of MCCFR
  - Parallelisation of Key Areas
- Results
- Conclusion

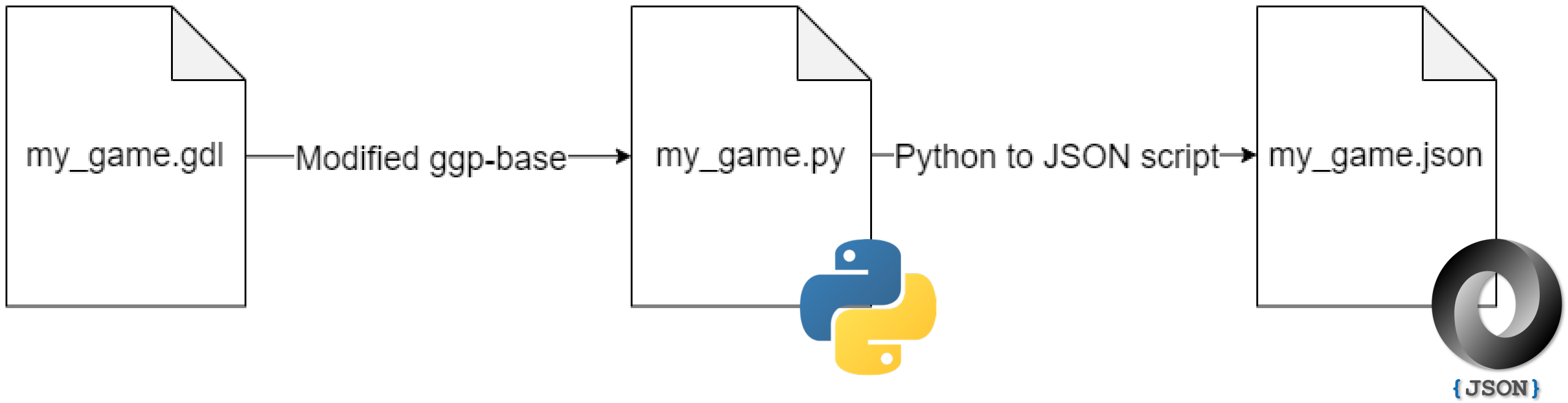
Propositional Network Reformatting

- Previous frameworks store propositional networks as Python files
- Files are dynamically loaded as modules
- Sufficient for Python programs, but otherwise tedious



Propositional Network Reformatting

- A Python script transforms Python files into JSON
- Additionally, further computation is done (e.g., computing the topological ordering, separating pre- and post-transition nodes, …)

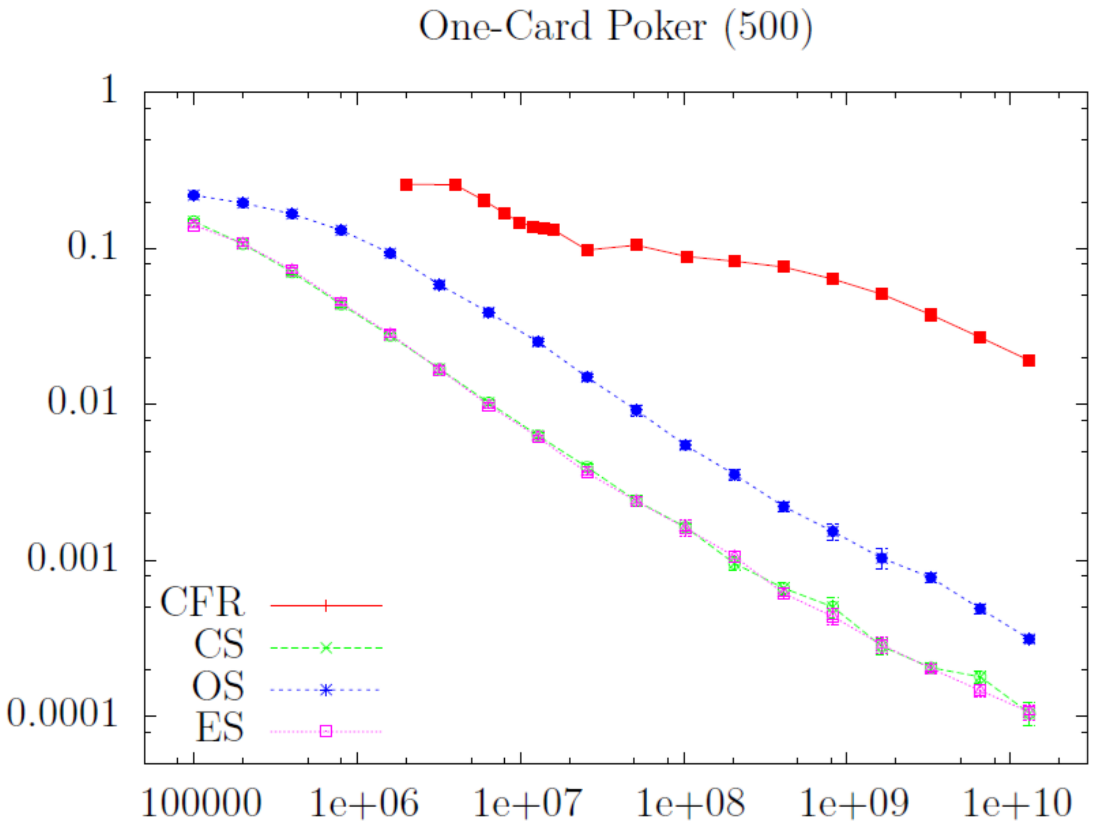


Reimplementation of Framework in C++

- C++ is the predominant language for game playing due to its speed and mature libraries
- Reimplementation includes everything
  - Propositional network
  - Sampling methods
  - CFR
  - Training loop

Use of MCCFR

- Original framework uses Vanilla CFR
- Updated framework uses MCCFR with external sampling
- New implementation uses MCCFR for two reasons
  - Achieves significantly faster convergence times (Lanctot, 2013) particularly in games with high branching factors
  - Using external sampling, reach probabilities don't need to be calculated which simplifies implementation



Comparison of Exploitability (y-axis) Versus Number of Nodes Visited (x-axis) of MCCFR and Vanilla CFR in One-Card from Lanctot

Parallelisation of Key Areas

- Original framework ran single threaded
- State sampling
  - Invalid state caches are shared
  - Threads concurrently sample, search and add results to a buffer
- Training loop
  - Complete search and agents concurrently search the current state
  - Current work includes playing multiple training games in parallel

Outline

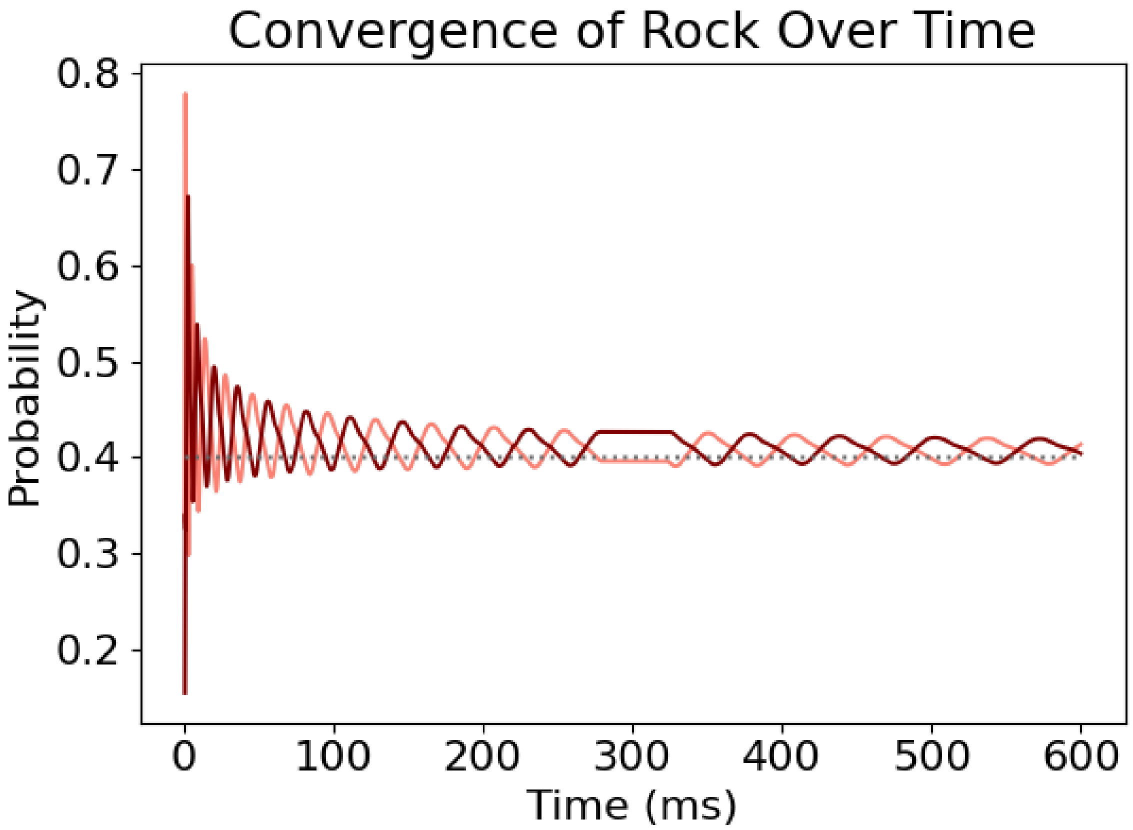
- Background
- Methodology
- Results
  - Scissor Paper Rock+
  - Blind Tic-Tac-Toe
- Conclusion

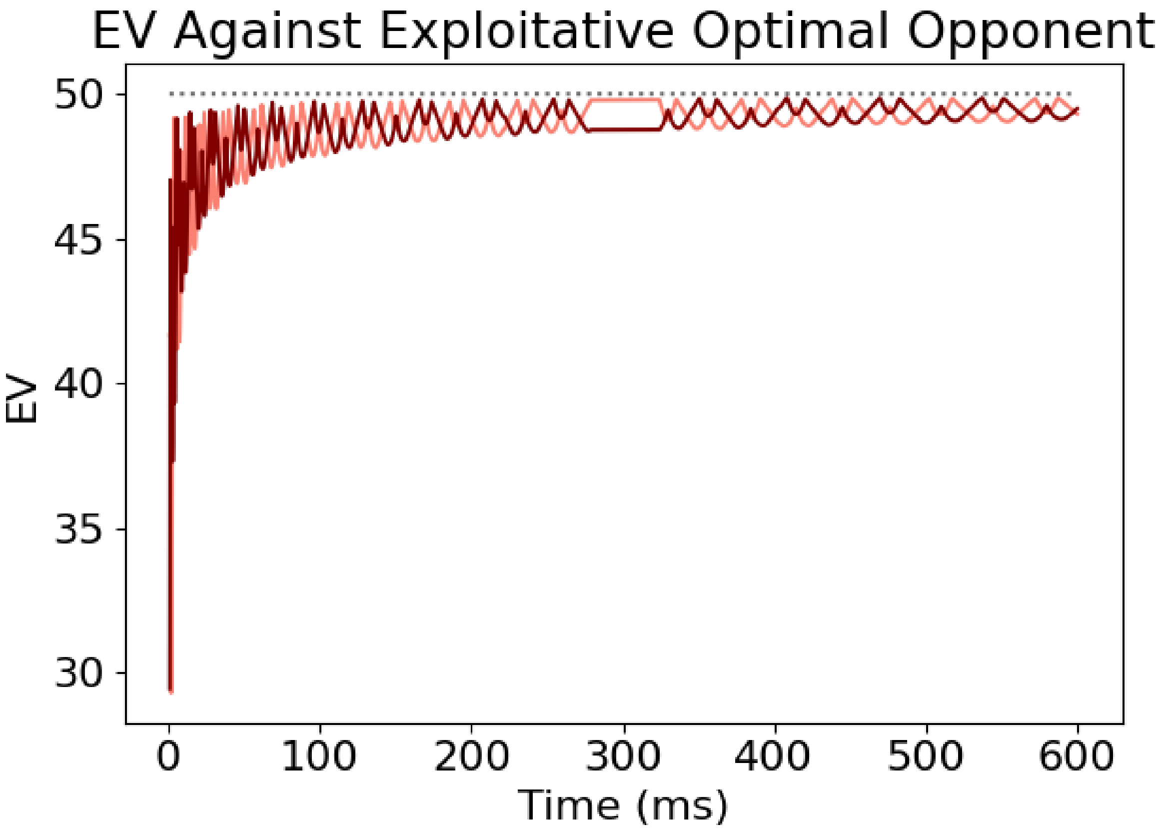
Scissor Paper Rock+ - Overview

- Simple two player zero-sum game
- Players must simultaneously choose between scissors, paper or rock
- Rewards are asymmetric (e.g., paper beating rock nets 75 points whereas rock beating scissors nets 100)

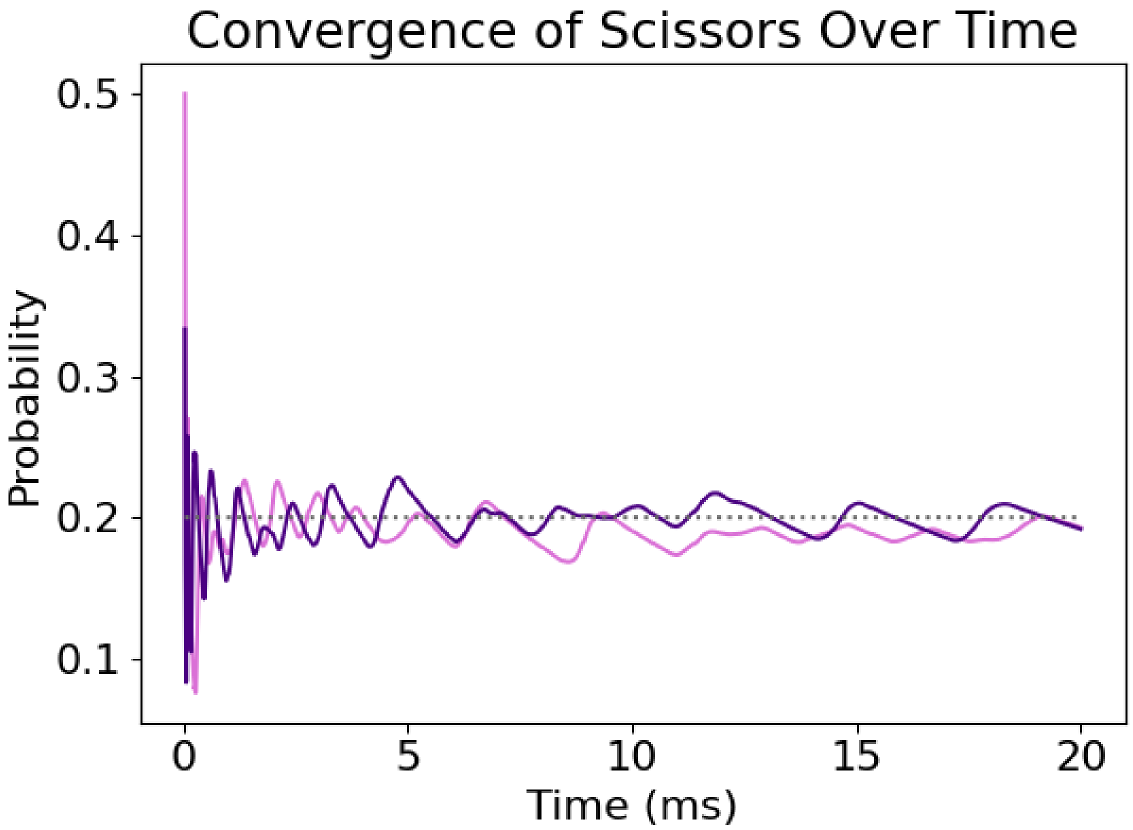
		Player 2		
		Scissors	Paper	Rock
Player 1	Scissor	50, 50	100, 0	0, 100
	Paper	0, 100	50, 50	75, 25
	Rock	100, 0	25, 75	50, 50

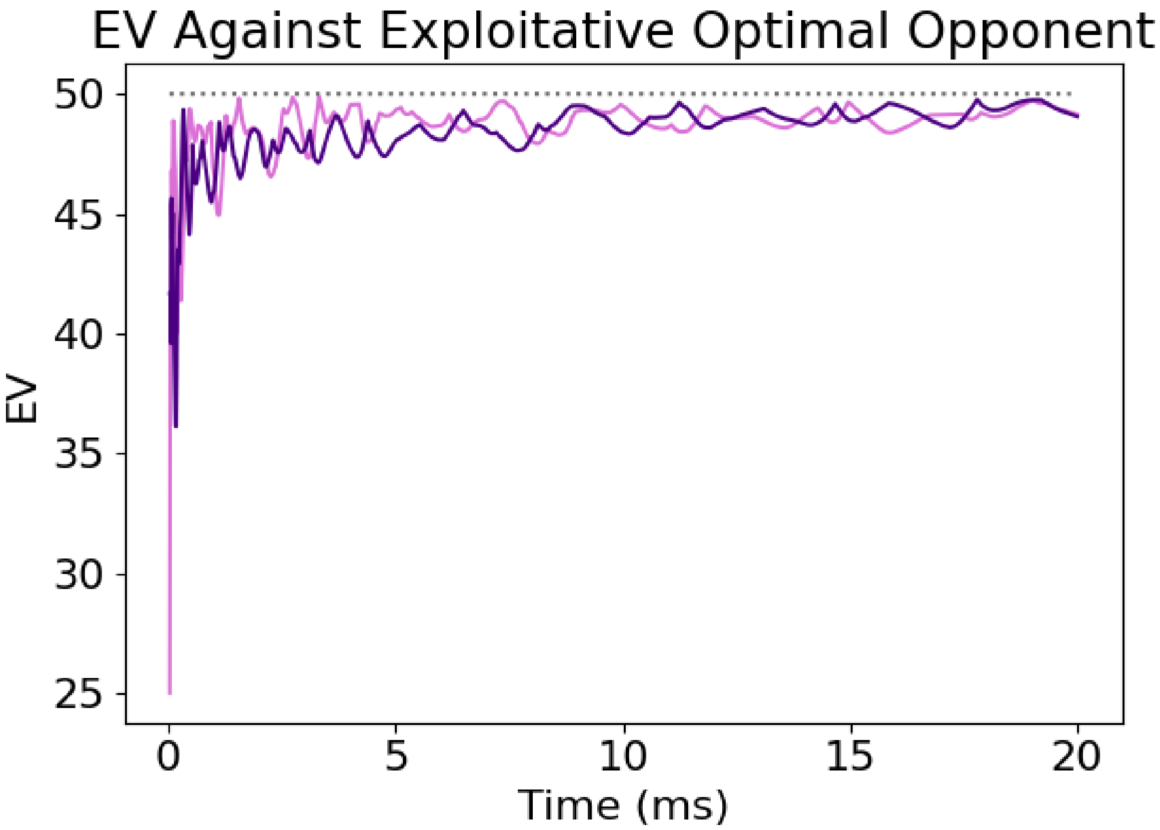
Scissor Paper Rock+ – Original



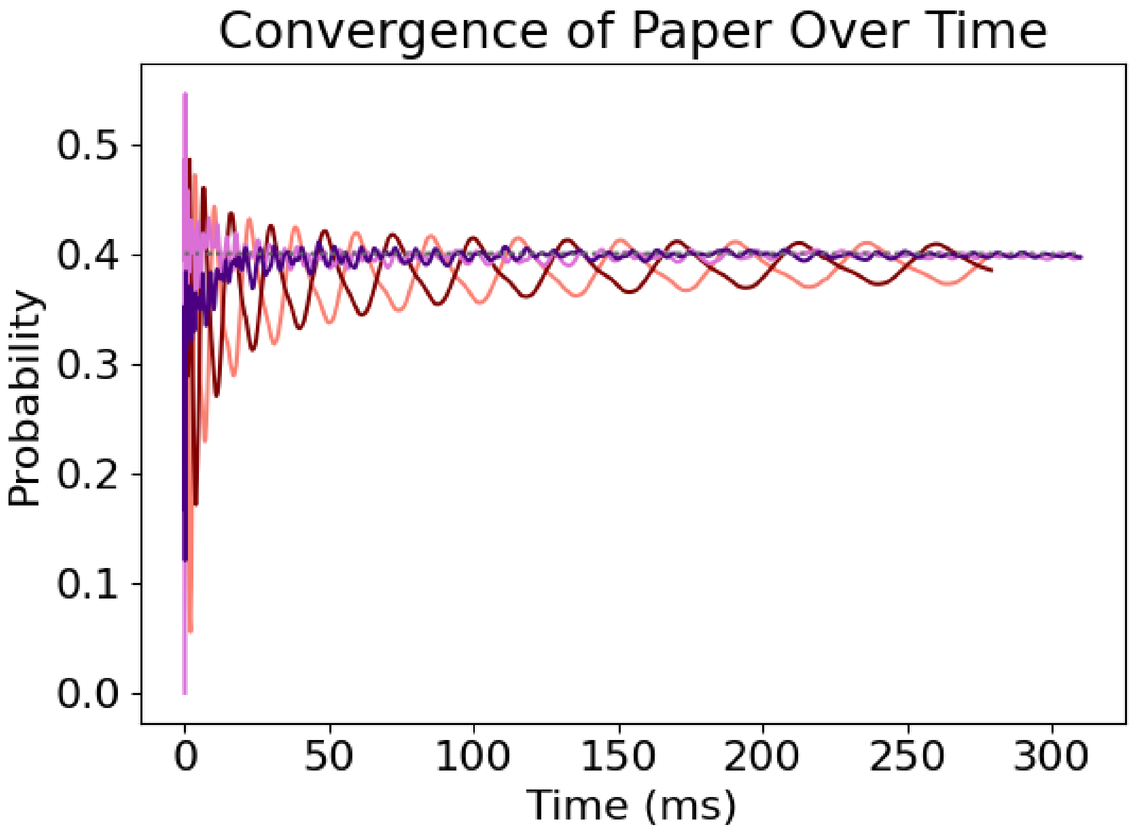


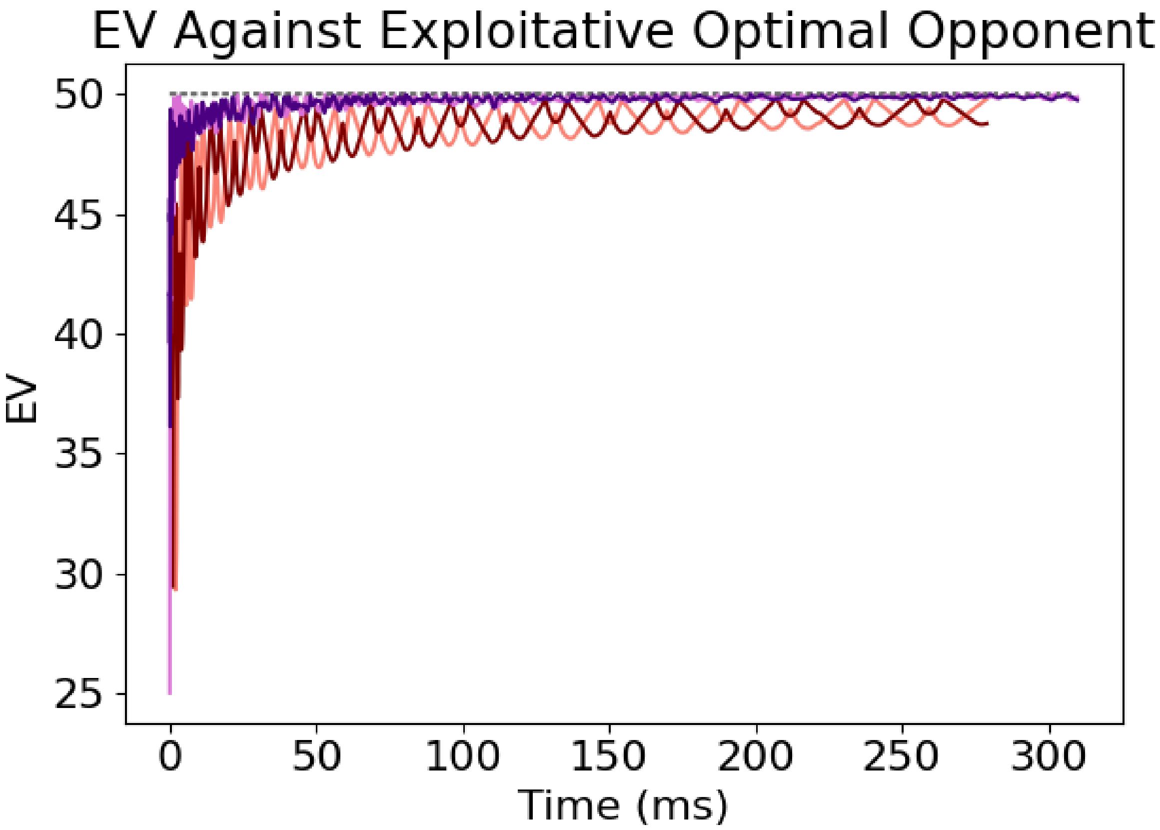
Scissor Paper Rock+ – Optimised





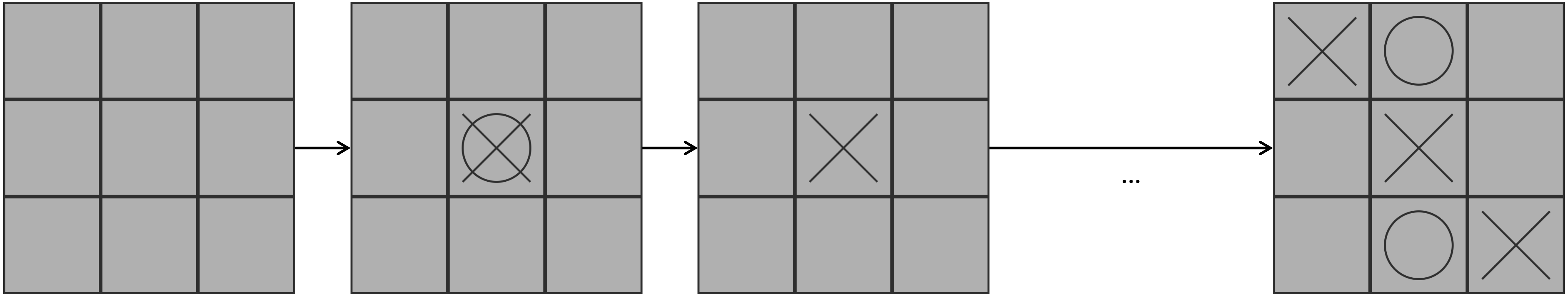
Scissor Paper Rock+ – Comparison





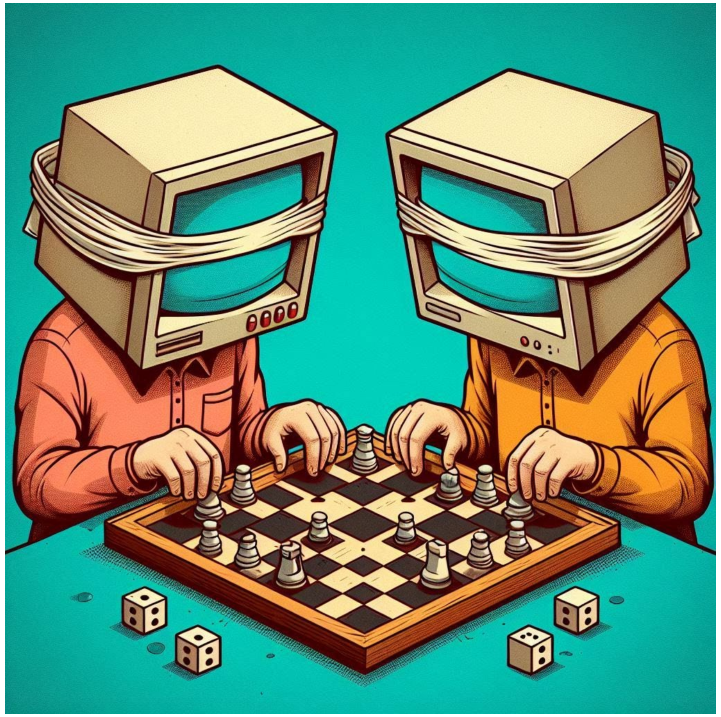
Blind Tic-Tac-Toe - Overview

- Players take turns simultaneously with no board visibility
- Players are informed if their moves are successful
- Game is won the same way as regular tic-tac-toe with three of a players “marking” in a row



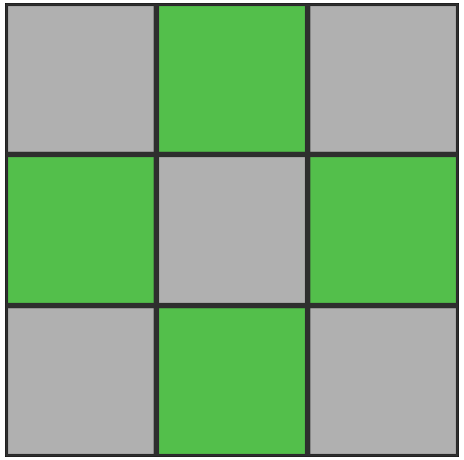
Blind Tic-Tac-Toe - Difficulties

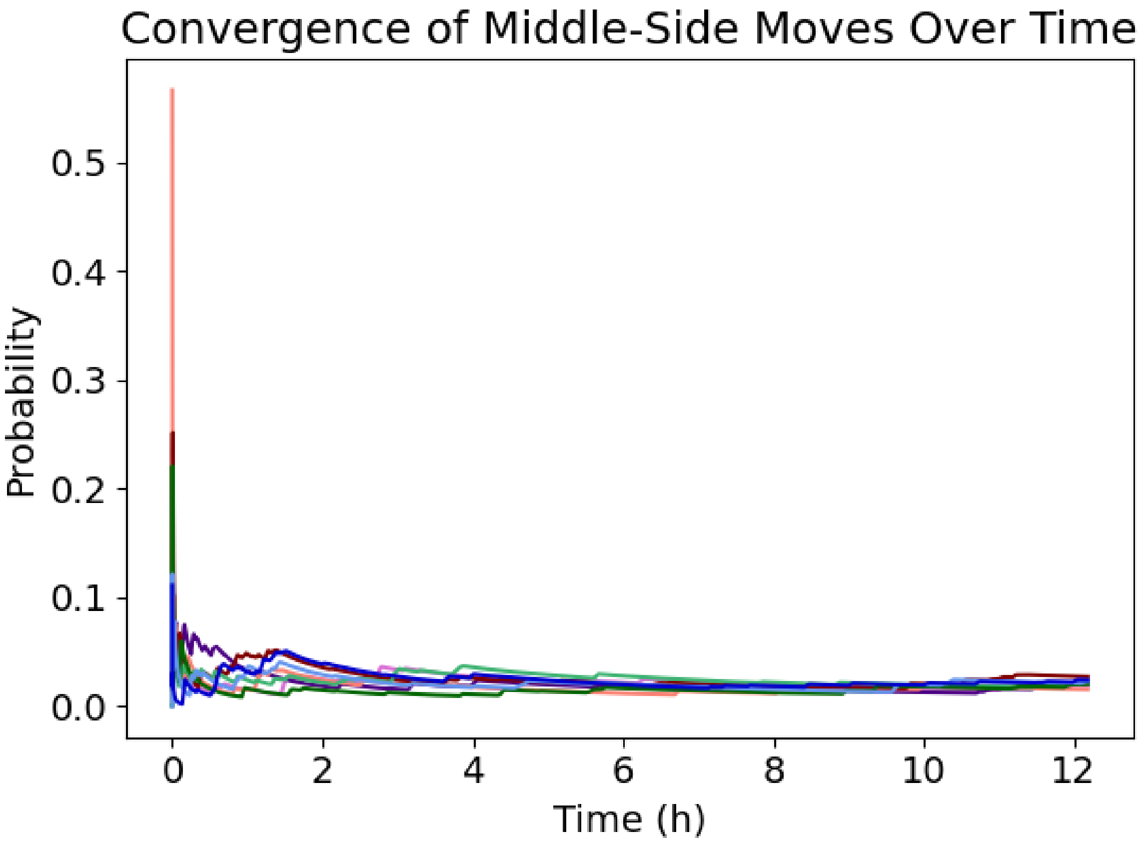
- Previous framework is unable to meaningfully search
- No visibility over actions
  - Large information sets that’re difficult to sample and search
- Large search space
  - A similar variant has approximately 10^{10} histories and $5.6 * 10^6$ information sets (Lanctot, 2013)
  - Difficult to create abstractions in general game playing



Blind Tic-Tac-Toe – Middle-Side Moves

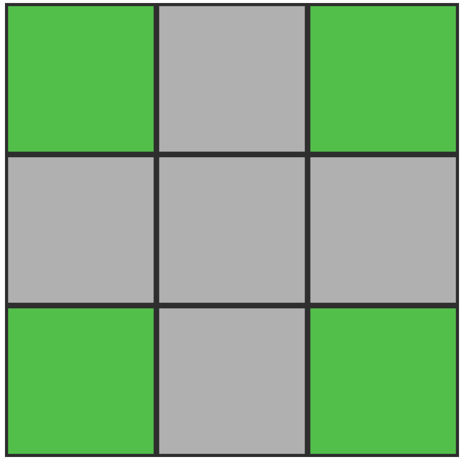
- Middle-side moves converge very quickly to essentially 0 probability of being played
- Terminates at between 1.5% and 2.7%

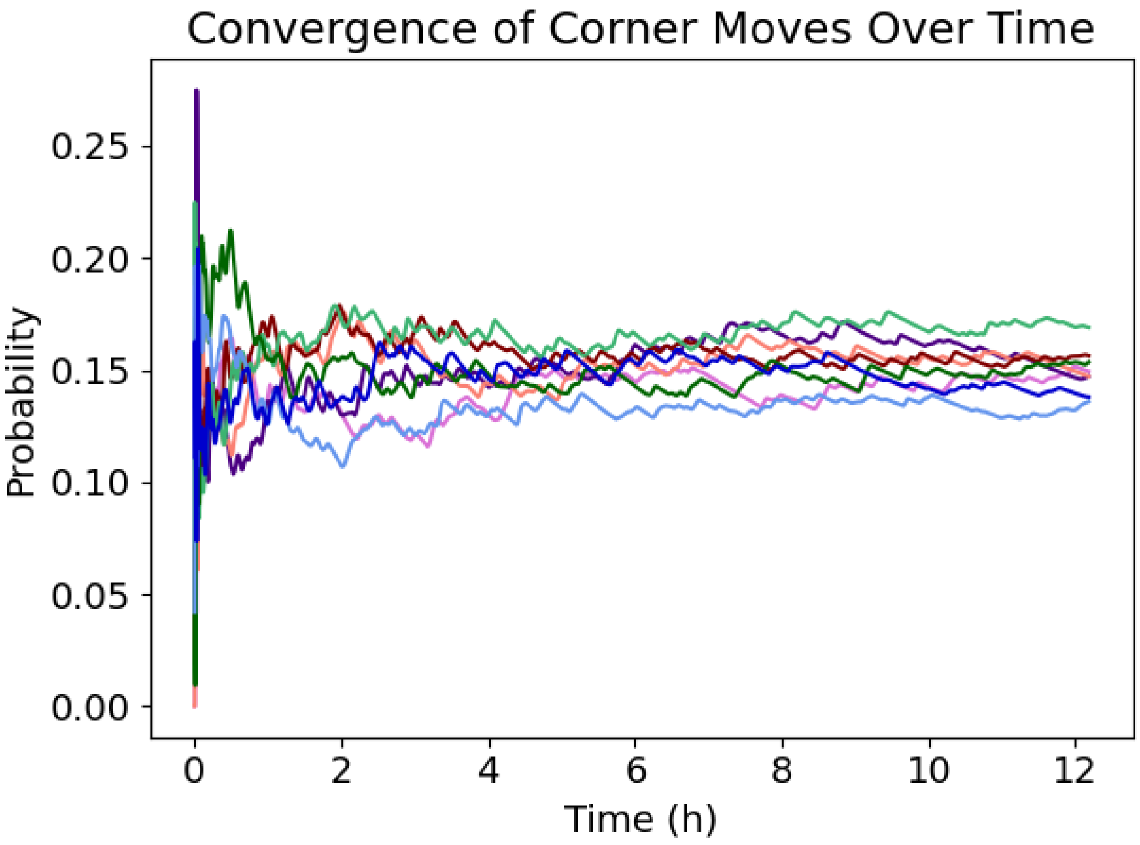




Blind Tic-Tac-Toe – Corner Moves

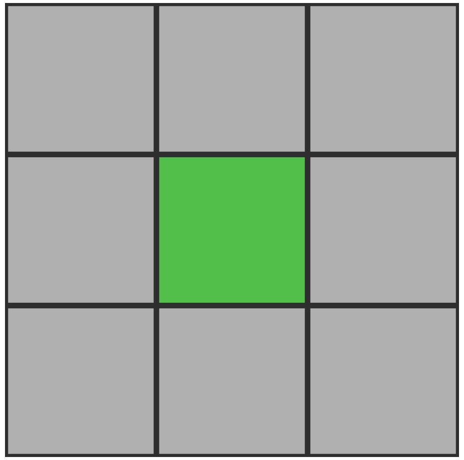
- A large variance between each of the corners that doesn't particularly converge
- Terminates at between 13.6% and 16.9%

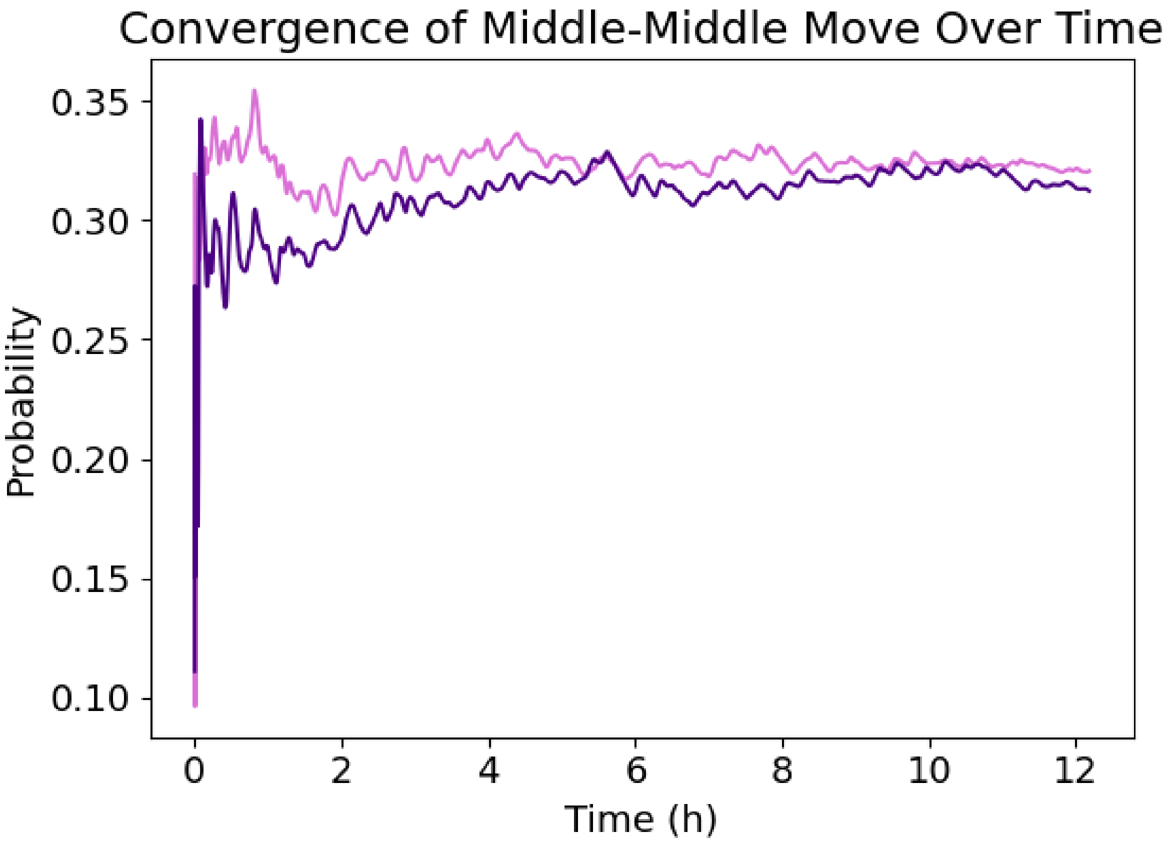




Blind Tic-Tac-Toe – Middle-Middle Move

- Initial converges quite quickly before plateauing
- Settles at 31.2% and 32.1%





Outline

- Background
- Methodology
- Results
- Conclusion
  - Summary
  - Future Work

Summary

- Written script to do some precomputation and reformat propositional networks into a more convenient format
- Reimplemented entirety of original framework in C++
- Changed CFR variant to instead use MCCFR with external sampling
- Parallelised key areas such as the state sampler and training loop
- Added testing and documentation to help future students

Future Work

- Adaptation of ReBeL into GGP
  - Assume actions are public knowledge and faithfully translate ReBeL into GGP
- Implementation of CFR-D
  - Current framework isn’t theoretically sound
  - States searched using CFR are biased towards the searching agent’s observations
- Further optimisation
  - Some games remain infeasible e.g., “blind” games with little observability over opponent actions
  - Depth limit, hyperparameters of neural network and number of iterations of CFR are merely estimations

References

https://www.youtube.com/watch?v=cn8Sld4xQjg&t

Genesereth, M., Love, N., & Pell, B. (2005). General Game Playing: Overview of the AAAI Competition. AI Magazine, 26(2), 62. https://doi.org/10.1609/aimag.v26i2.1813

Thielscher, M. (2011, July). The general game playing description language is universal. In IJCAI Proceedings -International Joint Conference on Artificial Intelligence (Vol. 22, No. 1, p. 1107).

Goldwaser, A., & Thielscher, M. (2020, April). Deep reinforcement learning for general game playing. In Proceedings of the AAAI conference on artificial intelligence (Vol. 34, No. 02, pp. 1701-1708).

Brown, N., Bakhtin, A., Lerer, A., & Gong, Q. (2020). Combining deep reinforcement learning and search for imperfect-information games. Advances in Neural Information Processing Systems, 33, 17057-17069.

Genesereth, M., & Thielscher, M. (2022). General game playing. Springer Nature.

Partridge, Z., & Thielscher, M. (2022, November). Hidden Information General Game Playing with Deep Learning and Search. In Pacific Rim International Conference on Artificial Intelligence (pp. 161-172). Cham: Springer Nature Switzerland.

Heinrich, J., & Silver, D. (2016). Deep reinforcement learning from self-play in imperfect-information games. arXiv preprint arXiv:1603.01121.

Schiffel, S., & Thielscher, M. (2014). Representing and reasoning about the rules of general games with imperfect information. Journal of Artificial Intelligence Research, 49, 171-206.

https://www.youtube.com/watch?v=mCldyXOYNok&t

West, J., & Thielscher, M. (2023). Opponent Modelling in General Game Playing with Hidden Information and Deep Learning Thesis A Report

Lanctot, M. (2013). Monte Carlo sampling and regret minimization for equilibrium computation and decision-making in large extensive form games. University of Alberta (Canada).

Cox, E., Schkufza, E., Madsen, R., & Genesereth, M. (2009, July). Factoring general games using propositional automata. In Proceedings of the IJCAI Workshop on General Intelligence in Game-Playing Agents (GIGA) (pp. 13-20).