

## Context

### Artificial intelligence for games

- Several succes :
  - backgammon, othello, draughts, chess, go, ...
- perfect information :
  - games in turn, without chance or hidden information

### State of the Art

- AlphaZero :
  - reinforcement learning
    - using tree search
    - without knowledge
  - mastering Go, Chess, and Shogi
  - limit : 29 millions of matches / 10 000 billions state evaluations

### Questions

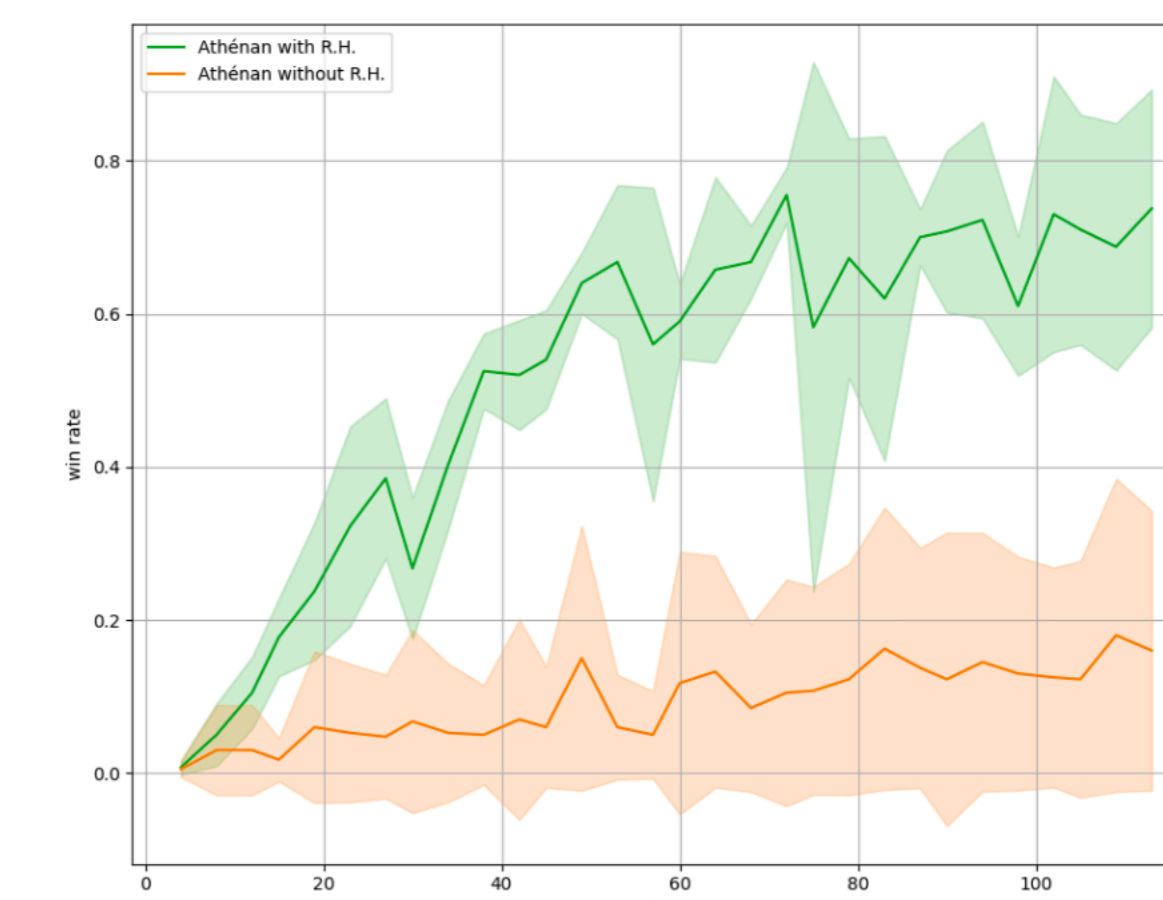
- minimax + reinforcement < MCTS + reinforcement ?
- can we do better than AlphaZero ?
  - reasonable requirements :
    - number of matches / number of state evaluations
    - used computer
  - better level of play

### Problematic

- Athénan :
  - a new minimax framework
  - learning without knowledge as AlphaZero
- **Goal**  $\Rightarrow$  Comparison of Athénan with AlphaZero

## 113 Day Training

Win rate against a strong hex program (Mohex 2.0) :

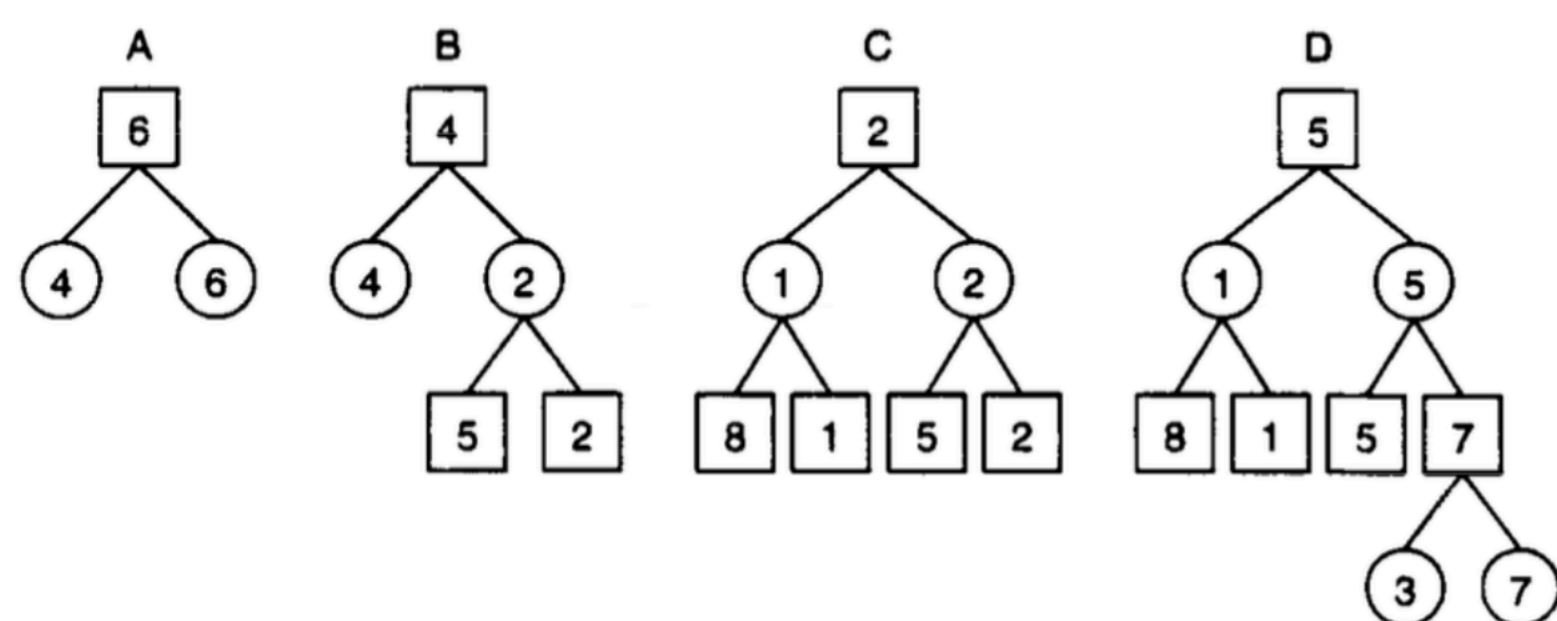


Polygames : 0% of win at any time of the training...

## Game Tree Searches

### Unbounded Best First Minimax

- iteratively extend the best sequence of actions
- value of a state : minimax of the partial game tree



### Descent Minimax algorithm

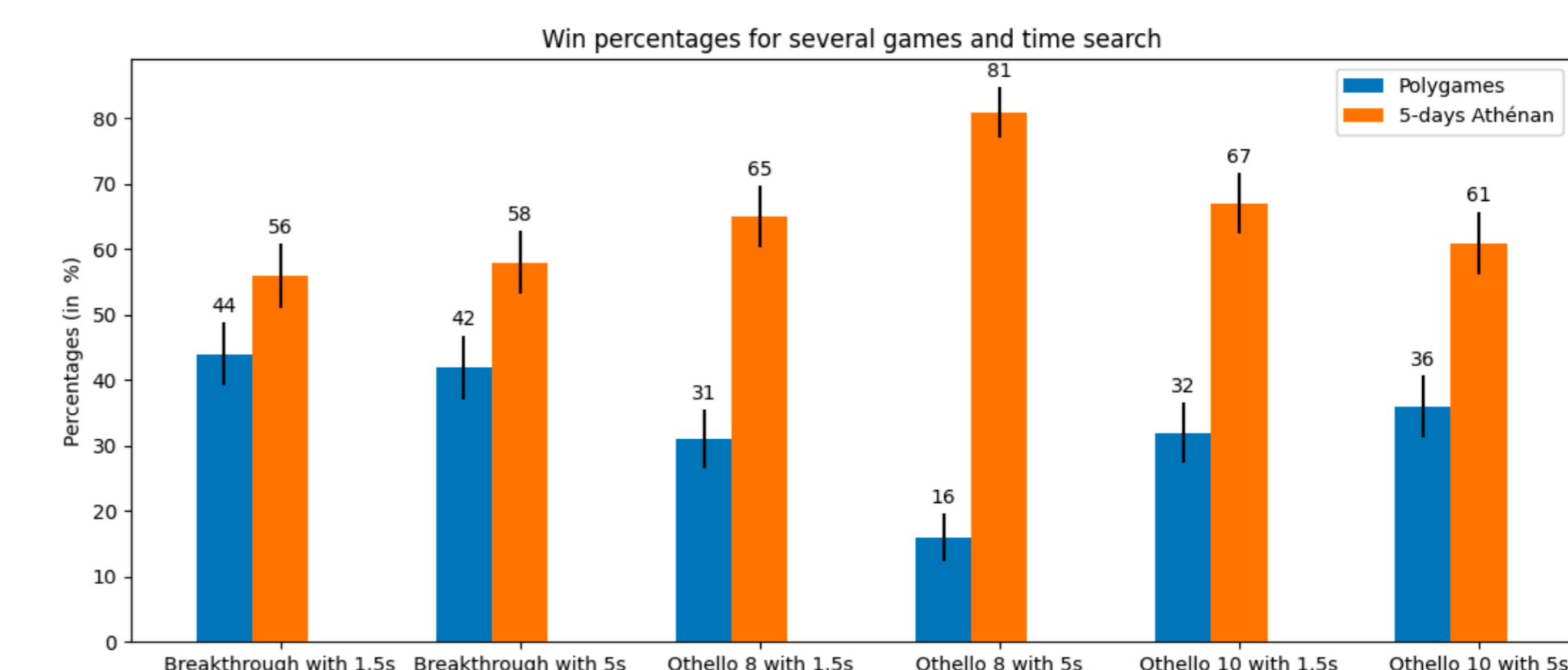
- variant of Unbounded Minimax
  - one iteration : until the **end of the game**
    - instead of : until reaching the **horizon**
  - $\Rightarrow$  endgame deterministic simulations
  - always choosing the best action

### Monte Carlo Tree Search

- iteratively extend the sequence of actions maximizing
  - state value : victory statistics
  - exploration term / confidence bound

## Tournament Networks

Polygames Tournament Networks : 100 GPUs and 1 learning week  
 Athénan : 1 GPU and 5 learning days



## Frameworks for Zero Learning

### Data for learning

- terminal learning :
  - states : states of the match
  - target value : gain of the endgame
- tree learning :
  - states : states of searches during the game
  - target value : minimax

### Athénan

- no use of policy
- search :
  - during training : Descent Minimax
  - after training : Unbounded Minimax
- learning target :
  - tree learning
- use of reinforcement heuristic
  - terminal evaluation more expressive than  $-1/0/1$
  - ex : score ; winning fast and losing slowly

### AlphaZero

- use of a policy : probability of playing an action
  - calculated by the neural network
- MCTS : exploration term : PUCT
  - with respect to the policy
- learning target :
  - terminal learning
  - policy : proportional to  $v^{\tau}$ 
    - $v$  : number of selections
    - $\tau$  : parameter

Open-source re-implementation of AlphaZero : **Polygames**

## Competitions

### Computer Olympiad

- 2020 : 5 gold medals
  - Othello 10, Breakthrough, Clobber, Amazons, Surakarta
- 2021 : 11 gold medals
  - news : Brazilian & Canadian Draughts, Hex 11&13&19, Othello 8, Havannah 8&10
- 2022 : 5 gold medals
  - losses : Othello 8, Brazilian Draughts
  - news : Santorini, Ataxx
- 2023 : 16 gold medals
  - news : Arimaa, Xiangqi, Lines of Action,
- 2024 : 11 gold medals
  - losses : Santorini
  - news : Shobu, Othello 16

## Conclusion

### Athénan Results compared to Polygames (AlphaZero)

- at least 300 times more learning data efficient
- at least more than 3 times more wins
- learning speed at least 30 times faster
- at least on some games
  - Athénan + 1 GPU > Polygames + 100 GPUs
- Computer Olympiad
  - 48 gold medals in five years
  - triple the record ever achieved in a single year

### References

- 2020, arxiv :
  - Learning to Play Two-Player Perfect-Information Games without Knowledge
    - Quentin Cohen-Solal
- 2023, AMAAS :
  - Minimax Strikes Back
    - Quentin Cohen-Solal and Tristan Cazenave

Thanks to GREYC, CRIL, IDRIS, LAMSADE for their computing servers !

Thank you for your attention !

## 15 Day Training for 8 Games

Average results :

	Connect6	Havannah 10	Havannah 8	Outer-Open-Gomoku	Hex 13	Surakarta	Othello	Breakthrough
Learned states	55	64	111	115	359	442	529	693

### In average

- Athénan 296 times **more** learned states :  
 $\Rightarrow$  use of Tree Learning

Evolutions of performances (win - loss) :

