Computer Go Research - The Challenges Ahead

Martin Müller
University of Alberta

CIG 2015

Computer Go Research

- Brief history
- Recent progress

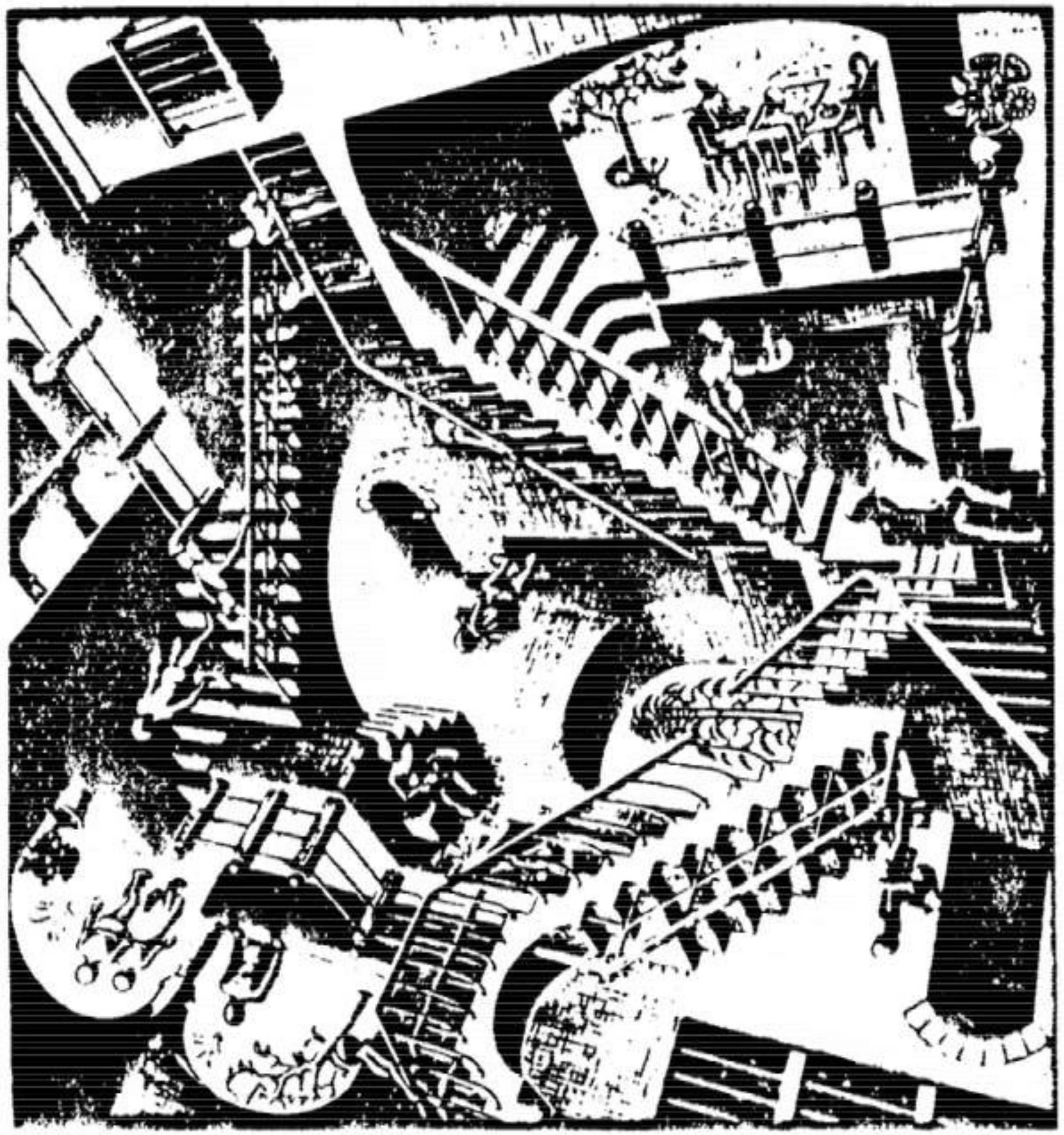
- Challenges
- Outlook

Early History

- Early work in the 1960s and 1970s, e.g. Reitman and Wilcox
- Tournaments start in mid 1980s when personal computers become available
- Big sponsor in Taiwan: Ing foundation

Computer Go

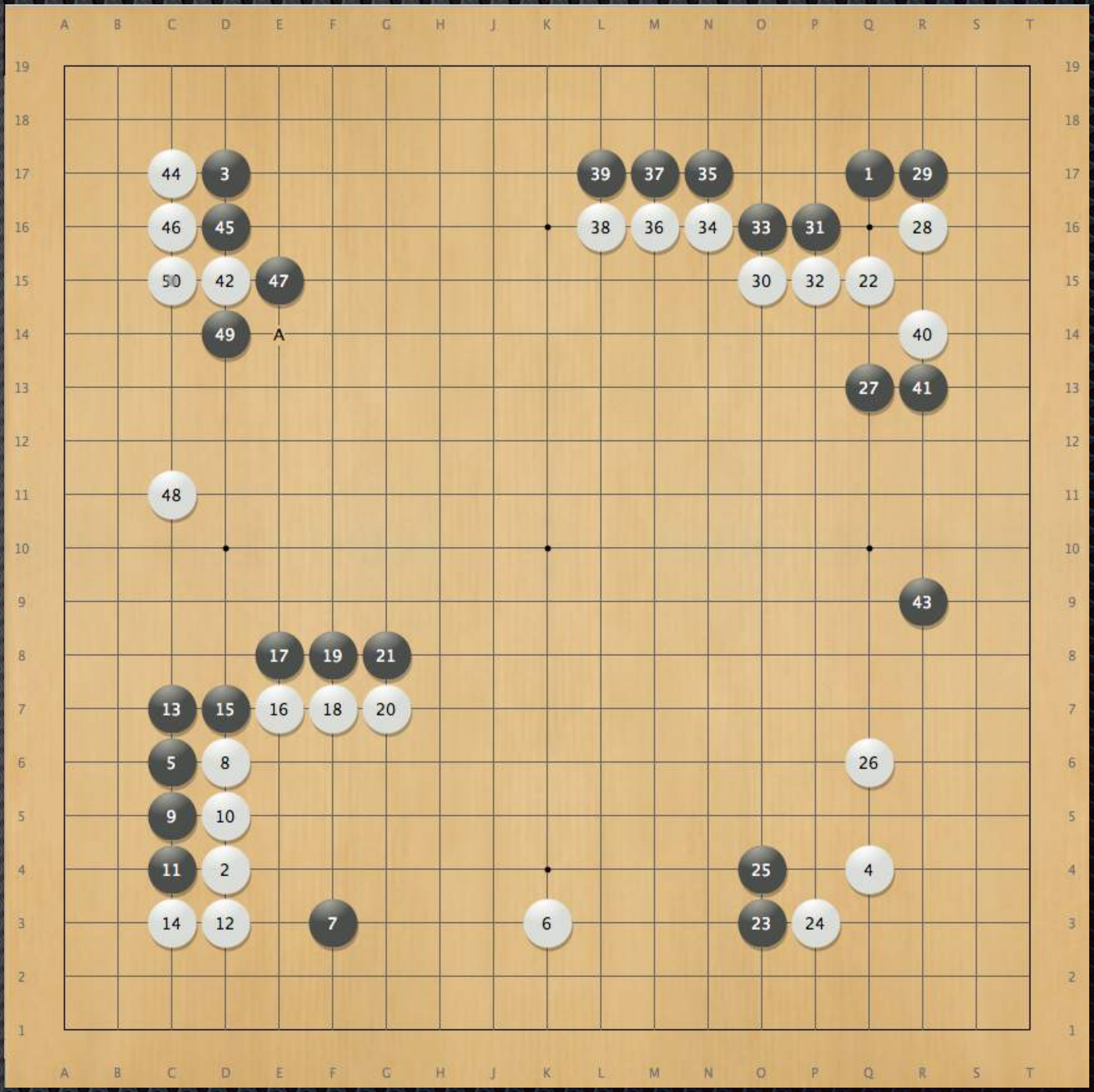
Winter 1986-87 No. 1



An international bulletin devoted to the generation and exchange of ideas about Computer Go

Early Go Programs

- Used patterns, often hand-made
- Limited tactical search, ladders
- Little or no global-level search
- Lost with 17 handicap stones against humans



ICGC 1988, Taiwan, Dragon (W) vs Explorer (B)

Progress vs Humans?

Ing Cup winning programs -
wins against humans (1985 - 2000):

17 stones - Goliath wins 1991

15 stones - Handtalk wins 1995

13 stones - Handtalk wins 1995

11 stones - Handtalk wins 1997

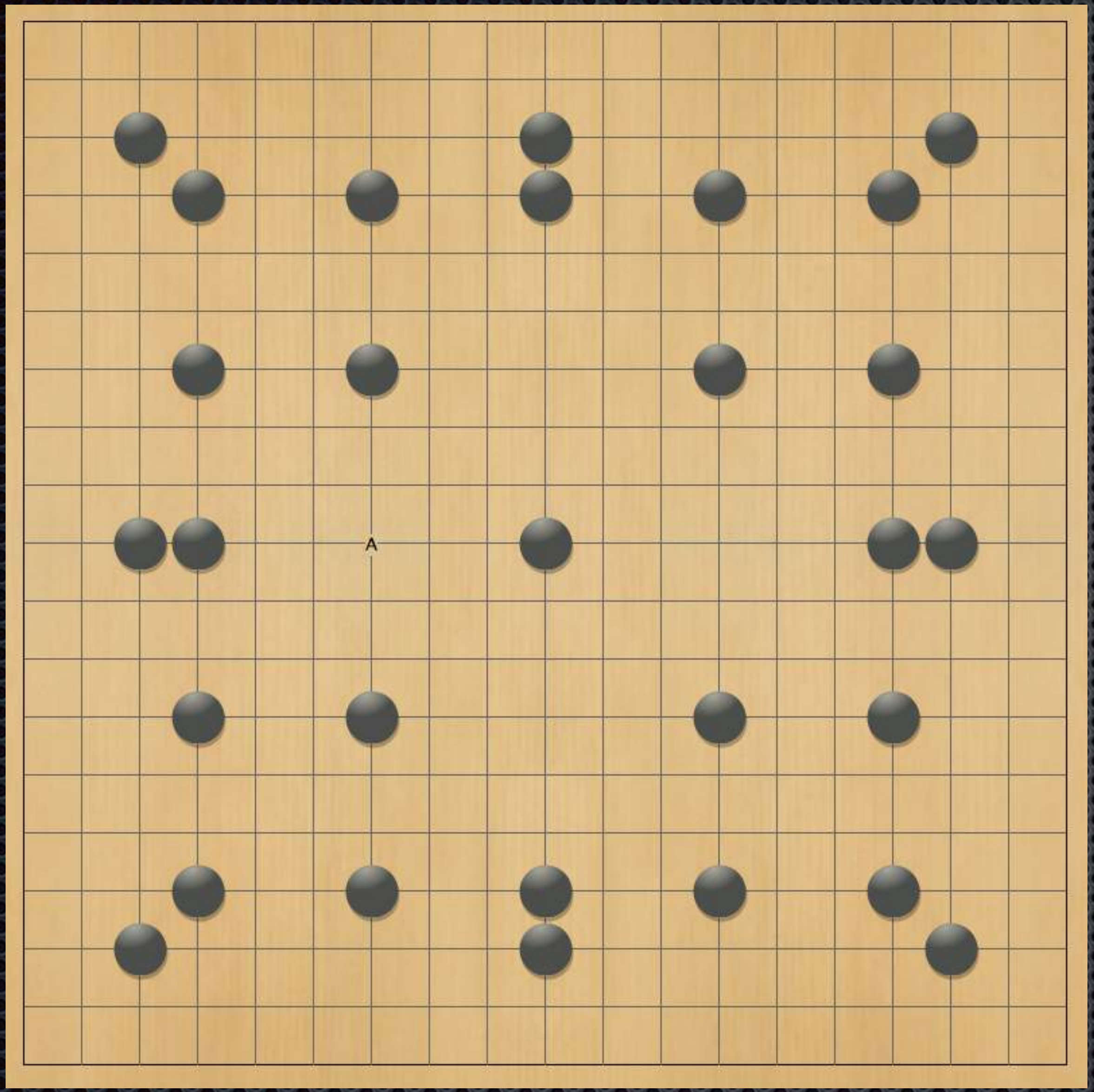
But: Two test games in 1998

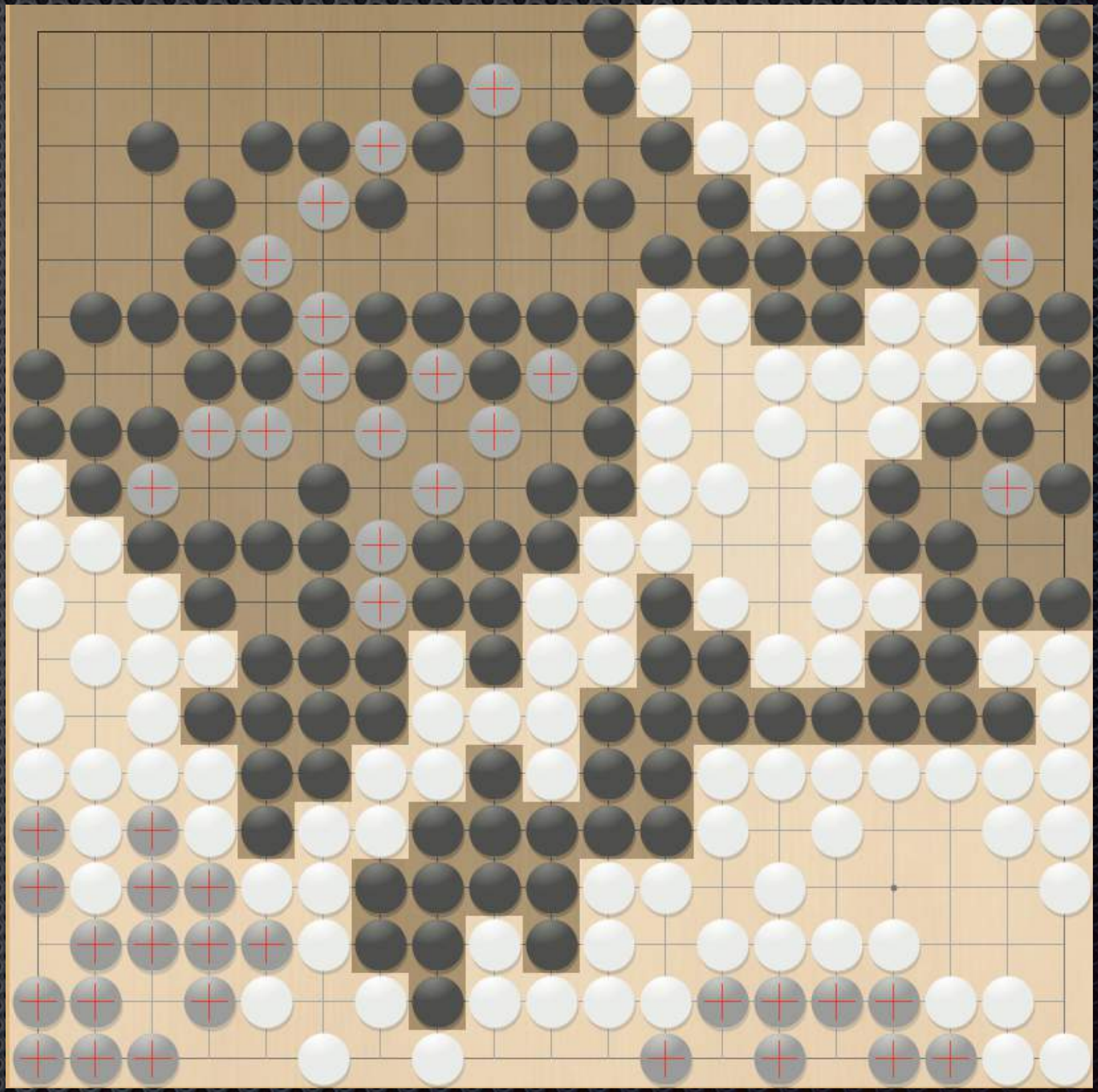
17 stones - Handtalk loses to Gailly 5 kyu
29 stones - Many Faces of Go loses

Mark Boon (Goliath)

Chen Zhixing (Handtalk)
Credits: M. Reiss

Martin Müller vs Many Faces of Go 29 handicap (1998)

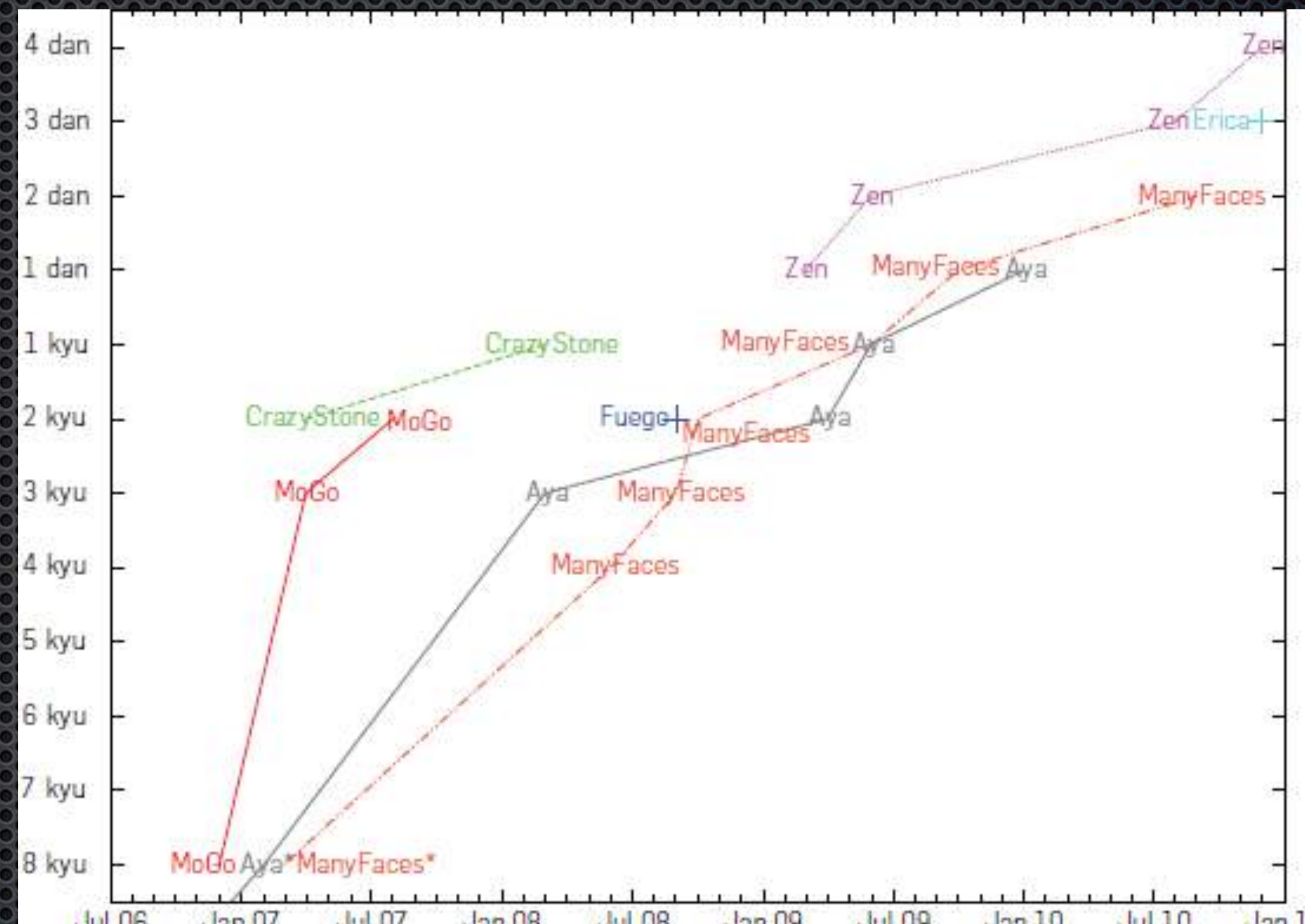




279 moves, White wins by 6 points

Monte Carlo Tree Search

- About 10 years ago, French researchers revive the idea of random simulations for Go
- Kocsis and Szepesvari develop UCT
- Soon Crazy Stone and MoGo become strong and start the MCTS revolution



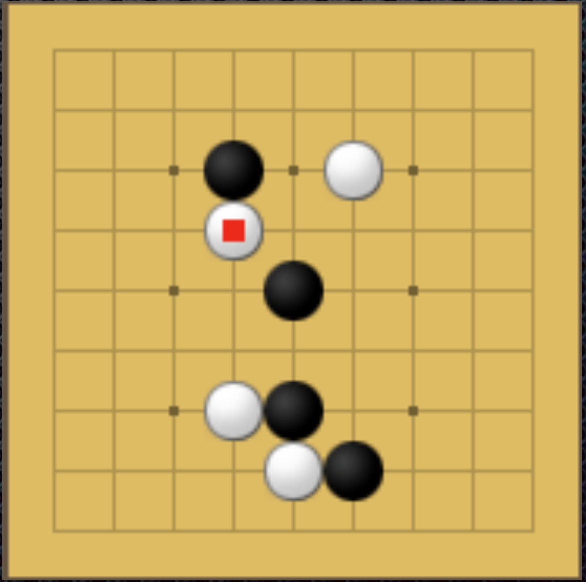
source: acm.org

Some MCTS Go Milestone Wins

- 2008 Mogo vs Kim 8p, 8 handicap
- 2008 Crazy Stone vs Aoba 4p, 7 stones
- 2009 MoGo vs Chou 9p, 7 stones
- 2009 Fuego vs Chou 9p, 9x9, even

Olivier Teytaud (Mogo)

Remi Coulom (Crazy Stone) and Ishida 9p



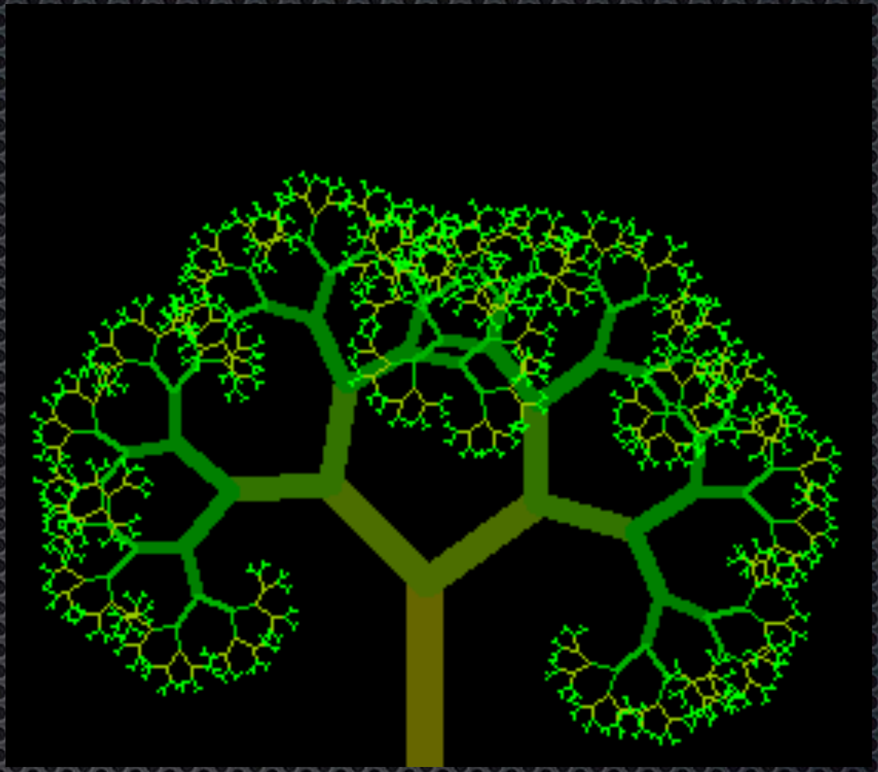
Credits: http://www.computer-go.info, gogameguru.com

Current Strength

- Programs ~~often~~ sometimes win with 4 handicap against pro
- Lose with 3
- Yesterday, Chou 9p and Yu 1p beat Zen with 4 handicap

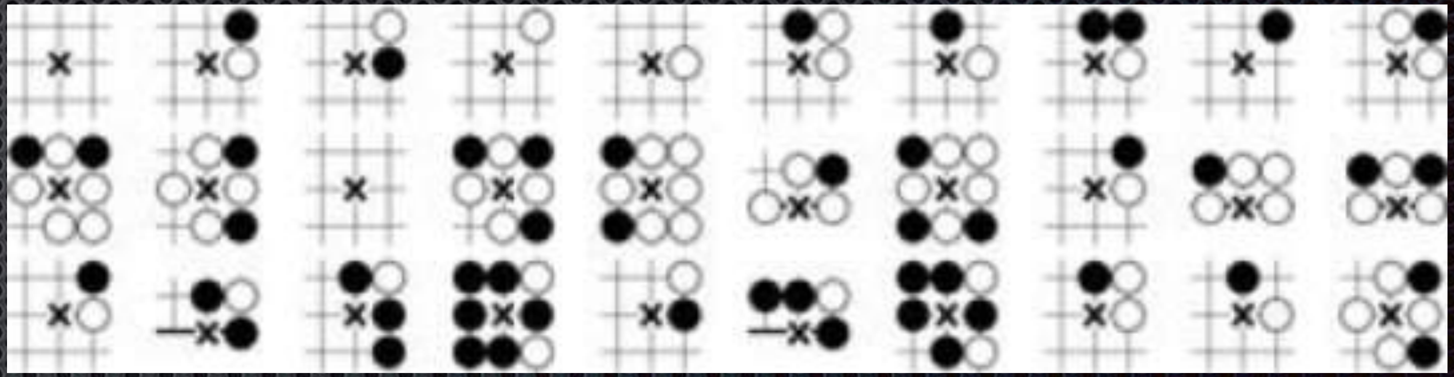
Cho Chikun vs Crazy Stone,
3 handicap, Densei-sen 2015
Credit: http://www.go-baduk-weiqi.de

State of the Art in Computer Go



Three main ingredients:

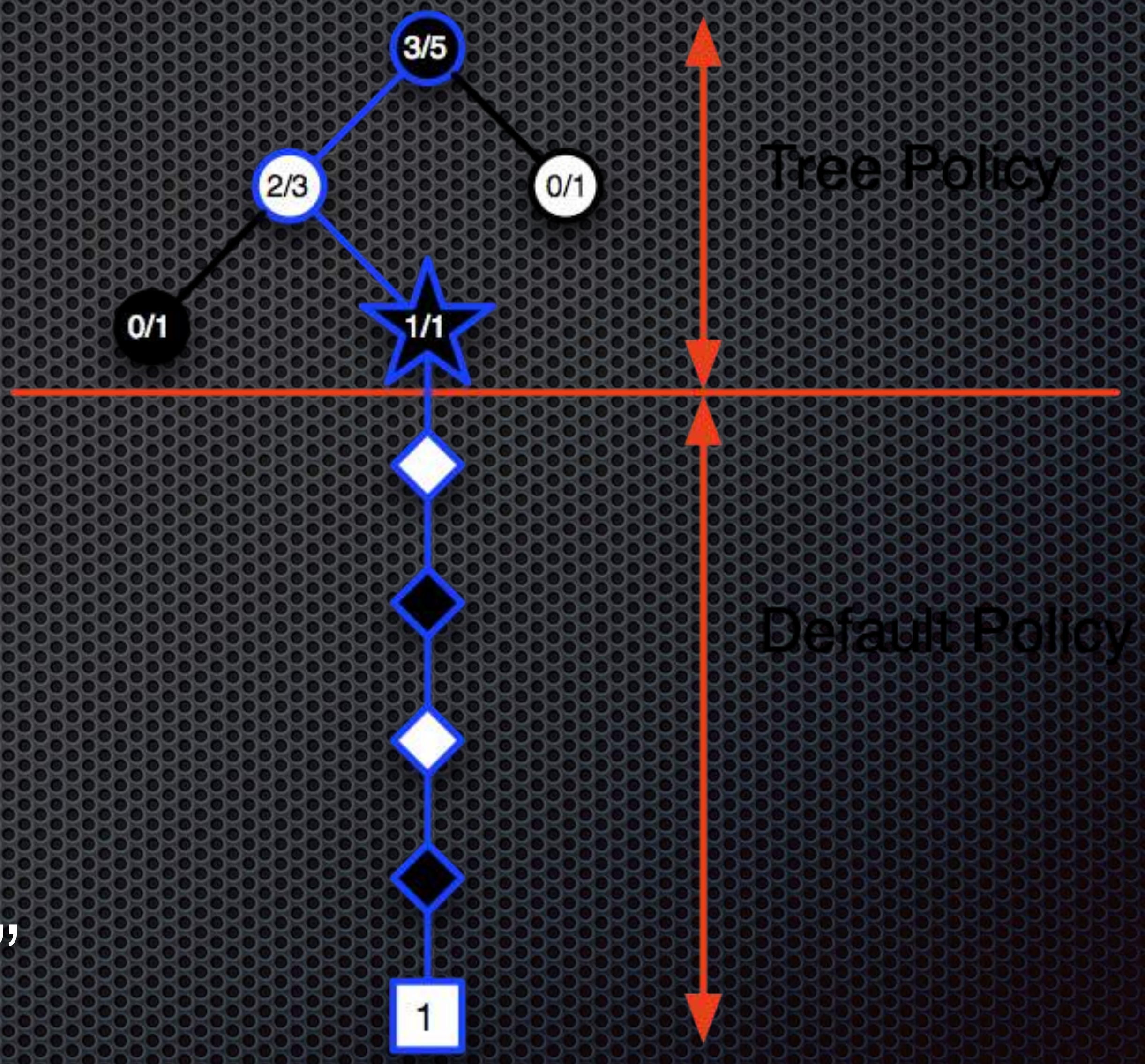
1. Tree Search
2. Simulation
3. Knowledge



Credits: visualbots.com, sciencedaily.com,

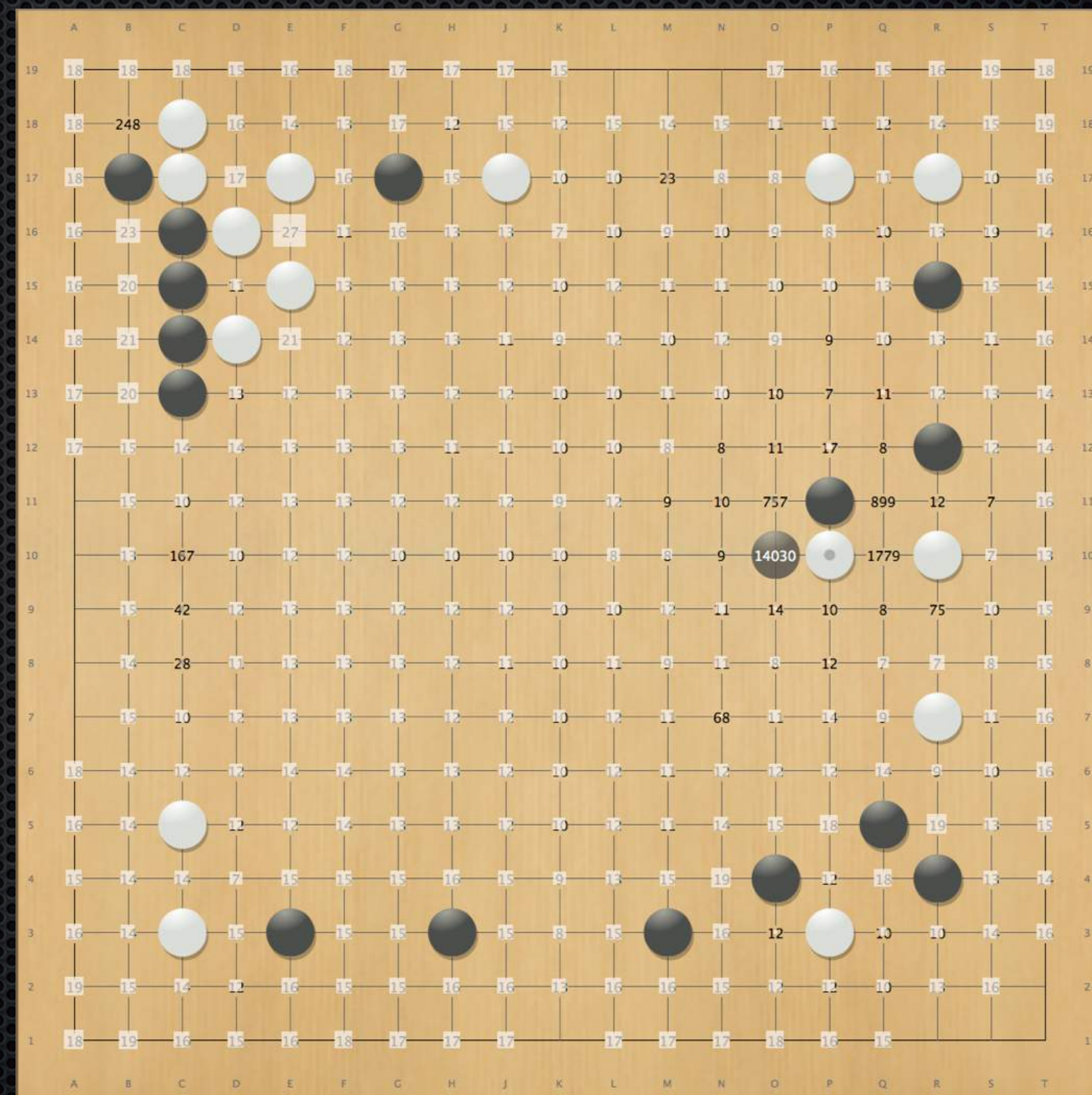
1. Tree Search

- Very selective search
- Driven by two main factors
  - Statistics on outcome of simulation
  - Prior knowledge “bias”



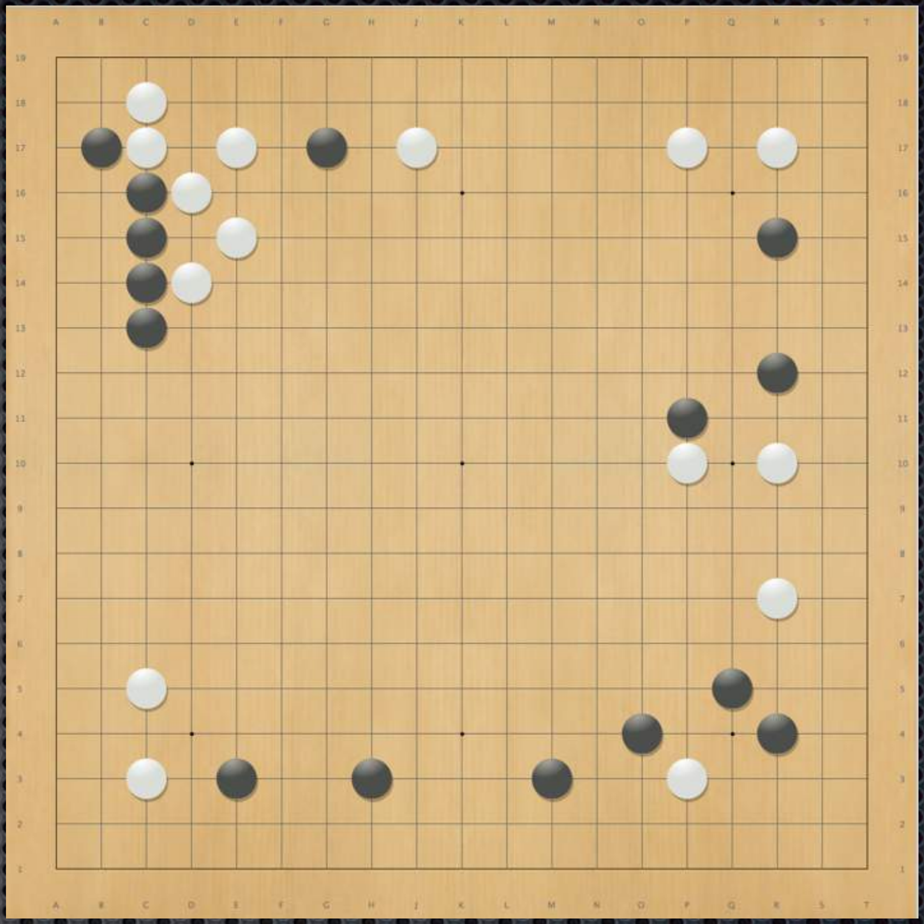
Highly Selective Search

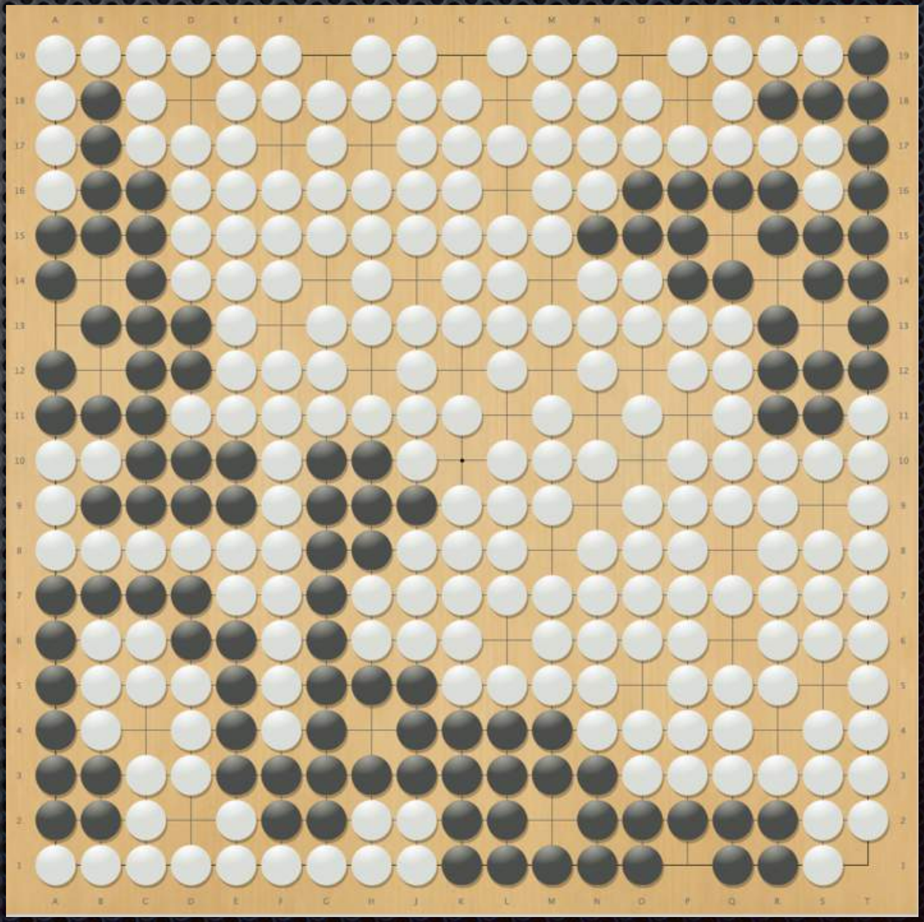
- Snapshot from Fuego
- 18000 simulations, of which more than 14000 on one move
- Most moves are not expanded due to knowledge bias
- Deep search: average 13.5 ply, maximum 31 ply



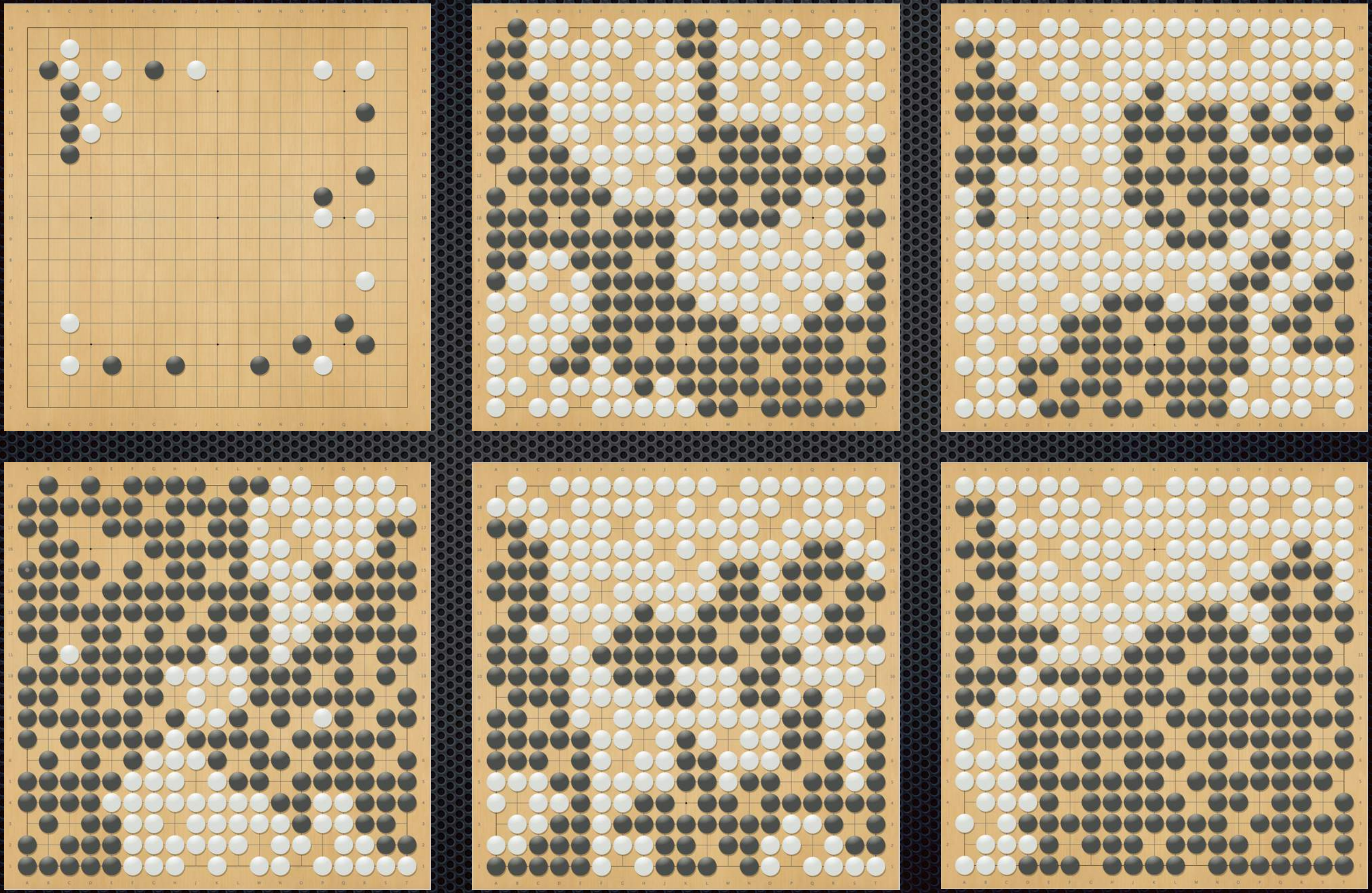
2. Simulation

- Play complete game
- Start at a leaf node in the tree
- Fast randomized *policy* generates moves
- Store only win/loss result of games in tree



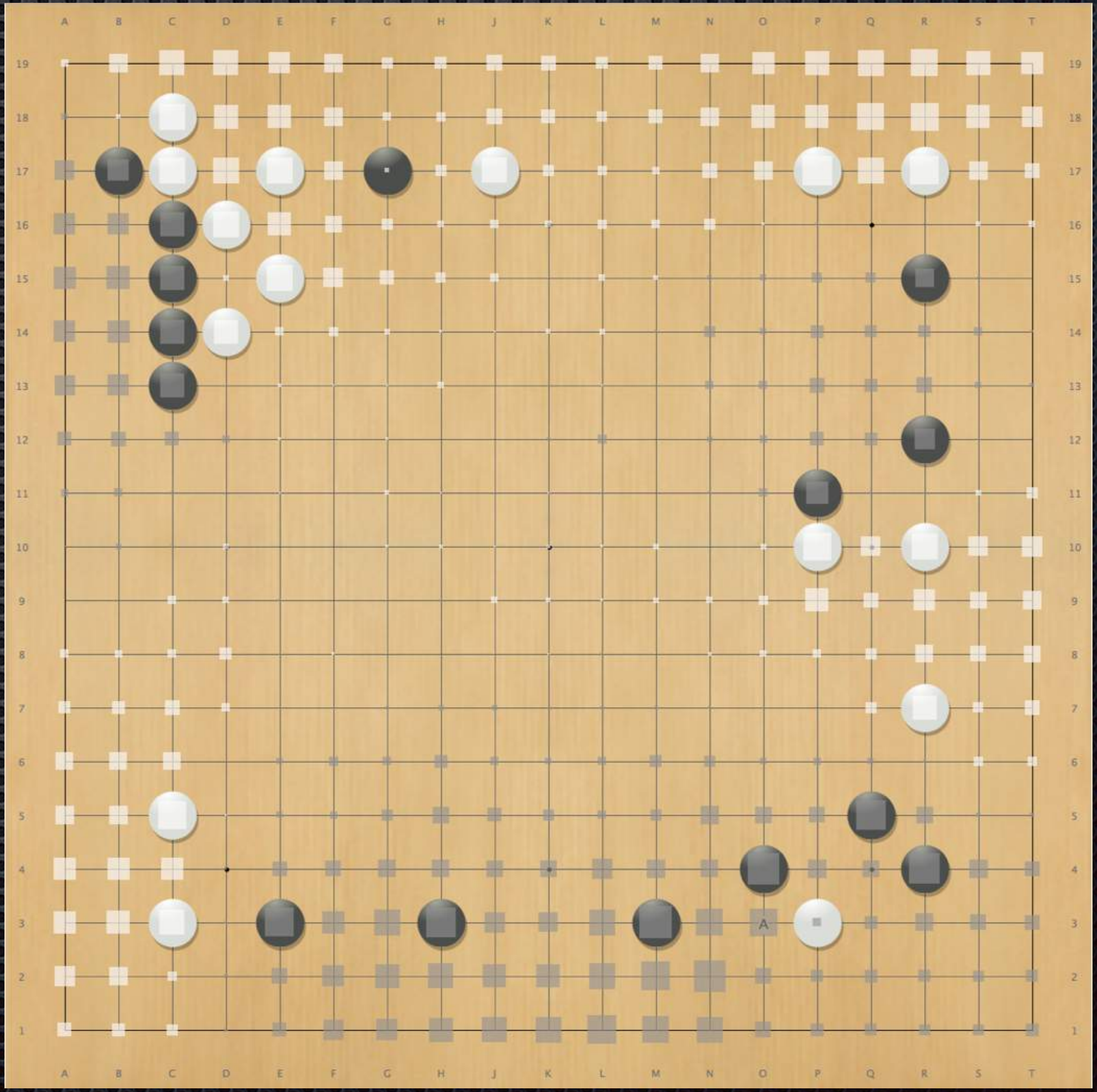


Large Variance: Five More Simulations From Same Starting Position



Average Outcome

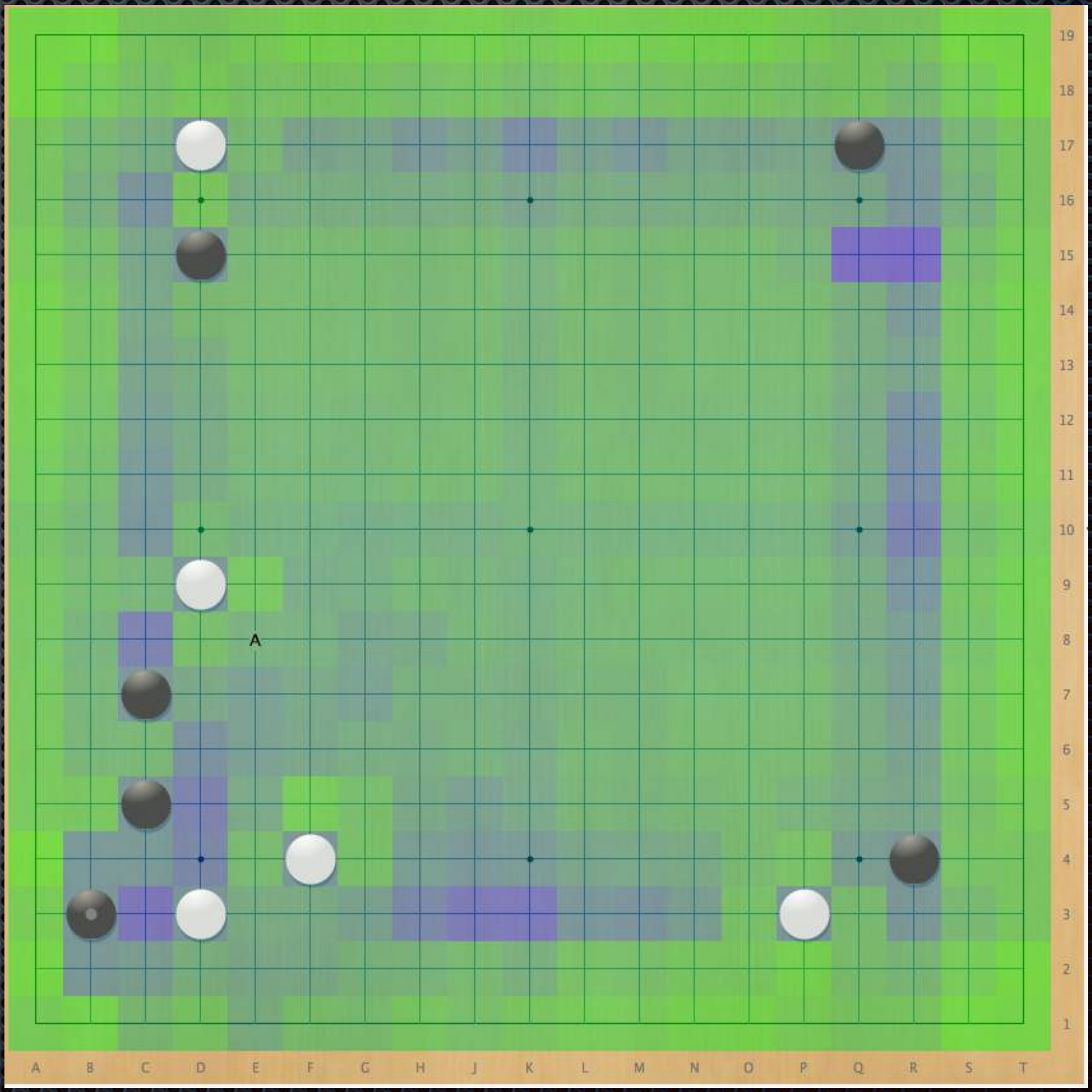
- Single simulation outcomes look almost random
- Average of 100 simulations looks good!
- Statistics over "almost random" outcomes are useful!



3. Go Knowledge for MCTS

1. Simple Features
2. Patterns
3. Deep Convolutional Neural Networks (DCNN)

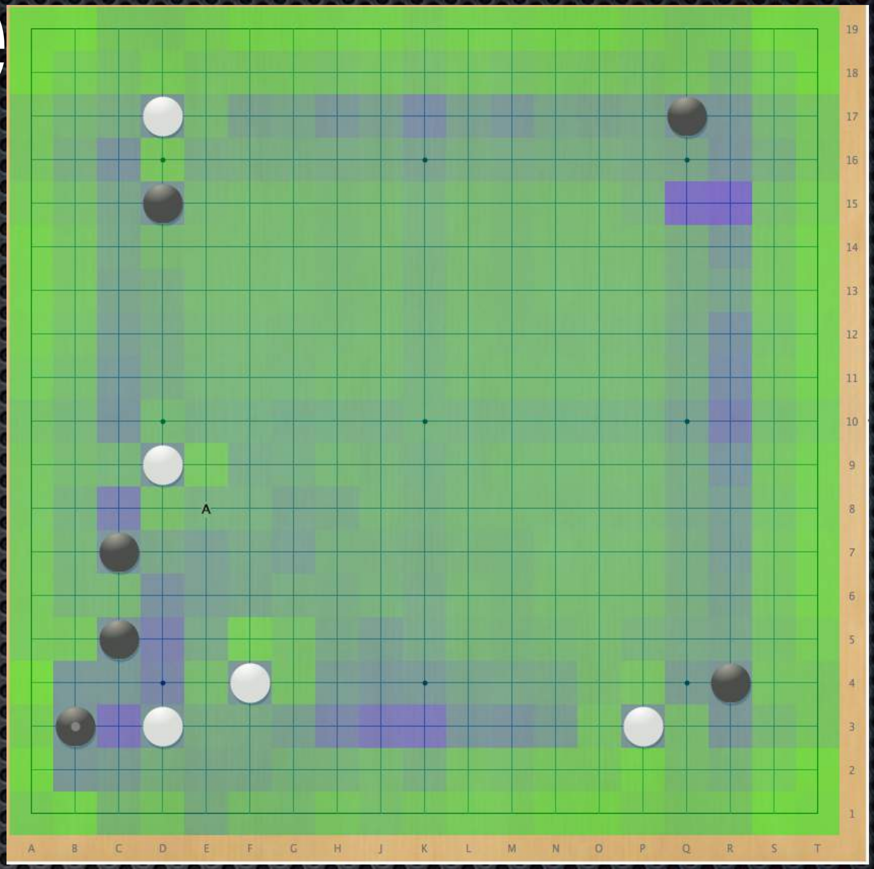
- First question: why use knowledge?



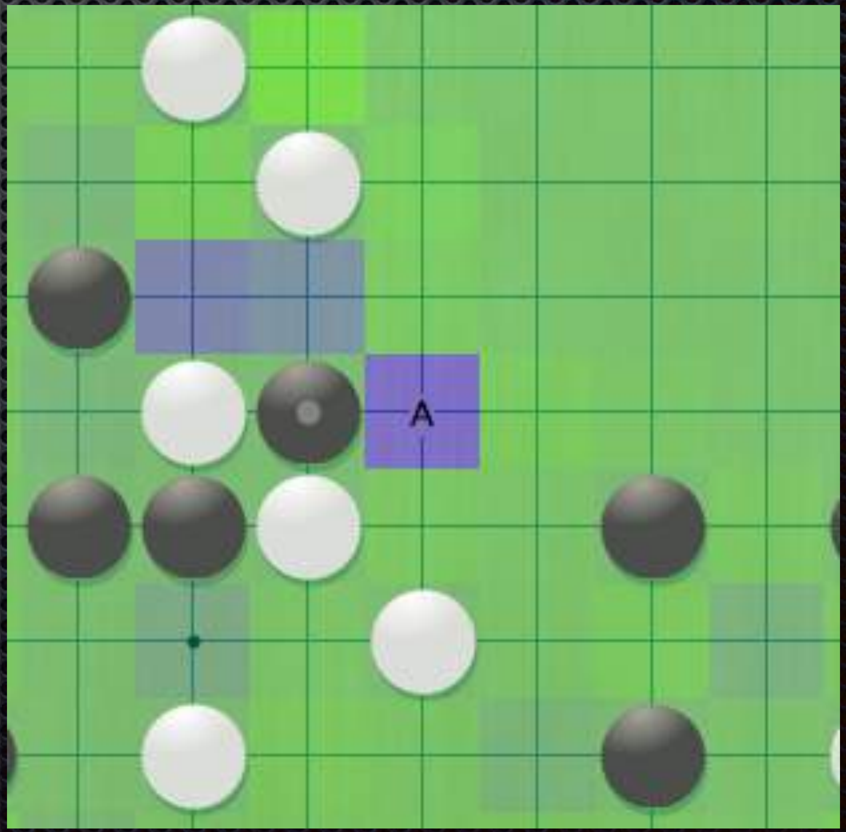
Using Knowledge

- Knowledge and simulations have different strengths
  - Use for moves that are difficult to recognize with simulation
- Use as evaluation function
- Describes which moves are expected to be good or bad
- Use as initial bias in search
- Use when no time to search

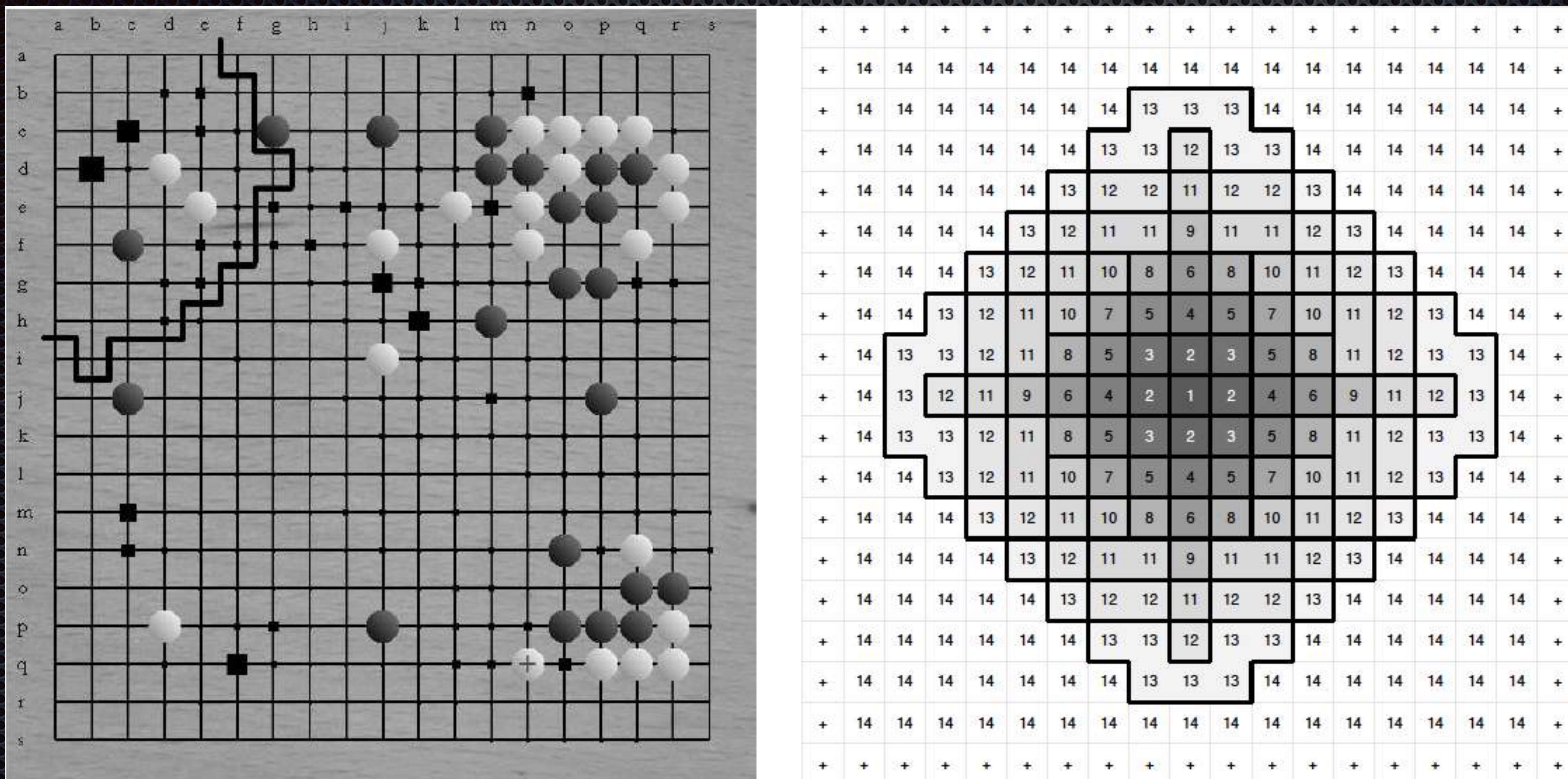
3.1 Simple Feature Knowledge



- Location - line, corner
- Distance -
  to stones of both players,
  to last move(s)
- Basic tactics -
  capture, escape,
  extend/reduce liberties



3.2 Pattern Knowledge

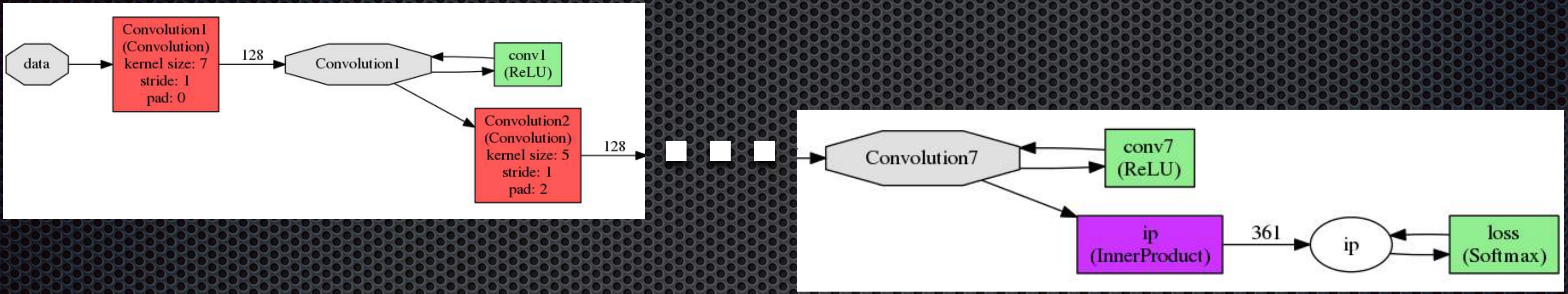


Source: Stern et al, ICML 2006

Using Patterns

- Small patterns (3x3) used in fast playouts
- Multi-scale patterns used in tree
- Weights set by supervised learning

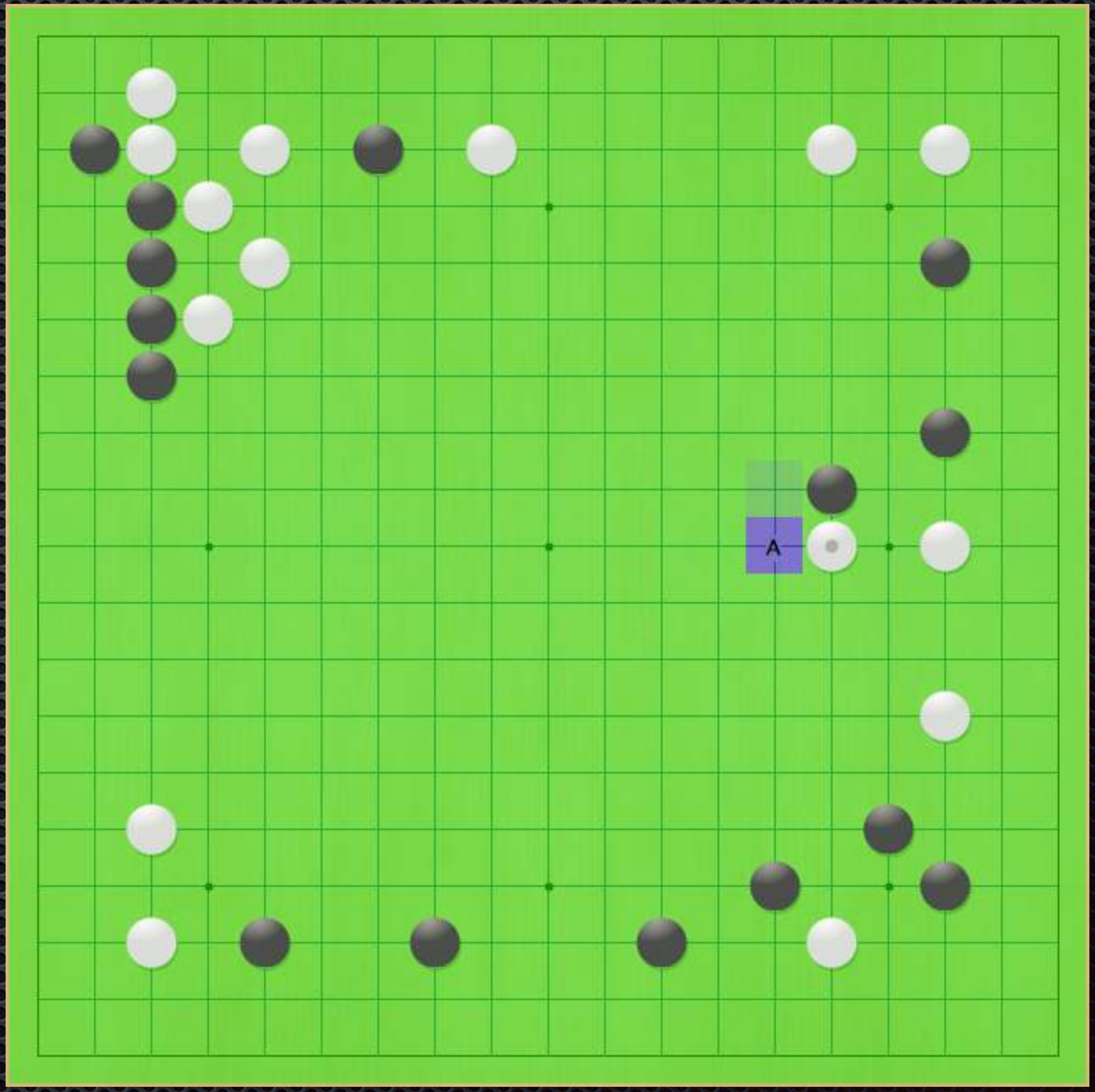
3.3 Deep Convolutional Neural Networks, DCNN



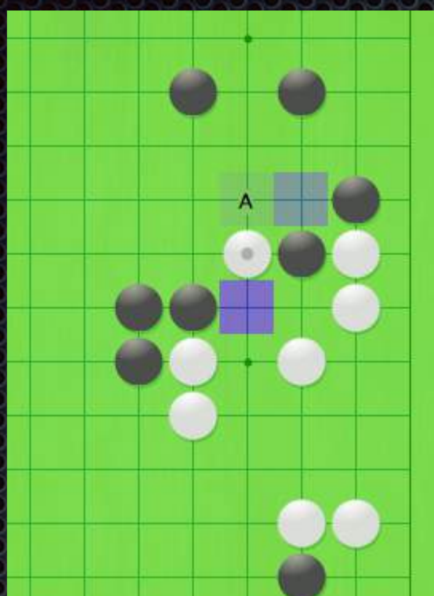
- Introduced for Go in two recent publications
  - Clark and Storkey, JMLR 2015
  - Maddison, Huang, Sutskever and Silver, ICLR 2015
- Very strong move prediction rates, 55.2% (Maddison et al)
- Slow to train and use (even with GPU)

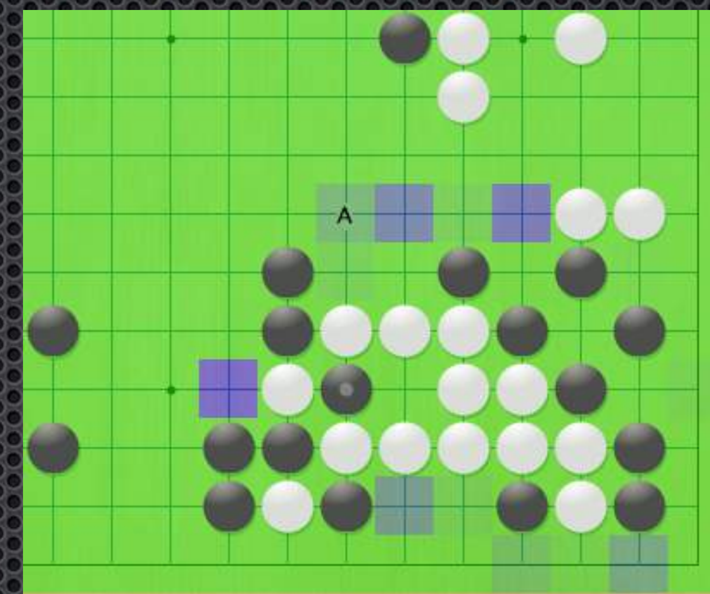
DCNN Move Prediction

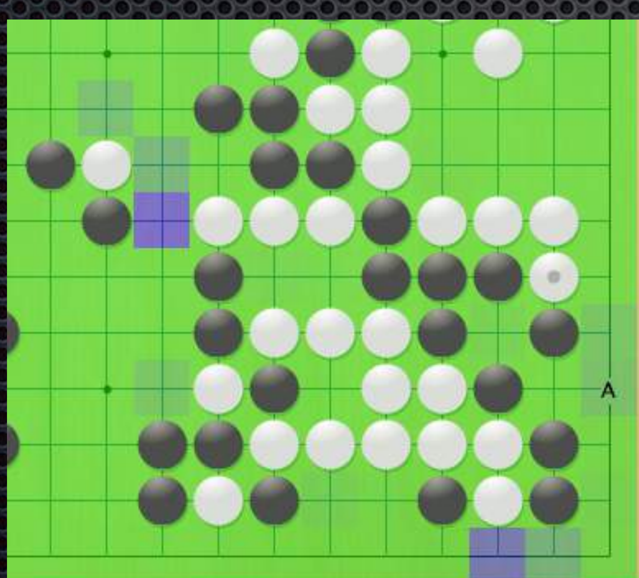
- Network provided by Storkey and Henrion
- Added to Fuego
- Often strong focus on one favorite move
- Often predicts longer sequences of moves correctly, but…

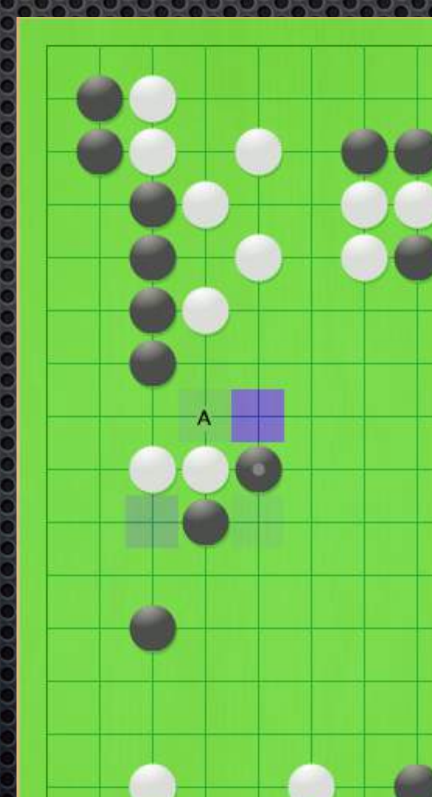


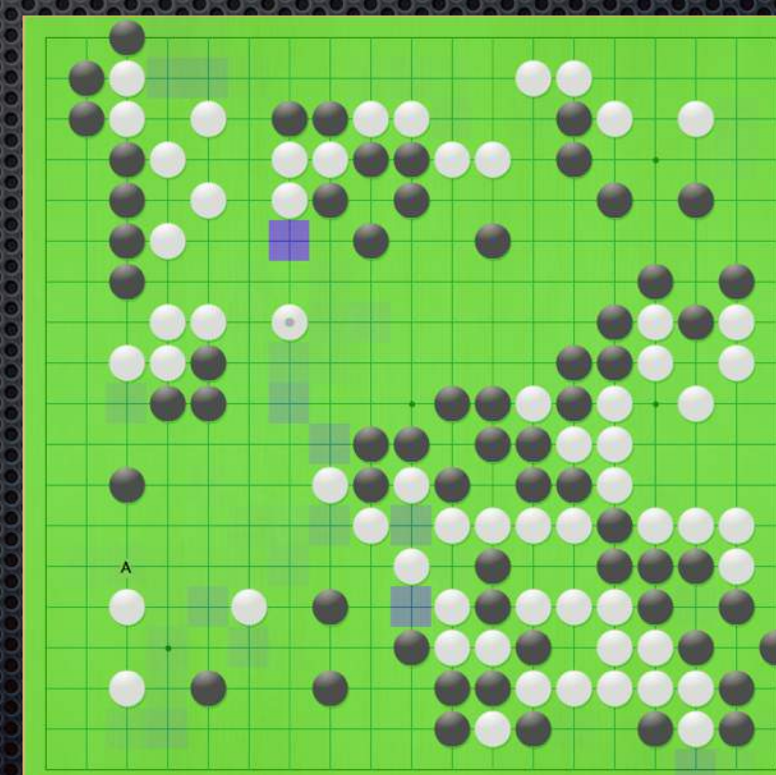
DCNN Are Not Always Right…









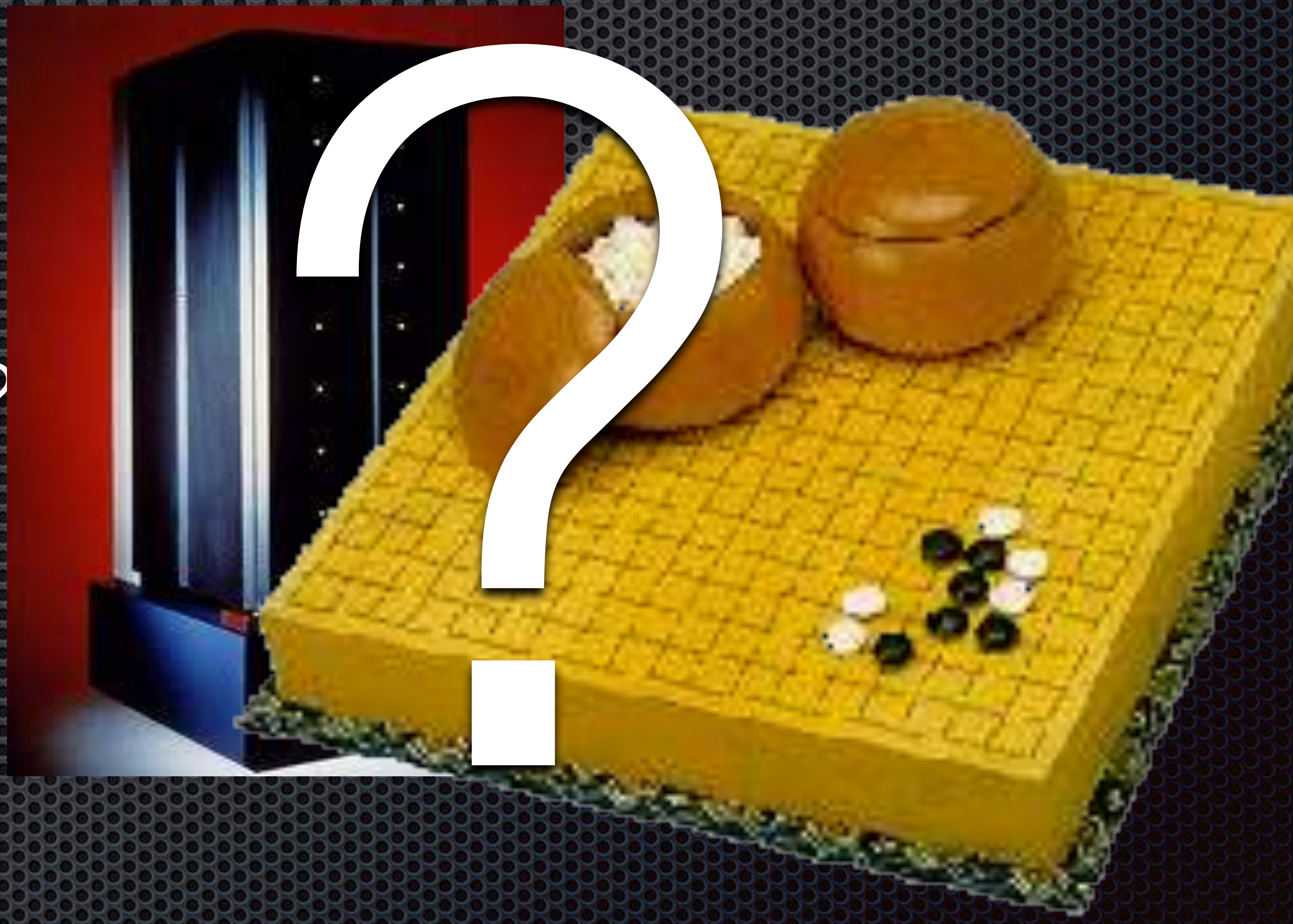


More Knowledge…

- Tactical search
- Solving Life and Death (Kishimoto and Müller 2005)
- Proving safety of territories (Niu and Müller 2004)
- Special cases such as seki (coexistence), nakade (large dead eye shapes), bent four, complex ko

Challenges for Computer Go

- How to improve?
- How to make progress?
- What should we work on?
- My personal list only, no broad consensus
- Format:
  1 slide to introduce a problem,
  1 slide to discuss



Challenge: Strengthen the Computer Go Research Community

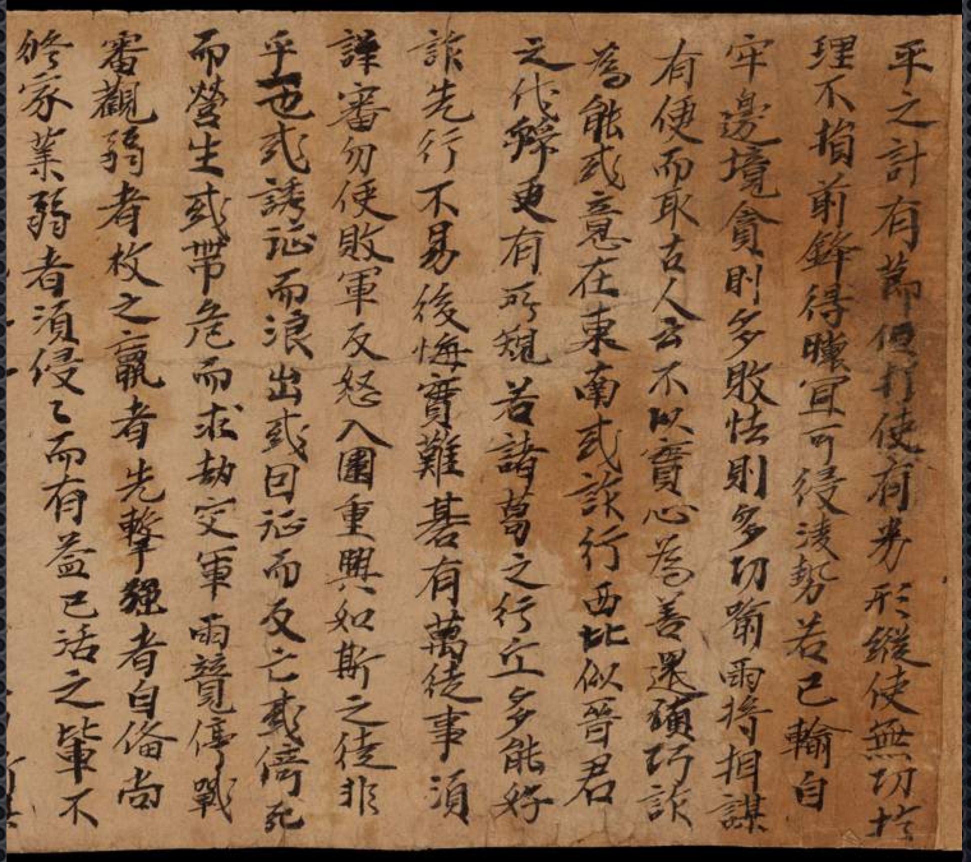
- Many program authors do not talk/publish enough
- No coordinated effort to build a top program

Research Questions

- Can we combine research results without duplicating effort?
- Can we use a common software platform?
- Can we share detailed results, including testing and negative results?

Challenge: Combine Many Types of Go Knowledge

- Many kinds of knowledge:
  - Simulation policy
  - In-tree knowledge
  - Neural Networks
  - Tactical search



Source: usgo.org

- How to make them all fit together in MCTS?

Research Questions

- Is there a "common currency" for comparing different knowledge (e.g. "fake" wins/losses in simulation)
- How does the quality of MCTS evaluation improve over time, with more search?
- What are the tradeoffs between more, faster simulations or fewer, smarter simulations (e.g. Zen)?

Challenge: Parallel Search

- Can scale up to 2000 cores (Yoshizoe et al, MP-Fuego at UEC Cup 2014/2015)
- New parallel MCTS algorithms such as TDS-df-UCT (Yoshizoe et al 2011)
- Controlling huge search trees is difficult
- Theoretical limits (Segal 2011)

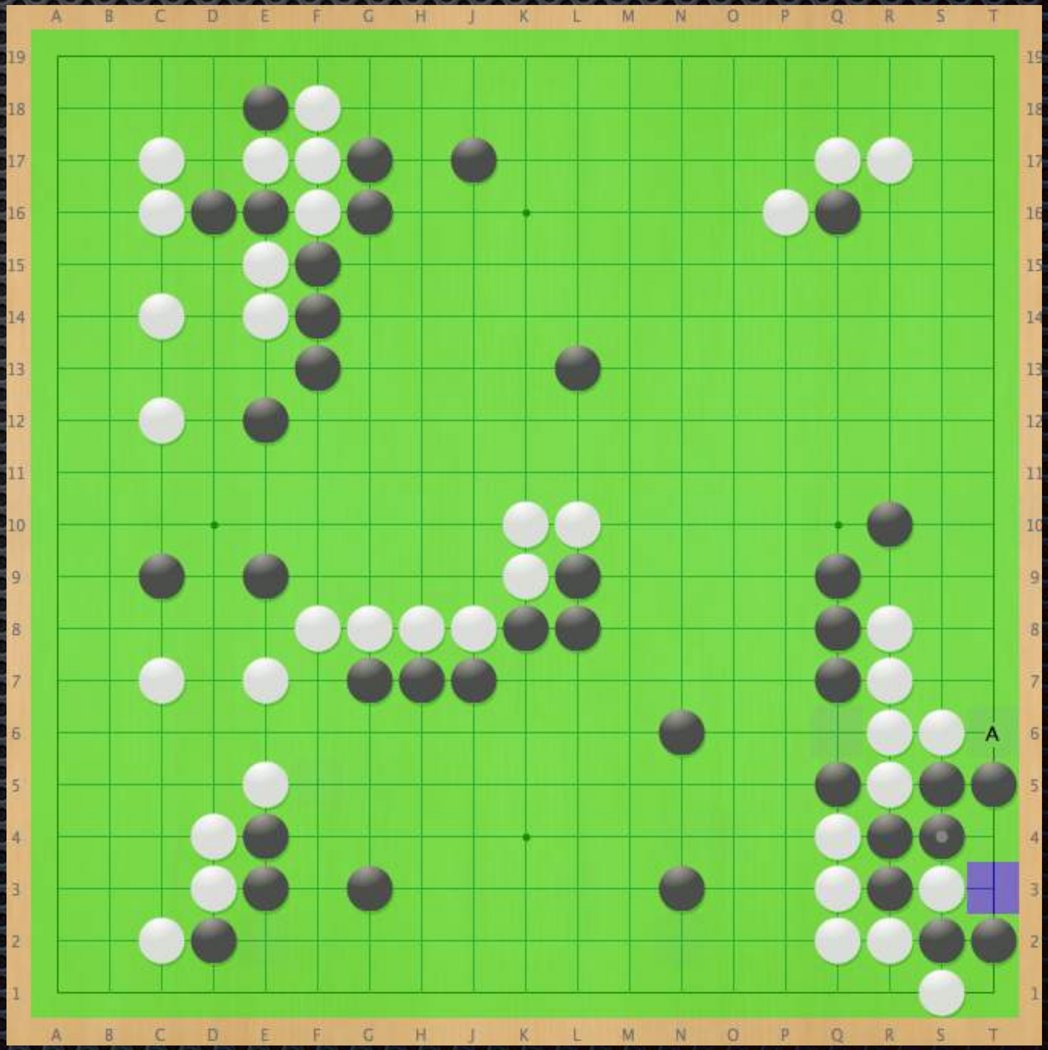
Credits: westgrid.ca, titech.ac.jp

Research Questions

- How to best use large parallel hardware?
- Adapt to changes in network, memory, CPU speed
- Make search fault-tolerant (hardware/software does fail)
- How to test and debug such programs?
- Further improve parallel MCTS algorithms

Challenge: integrate MCTS and DCNN Technologies

- DCNN with no search plays “much nicer looking” Go than Fuego
- DCNN makes a few blunders per game
- Example: analyzed game at http://webdocs.cs.ualberta.ca/~mmueller/fuego/Convolutional-Neural-Network.html

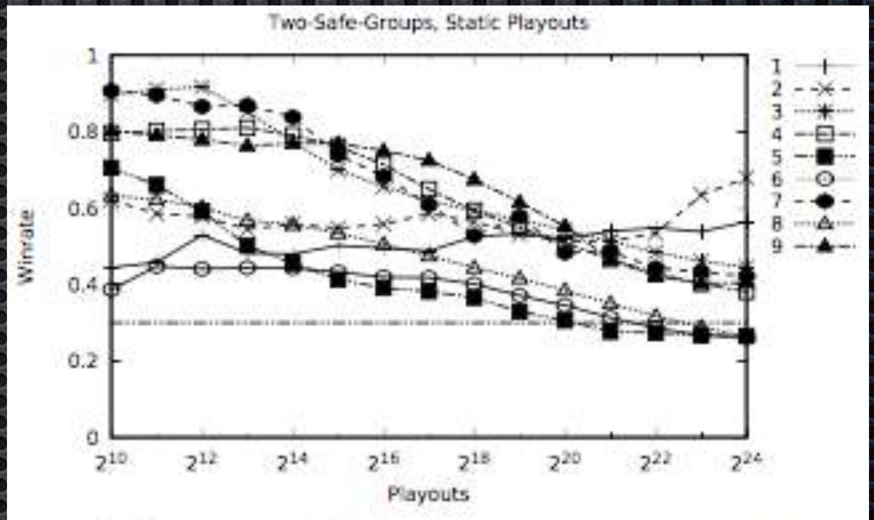


Research Questions

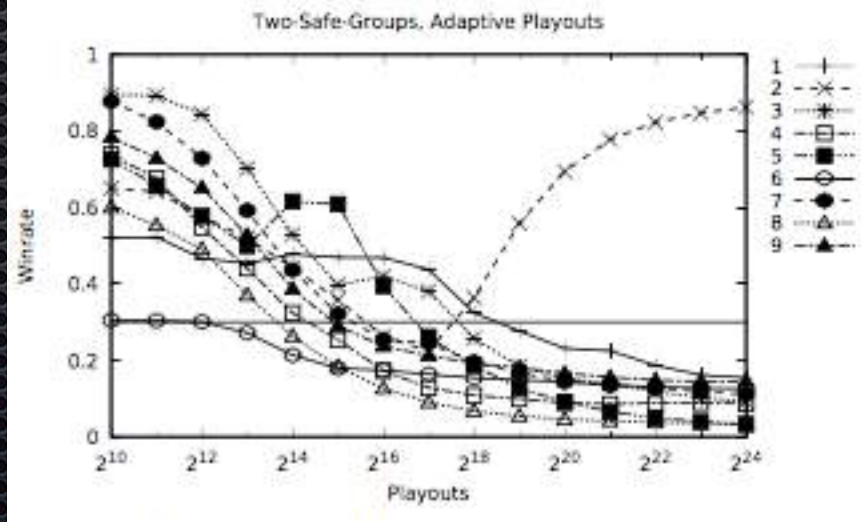
- How to add “slow but strong” evaluation from DCNN to MCTS?
- How to set up the search to overcome blunders and “holes” in knowledge?
- How to use faster DCNN implementations, e.g. on GPU hardware?
- Can we predict for which nodes in tree DCNN evaluation is most useful?

Challenge: Adapt Simulations at Runtime

- Simulations are designed to work “on average”
- Can we make them work better for a specific situation?
- Use reinforcement learning - (Silver et al ICML 2008), (Graf and Platzner, ACG 2015)
- Use RAVE values - (Rimmel et al, CG 2010)



(a) Static Playouts - Testcases 1-9



(c) Adaptive Playouts, Testcases 1-9

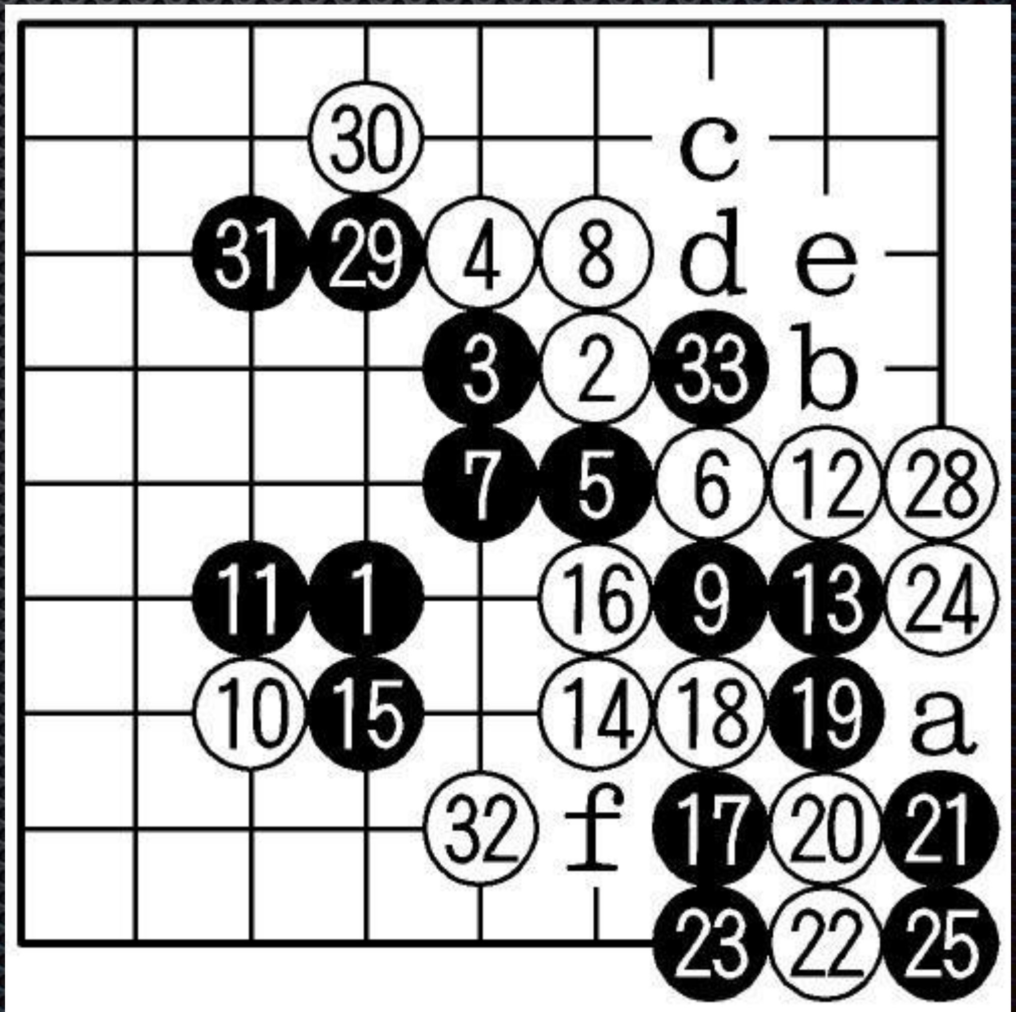
Source: Graf and Platzner 2015

Research Questions

- How to learn exceptions from general rules at runtime?
- How to analyze simulations-so-far?
- How to use the analysis to adapt simulations on the fly?

Challenge: Deep Search - Both Locally and Globally

- 2012, professionals win 6-0 vs Zen on 9x9 board
- Reason: they can search critical lines more deeply
- Huang and Müller (CG 2013): most programs can resolve one life and death fight, but not two at the same time



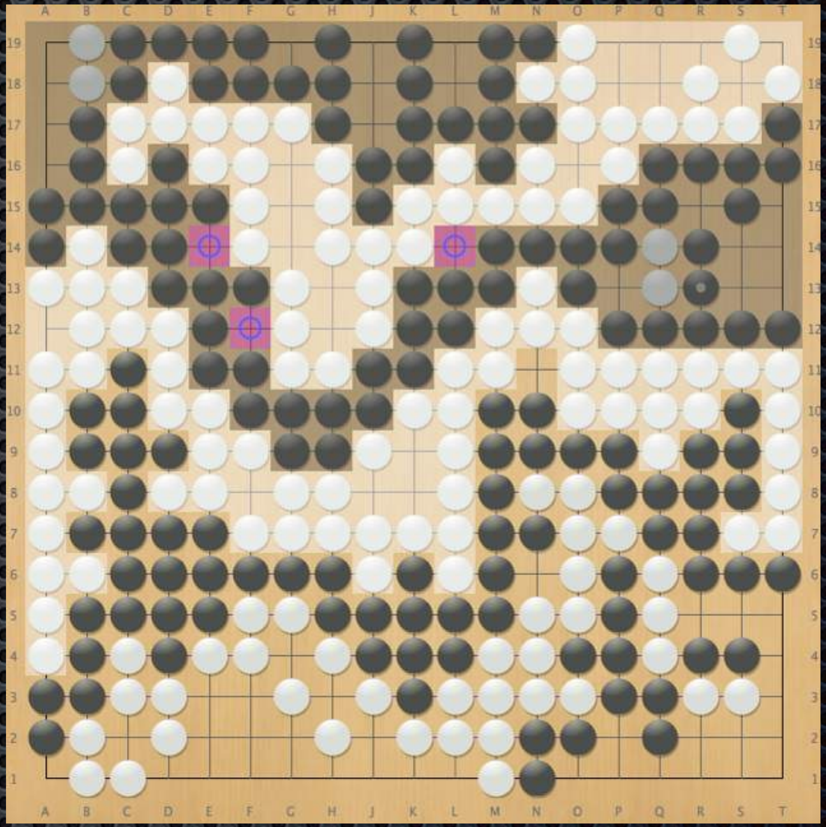
Source: asahi.com

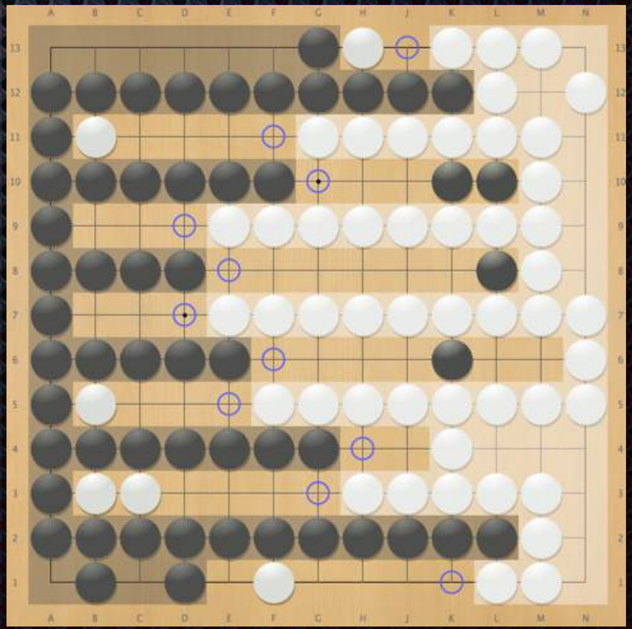
Research Questions

- What is "local search"?
  - Where does it start and stop? What is the goal?
- How to combine local with global search?
  - Example: use local search as a filter
    - Which parts of the board are currently not interesting?
    - Which local moves make sense ?

Challenge: use Exact Methods

- Monte Carlo Simulations
  introduce noise in evaluation
  - Kato: 99% is not enough
    (when humans are 100% correct)
- Go has a large body of exact theory
  - Safety of territory,
    combinatorial game theory for endgames
- Can we play “tractable” positions
  with 100% precision?





Research Questions

- Extend exact methods from puzzles and late endgames (Berlekamp and Wolfe 1994, Müller 1995, 1999) to earlier positions
- Use exact methods on parts of the board, such as corners, territories (Niu and Müller 2004)
- Extend temperature theory from combinatorial games to analyze more difficult earlier positions
(Kao et al, ICGA 2012), (Zhang and Müller AAAI 2015)

Challenge: Win a Match Against Top Human Players

- When will it happen in Go?
  - Simon Lucas: <10 years
  - Your prediction?
- Will it happen at all? It might not. (E.g. shogi, Chinese chess)

Deep Blue vs Kasparov
Source: http://cdn.theatlantic.com

Research Questions

- How to make programs strong enough to challenge humans?
- How to design now for future hardware?
- How to create positions that are difficult for humans?
  - Maybe create complete chaos???
- How to avoid positions where programs are relatively weak?
  - Where humans can read extremely deeply and accurately

Summary of Talk

- Computer Go has come a long way in the last 50 years
- MCTS has given a big boost in improvement
- We are getting closer to best humans, but gap still large
  - See yesterday's games
- Much research remains to be done
- Want more information? See my AAAI-14 tutorial https://webdocs.cs.ualberta.ca/~mmueller/courses/2014-AAAI-games-tutorial/index.html

Thank You!