Computer Go - from the Beginnings to MuZero

Martin Müller
Computing Science
University of Alberta

Topics of my Talk

- My Computer Go Education
- AlphaGo, Alpha Zero and MuZero
- I will tell you what these are, not how they work...
- How do they work? UofA is the right place to find out...

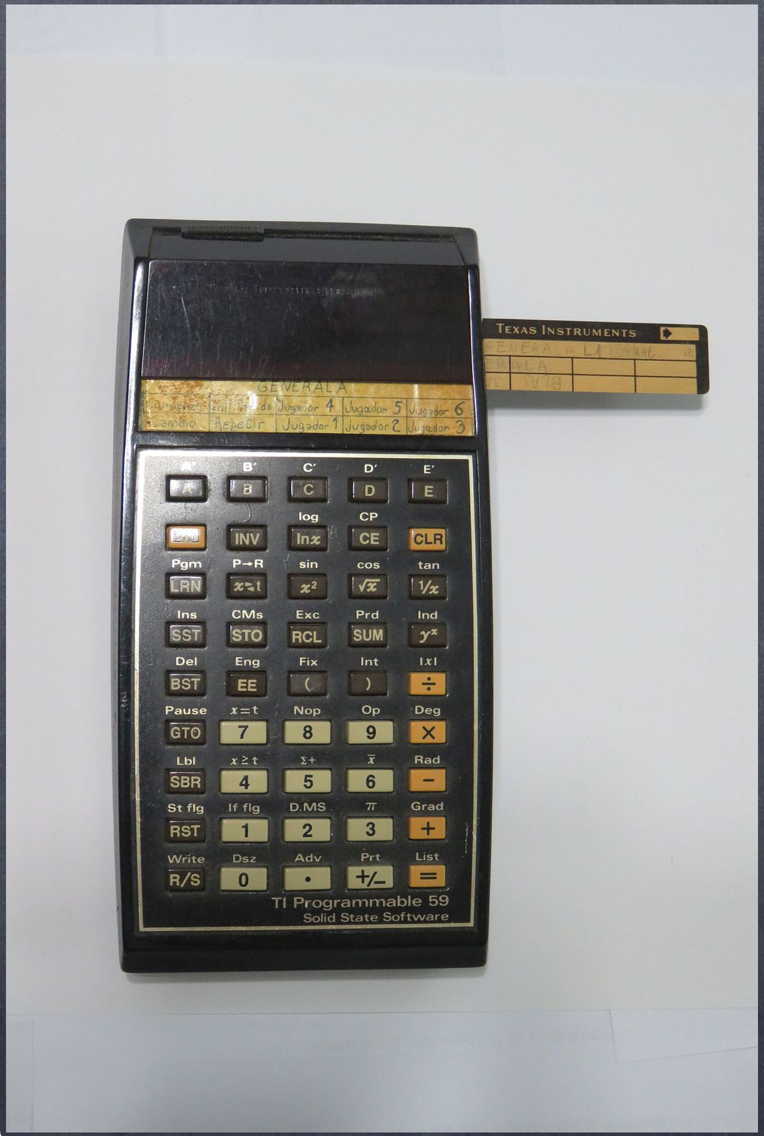
My Computer Go Education

Programming Games – My Story

- Austria, around 1980
- I was about 15 years old
- My math teacher had an early HP programmable calculator
- We played lunar lander...
- No graphics, just numbers...

My First Own Game

- About one year later
- My own TI-59
- Memory: 960 Bytes
- I wrote a Monopoly game (!)
- No graphics, just numbers...
- I sold 3 or 4 copies...



Computer Go Undergrad

About 3 years later

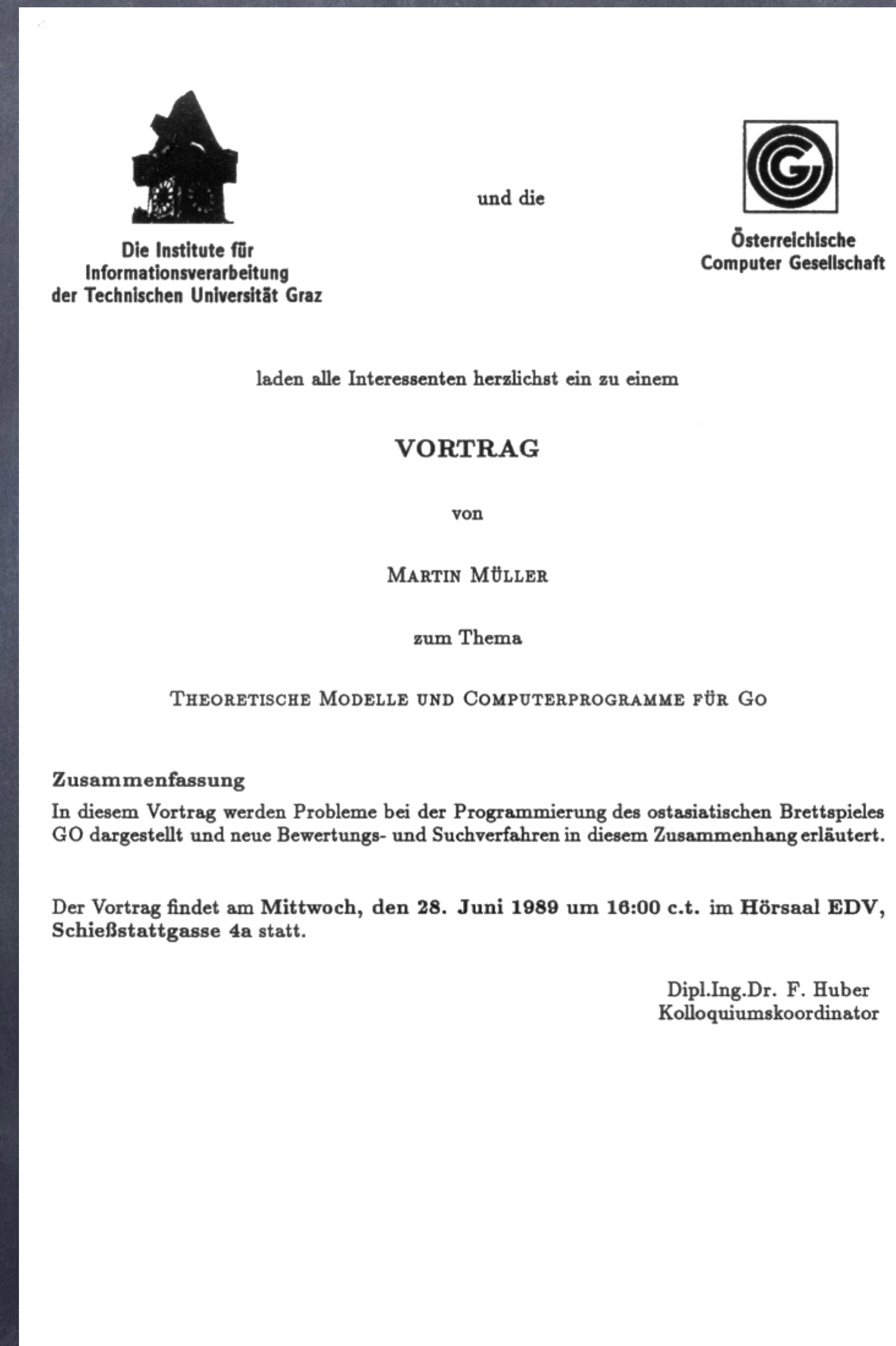
Undergrad in Austria

Supervisor: "let's write a Go program together..."

Me: "OK"

Computer Go Dipl. Ing.

- 5 years later...
- I have a Diploma (masters) thesis on Computer Go
- I'm hooked...
- I can do a PhD at ETH Zurich on Computer Go!



Die Institute für Informationsverarbeitung der Technischen Universität Graz

und die

Österreichische Computer Gesellschaft

laden alle Interessenten herzlichst ein zu einem

VORTRAG

von

MARTIN MÜLLER

zum Thema

THEORETISCHE MODELLE UND COMPUTERPROGRAMME FÜR GO

Zusammenfassung

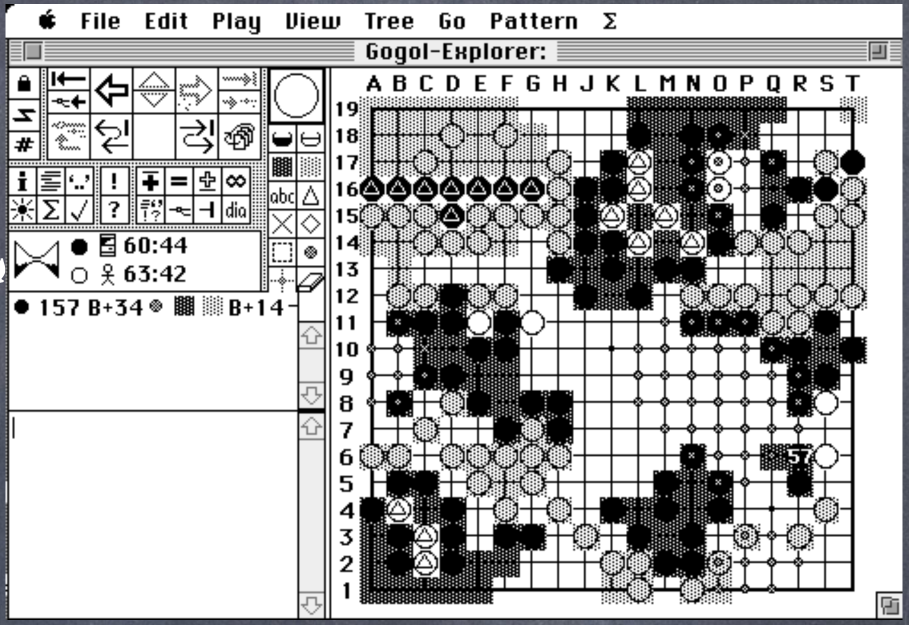
In diesem Vortrag werden Probleme bei der Programmierung des ostasiatischen Brettspieles GO dargestellt und neue Bewertungs- und Suchverfahren in diesem Zusammenhang erläutert.

Der Vortrag findet am **Mittwoch, den 28. Juni 1989 um 16:00 c.t.** im **Hörsaal EDV, Schießstattgasse 4a** statt.

Dipl.Ing.Dr. F. Huber
Kolloquiumskoordinator

Computer Go PhD

- 1989–1995 ETH Zurich, Switzerland
- I join a games research group
- Work for years on Go program "Explorer"
- Never better than mid-level club player
- PhD work: develop algorithms for solving Go endgames



1991 International Computer Go Congress

Here are the results of the Tournament. Programs with equal scores are ranked according to the SST tie-breaking regulations, which give preference first to SDOS, and then to SOS.

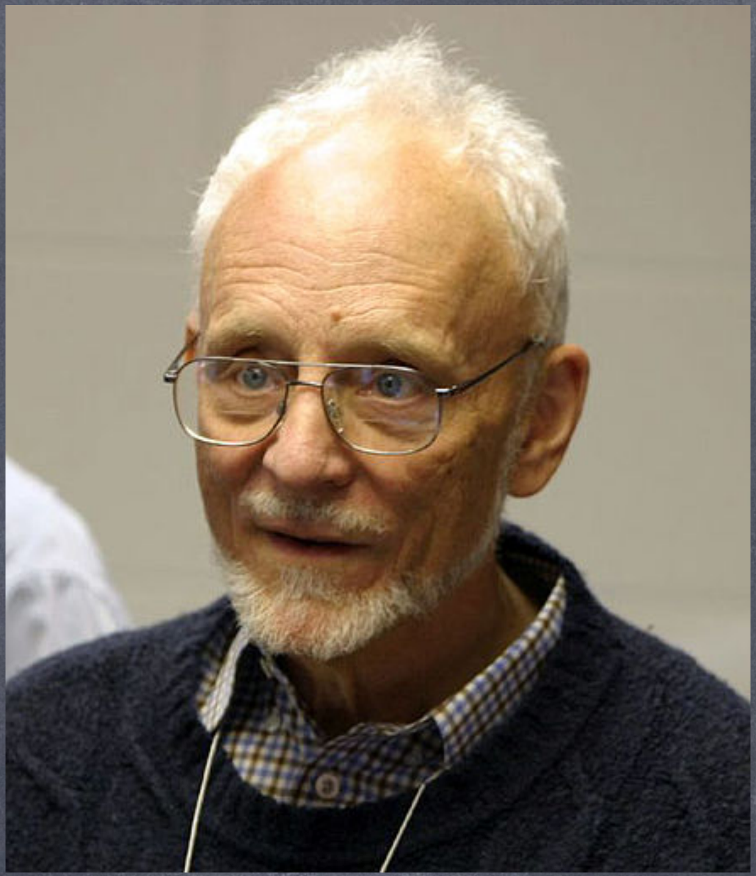
Computer vs. Computer

	Entrant	country	Rd1	Rd2	Rd3	Rd4	Rd5	Rd6	total
1	Goliath	HOL	2W	15W	6W	5W	3W	7W	6-0
2	Go Intellect	USA	1L	bye	15W	6W	7W	5W	5-1
3	Dragon	TAI	10W	6L	4W	11W	1L	8W	4-2
4	Igo III	JAP	6L	10W	3L	9W	12W	11W	4-2
5	Star of Poland	POL	14W	7W	11W	1L	6W	2L	4-2
6	Handtalk	PRC	4W	3W	1L	2L	5L	13W	3-3
7	Stone	TAI	[illegible]	[illegible]	8W	12W	2L	1L	3-3
8	ModGo	[illegible]	[illegible]	[illegible]	[illegible]	14W	11W	3L	3-3
9	Mac	[illegible]	[illegible]	[illegible]	[illegible]	4L	13W	14W	3-3
10	Many Fac[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	13W	14W	12W	3-3
11	Nemesis	[illegible]	[illegible]	[illegible]	[illegible]	3L	8L	4L	2-4
12	Hiratsuka	[illegible]	8W	11L	13W	7L	4L	10L	2-4
13	Explorer	CH	7L	14W	12L	10L	9L	6L	1-5
14	Daihoninbo	JAP	5L	13L	bye	8L	10L	9L	1-5
15	Go	PRC	bye	1L	2L	withdrew			1-5

:(

Computer Go Postdoc

- 1995-2000
- Berkeley
- Switzerland
- Japan
- Work on Go endgames and other games
- Go programs get better, but VERY slowly
- Human knowledge is the bottleneck



Elwyn Berlekamp
(1940 - 2019)

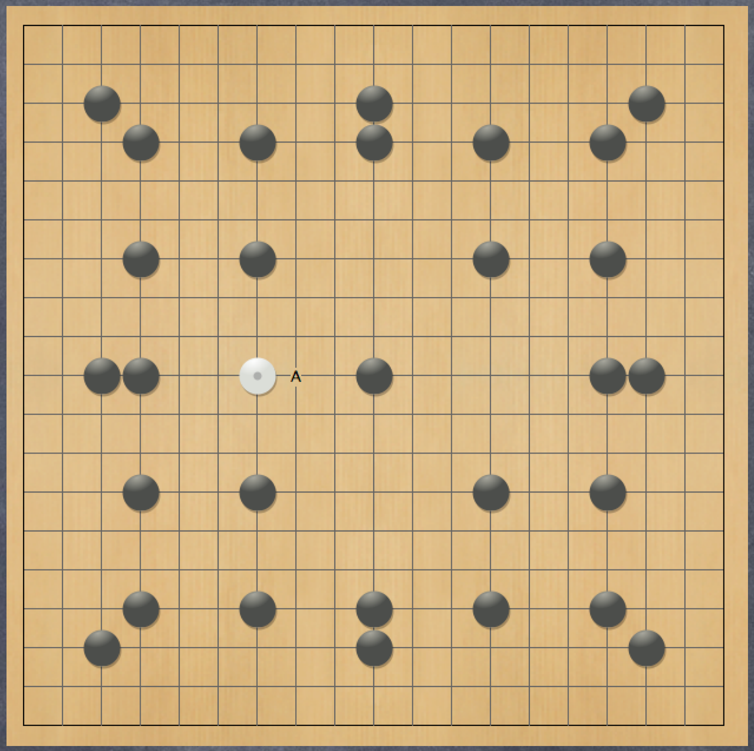
Computer Go in 1998

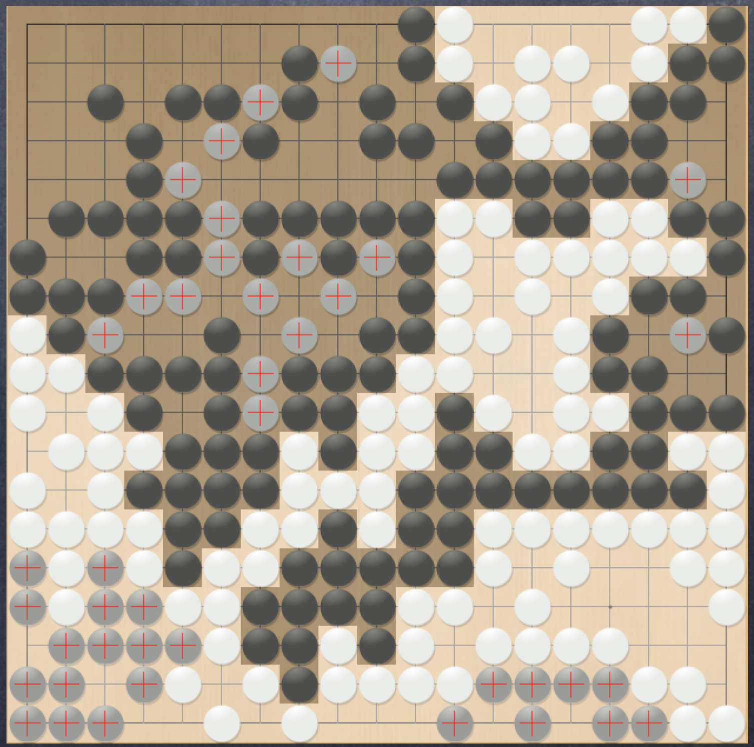
Black: Many Faces of Go, world champion

White: me

Handicap: 29 stones (!)

Result: I won by 6 points





Computer Go Professor

- Game Programming Workshop in Japan 1999
- Invited speaker: Jonathan Schaeffer, UAlberta
- Jonathan: "We have open faculty positions. You should apply"
- me: "OK"
- Fall 2000: joined our department

Computer Go Professor?

- 2000 - 2006, UofA:
- Go is still hard
- A professor needs to publish
- My students and me do lots of other research as well...

A. Kishimoto. Correct and Efficient Search Algorithms in the Presence c

A. Kishimoto and M. Müller. A solution to the GHI problem for depth-firs

A. Botea, M. Enzenberger, M. Müller, and J. Schaeffer. Macro-FF: Impr

A. Botea, M. Müller, and J. Schaeffer. Learning partial-order macros fror
Planning and Scheduling, pages 231-240, Monterey, California, 2005.

A. Kishimoto and M. Müller. Dynamic decomposition search: A divide a
(CIG'05), pages 164-170, 2005.
Erratum: On the final page, "move D6 in R2 makes half an eye" should r

A. Kishimoto and M. Müller. Search versus knowledge for solving life ar

J. Schaeffer, Y. Björnsson, N. Burch, A. Kishimoto, M. Müller, R. Lake, F
Distinguished paper award

2004

A. Botea, M. Enzenberger, M. Müller, and J. Schaeffer. Macro-FF. In bo

M. Müller. Go-related research at the University of Alberta. In T. Ito and T
on Game Informatics). Extended abstract.

J. Zhou and M. Müller. Solving systems of difference constraints increm

A. Botea, M. Müller, and J. Schaeffer. Near optimal hierarchical path-fin

X. Niu and M. Müller. An improved safety solver for computer Go. In J. v
Lecture Notes in Computer Science, pages 97-112, Ramat-Gan, Israel, 2

M. Müller and Z. Li. Locally informed global search for sums of combina
volume 3846 of Lecture Notes in Computer Science, pages 273-284, Ra

A. Kishimoto and M. Müller. A general solution to the graph history interaction p

M. Müller, M. Enzenberger, and J. Schaeffer. Temperature discovery search. In

L. Zhao and M. Müller. Game-SAT: A preliminary report. In Seventh International

A. Botea, M. Müller, and J. Schaeffer. Using component abstraction for automa
International Conference on Automated Planning and Scheduling, pages 181-19

A. Kishimoto and M. Müller. Df-pn in Go: an application to the one-eye problem

X. Niu. Recognizing safe territories and stones in computer Go. Master's thesis,

L. Zhao. Tackling Post's correspondence problem. 22 pages. Accepted 2/2004

2003

Jonathan Schaeffer, Martin Müller, and Yngvi Björnsson, editors. Computers and
Notes in Computer Science. Springer, 2003.

J. Zhou and M. Müller. Depth-first discovery algorithm for incremental topologic

J. Zhou. Incremental search algorithms. Master's thesis, University of Alberta, 2

M. Müller. Conditional combinatorial games, and their application to analyzing c
Erratum: Fig. 1 shows three Nim heaps with 6, 4 and 3 pebbles. The size 6 heap

M. Müller. Proof-set search. In J. Schaeffer, M. Müller, and Y. Björnsson, editors,

A. Botea, M. Müller, and J. Schaeffer. Using abstraction for planning in Sokoban
Science, pages 360-375. Springer Verlag, 2003.

A. Botea, M. Müller, and J. Schaeffer. Extending PDDL for hierarchical planning

A. Botea. Reducing planning complexity with topological abstraction. Proceedi

A. Botea. Reducing planning complexity with topological abstraction. P

L. Zhao. Tackling Post's correspondence problem. In J. Schaeffer, M. M
Verlag, 2003.

A. Kishimoto and M. Müller. A solution to the GHI problem for depth-firs

2002

A. Botea. Using abstraction for heuristic search and planning. In S. Koe
Computer Science, pages 326-327. Springer Verlag, 2002.

N. Bullock. Domineering: Solving large combinatorial search spaces. IC

N. Bullock. Domineering: Solving large combinatorial search spaces. Ma

M. Müller. Counting the score: Position evaluation in computer Go. ICGA

M. Müller. A generalized framework for analyzing capturing races in Go.

M. Müller. Multicriteria evaluation in computer game-playing, and its rela
Scheduling (AIPS-2002) Workshop on Planning and Scheduling using Mu

M. Müller and T. Tegos. Experiments in Computer Amazons. In R. Nowa

T. Tegos. Shooting the last arrow. Master's thesis, University of Alberta,

L. Zhao. Solving and creating difficult instances of Post's corresponden

M. Müller. Computer Go. Artificial Intelligence, 134(1-2):145-179, 2002. (

2001

M. Müller. Solving 5x5 Amazons. In The 6th Game Programming Worksh

M. Müller. Proof Set Search. Revised and improved version of previous r
M. Müller. Global and local game tree search. Information Science

M. Müller. Review: Computer Go 1984 - 2000. In T. Marsland and

M. Müller. Partial Order Bounding: A new Approach to Evaluation i

1996-2000

- 2000
- 1999
- 1998
- 1997
- 1996

2000

M. Müller. Generalized Thermography: A new approach to evaluati
published in Iida, H. (Ed.), Proceedings IJCAI-97 Workshop on Usi

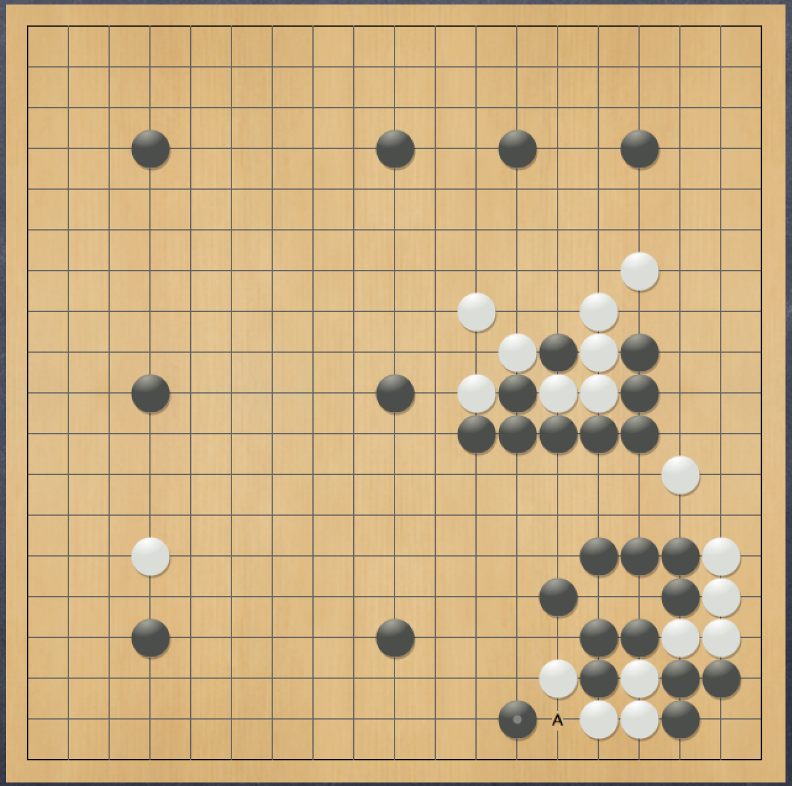
M. Müller. Not like other games - why tree search in Go is different
special session on Heuristic Search and Computer Game Playing.

H. Iida and M. Müller. Report on the Second Open Computer-Ama

MCTS, AlphaGo and Alpha Zero

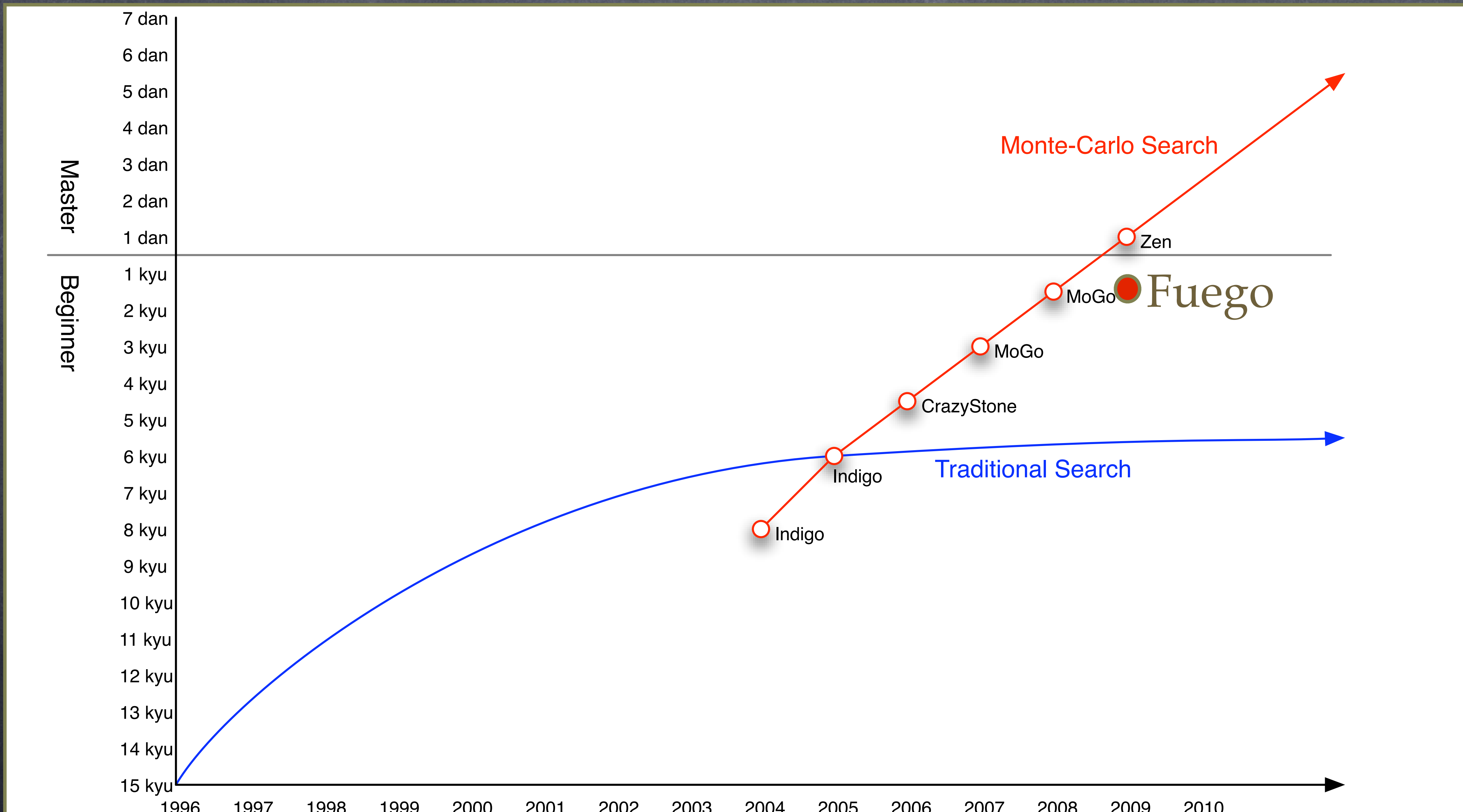
MCTS

- 2006-08: the Monte Carlo Revolution
- Coulom: Monte Carlo Tree Search (MCTS)
- Kocsis and Szepesvari: UCT
- Gelly, Teytaud,...: MoGo program
- Can beat human pros with 8-9 stones handicap
- Suddenly there is hope...



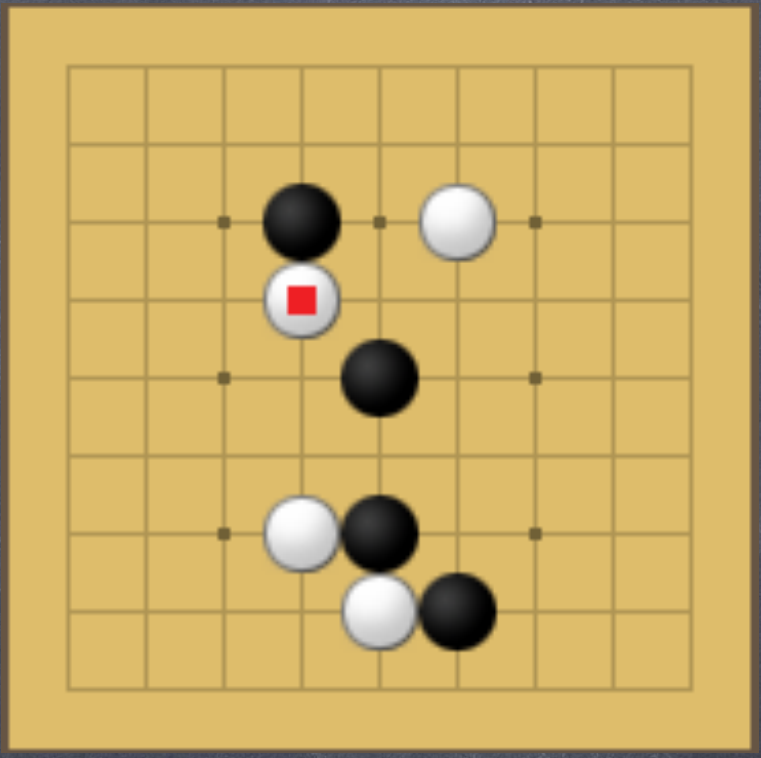
MoGo, 3200 cores vs Kim, 8 Dan pro

Computer Go Progress 1996-2010



Small Board Success

- 2009, 9x9 Go board
- First win vs top human pro
- On even terms, no handicap
- Our program Fuego did it!
- How? MCTS, deep search
- Primitive Go knowledge only



White: Fuego, 80 cores
Black: Chou 9 Dan
White wins by 2.5 points

Dave Silver

- December 2003, Dave Silver: "I am hoping to study for a PhD in computer go and machine learning..."
- Rich Sutton and me: Come to UofA!

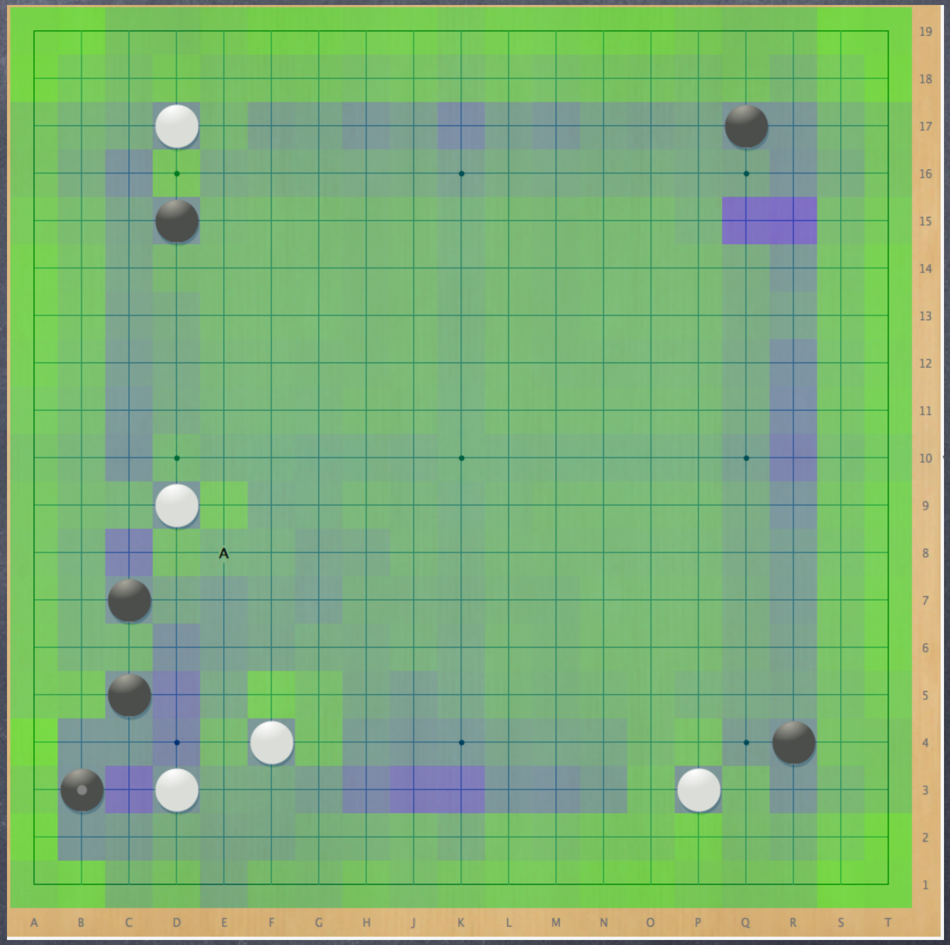
- 2004-2009: Dave's PhD at UofA
- RLGo strongest learning Go program
- Not as strong as MCTS
- Very primitive "features" for knowledge

Computer Go Before AlphaGo

2008-2015 Improve Monte Carlo Tree Search

Add simple Go knowledge

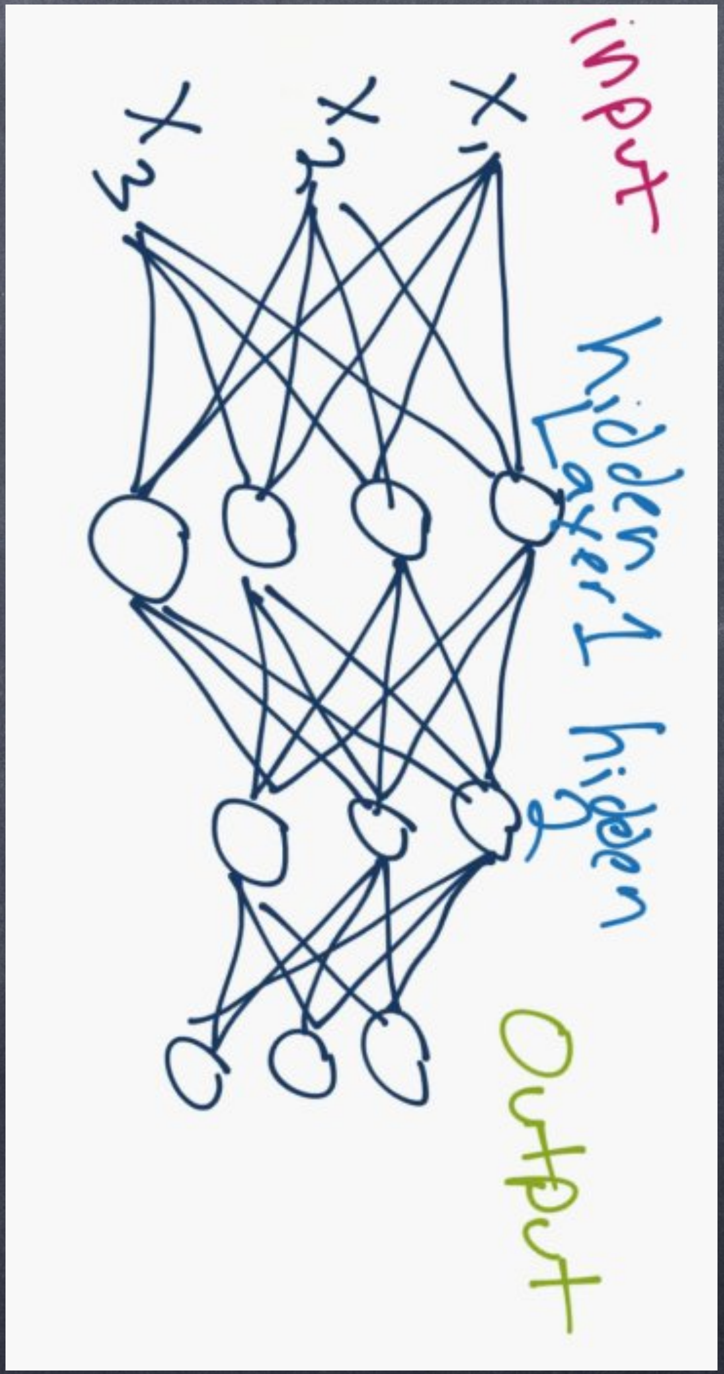
Level: about 3-4 stones handicap from top humans



Knowledge based on simple features in Fuego

Deep Neural Nets

- 2011-2012 deep neural nets start winning image recognition contests
- 2015 used for learning Go knowledge
- At first: learn moves from human master games
- Massively better knowledge than anything we had before

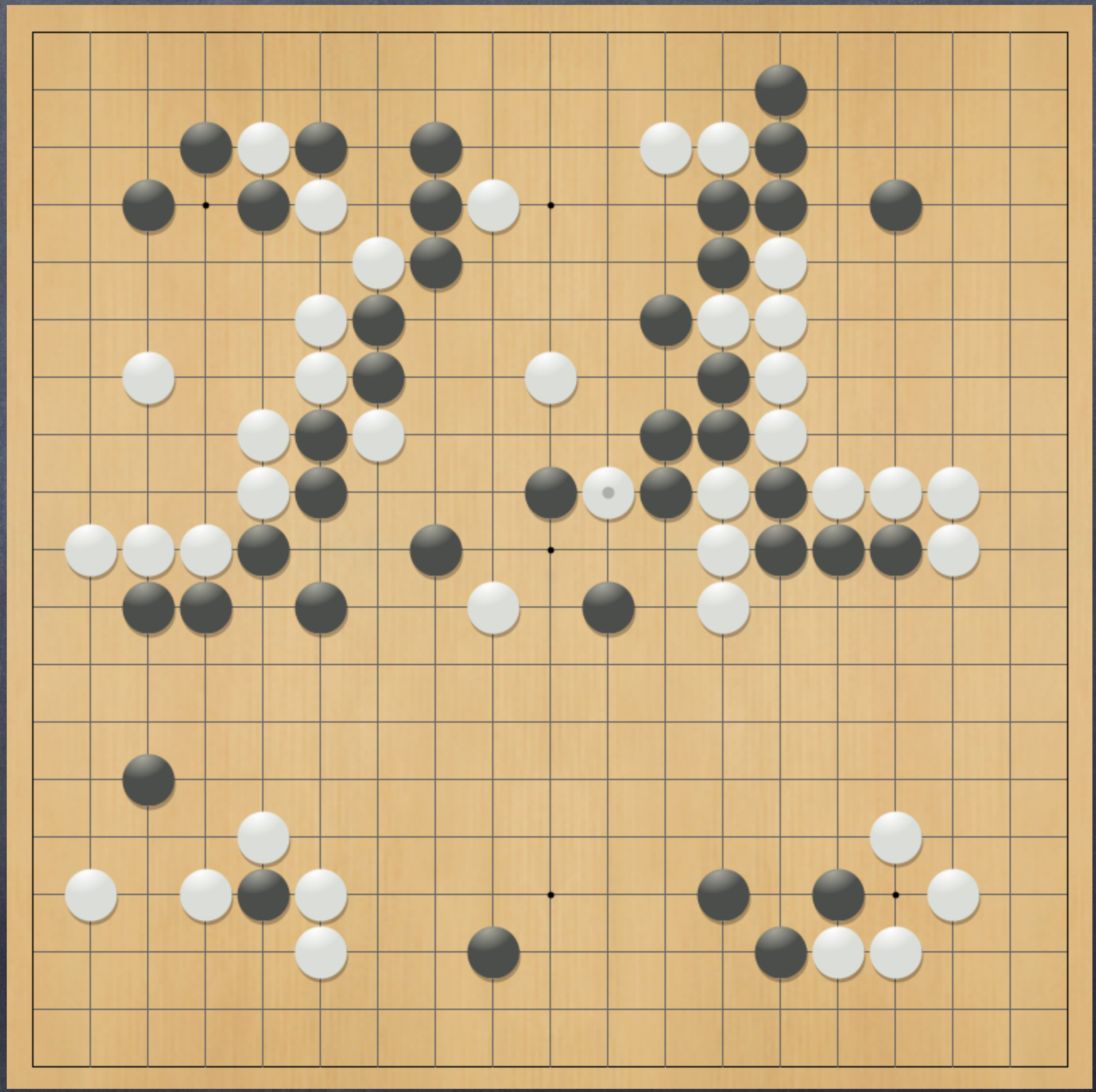


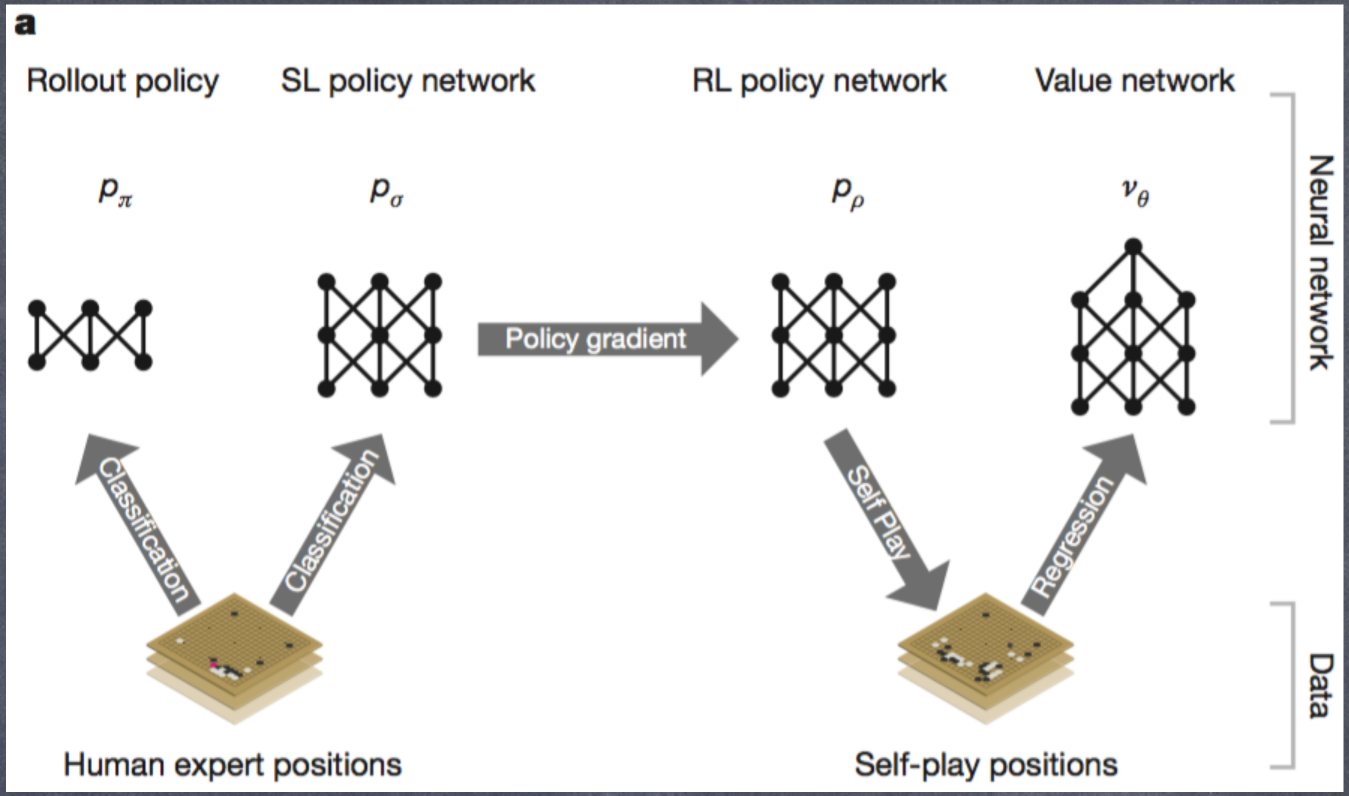
AlphaGo

- Program by DeepMind
- Team lead: Dave Silver
- Combines MCTS, deep networks, RL
- Plays full 19x19 game
- 2015: beats human 2 dan pro on even (no handicap)

More AlphaGo

- March 2016 beats top player Lee Sedol 4:1
- Lee wins game 4
- Human outsearches machine?



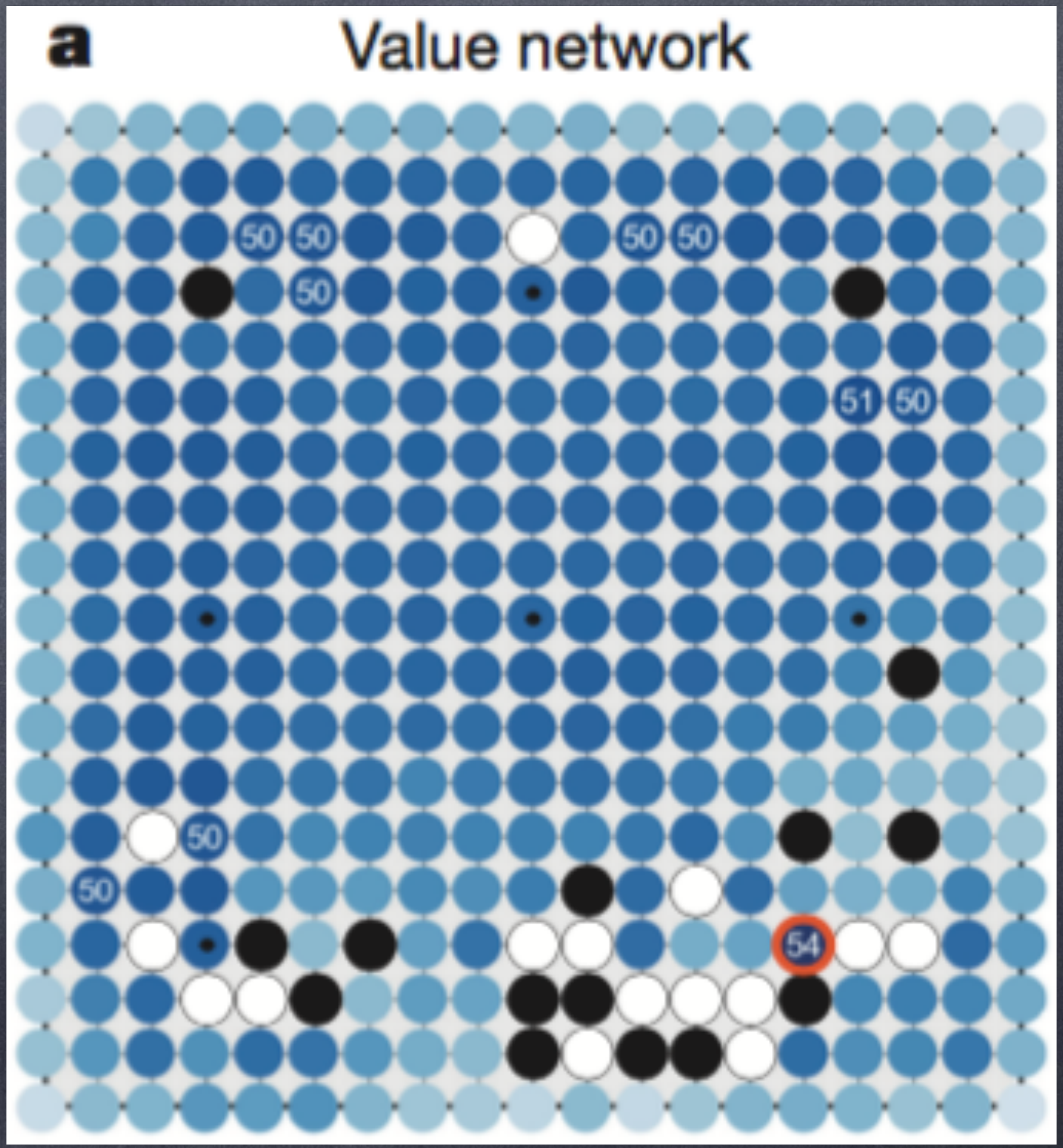


How Did AlphaGo Work?

- The original AlphaGo was very complex
- Four different networks
- Supervised learning, then RL, then regression
- Used in a massively parallel system
- Large number of both CPU and GPU

How Did It Work? (2)

- Search is MCTS
- Two main neural nets:
- Policy net proposes good moves to search
- Value net evaluates positions

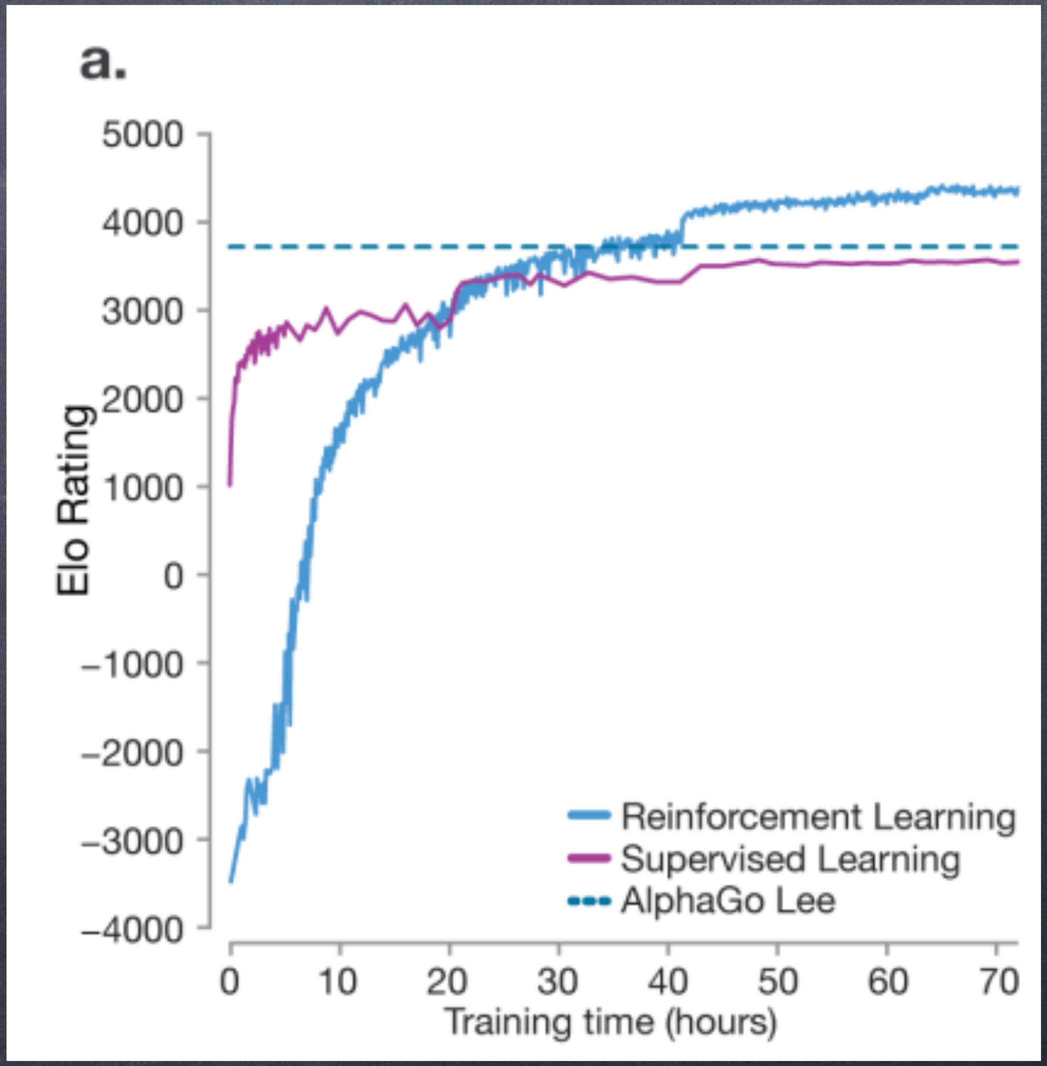


AlphaGo vs Ke Jie

- 2017 match
  vs world #1 Ke Jie
- Improved version
  of AlphaGo
- Result: 3-0 for machine
- AlphaGo retires from
  competitive play

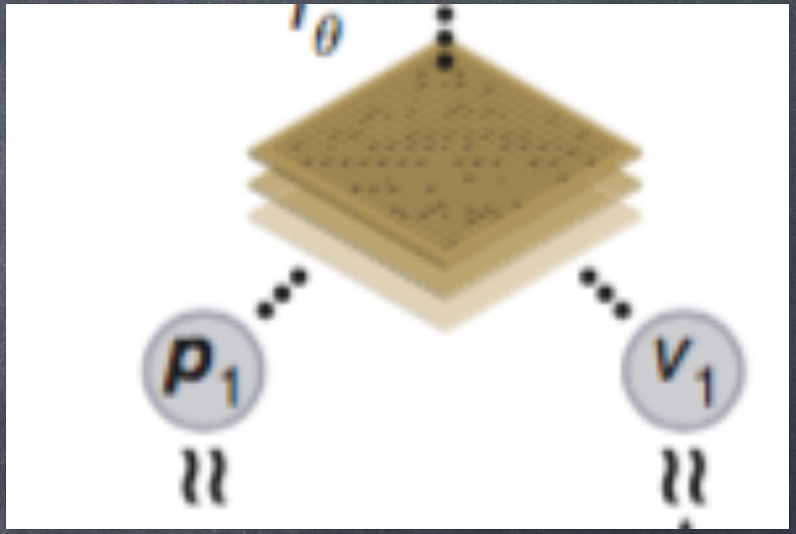
AlphaGo Zero

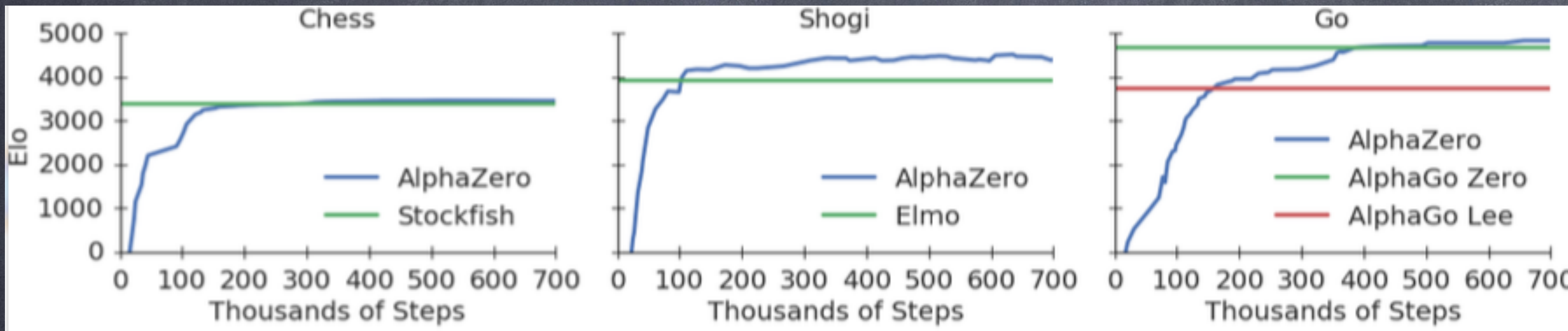
- October 2017 article in Nature "Mastering the game of Go without human knowledge"
- New simplified architecture
- Learns entirely from self play using RL
- Only human knowledge: rules of game
- Stronger than previous AlphaGo



How did It Work?

- One network, two different "heads" for output
- Learn policy and value together
- New architecture: deep residual network (resnet) learns better
- Search: still MCTS
- Stronger network, could use smaller computer



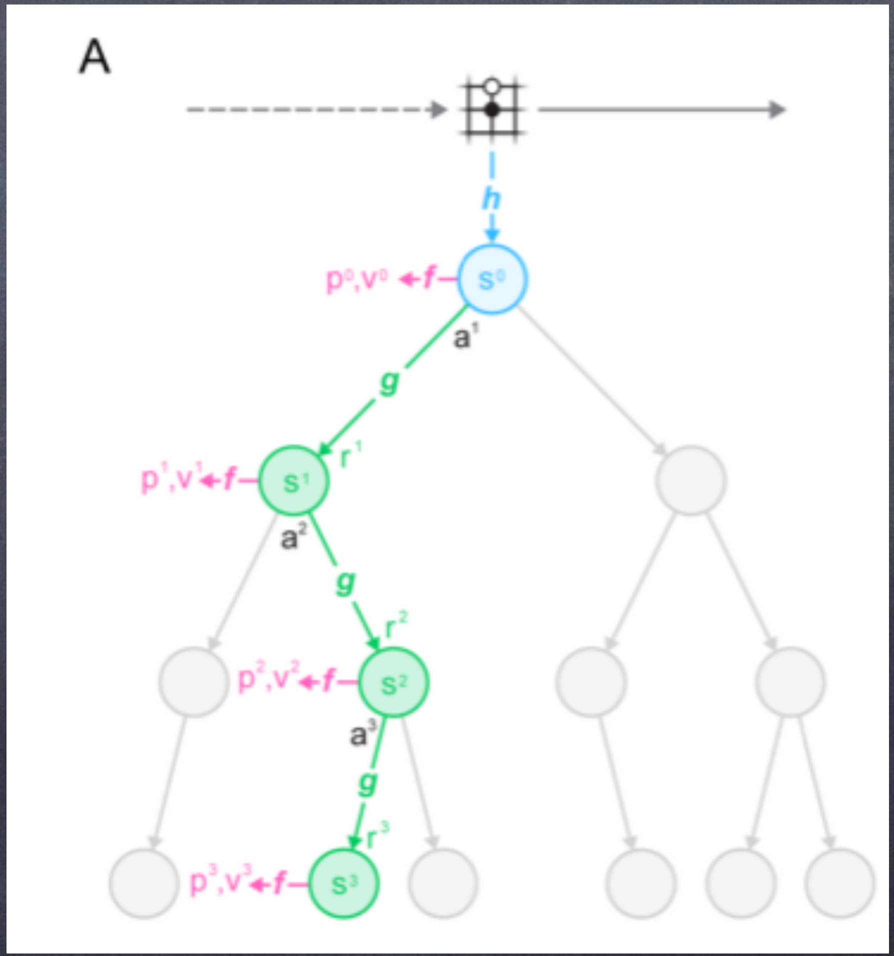


Alpha Zero

- Early version 2017, final version 2018 in Science
- Simplify, remove more Go-specific tricks
- Learn chess and shogi as well
- Beat top chess, shogi programs
- Learn from only rules of game by selfplay

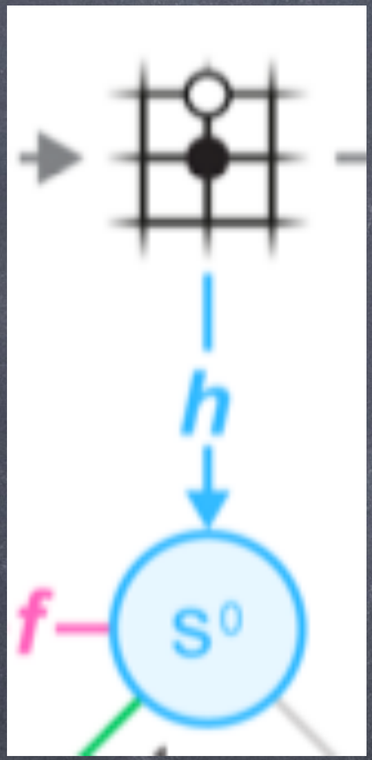
MuZero

- Fall 2019 arXiv preprint
- Newest program in the AlphaGo line
- Novelty: it is not even given the rules of the game
- Plays Go, chess, shogi, and also Atari games



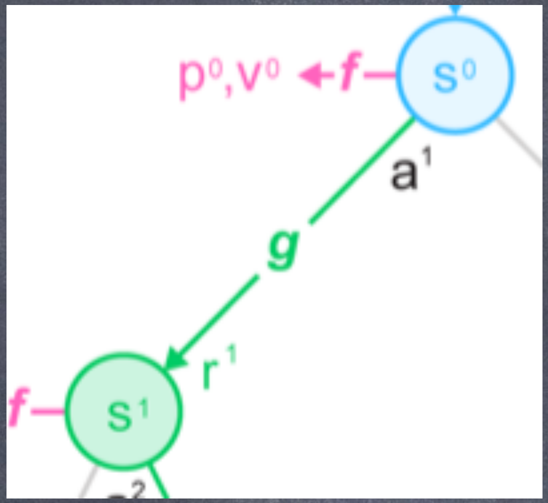
MuZero – How does It Work?

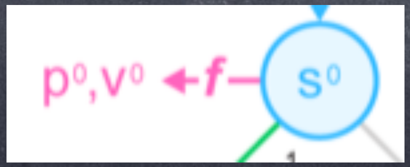
- Input: game records with correct (legal) moves
- Learns three neural nets:
- First net: h
  Maps from raw game state to a learned internal state representation



MuZero – How does It Work?

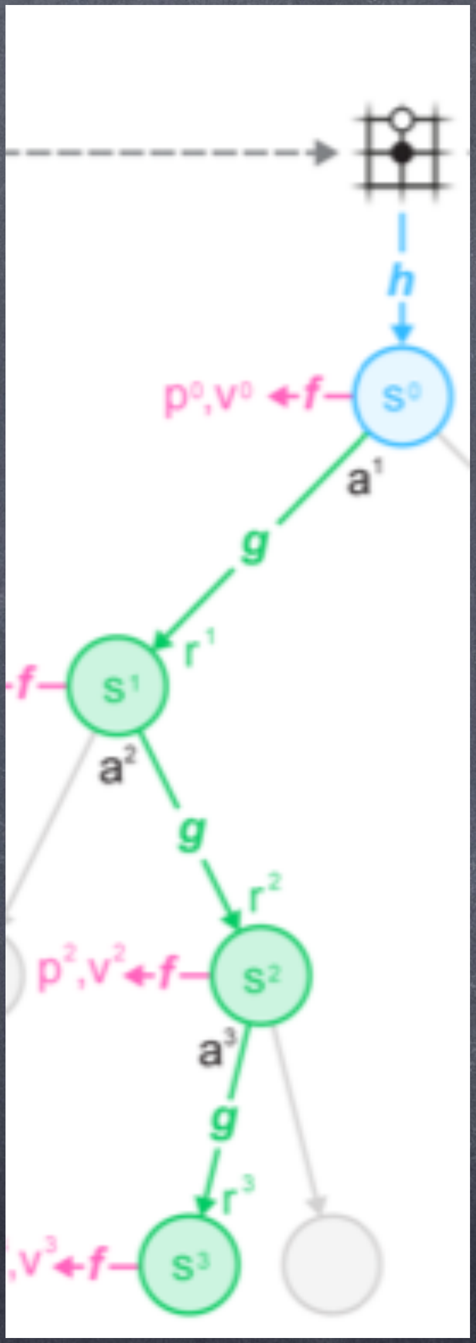
- Second net: g
- Learns how to "make a move" in the internal representation
- Third net: f
- Computes policy and value, as in Alpha Zero, but from the internal representation, not the game itself



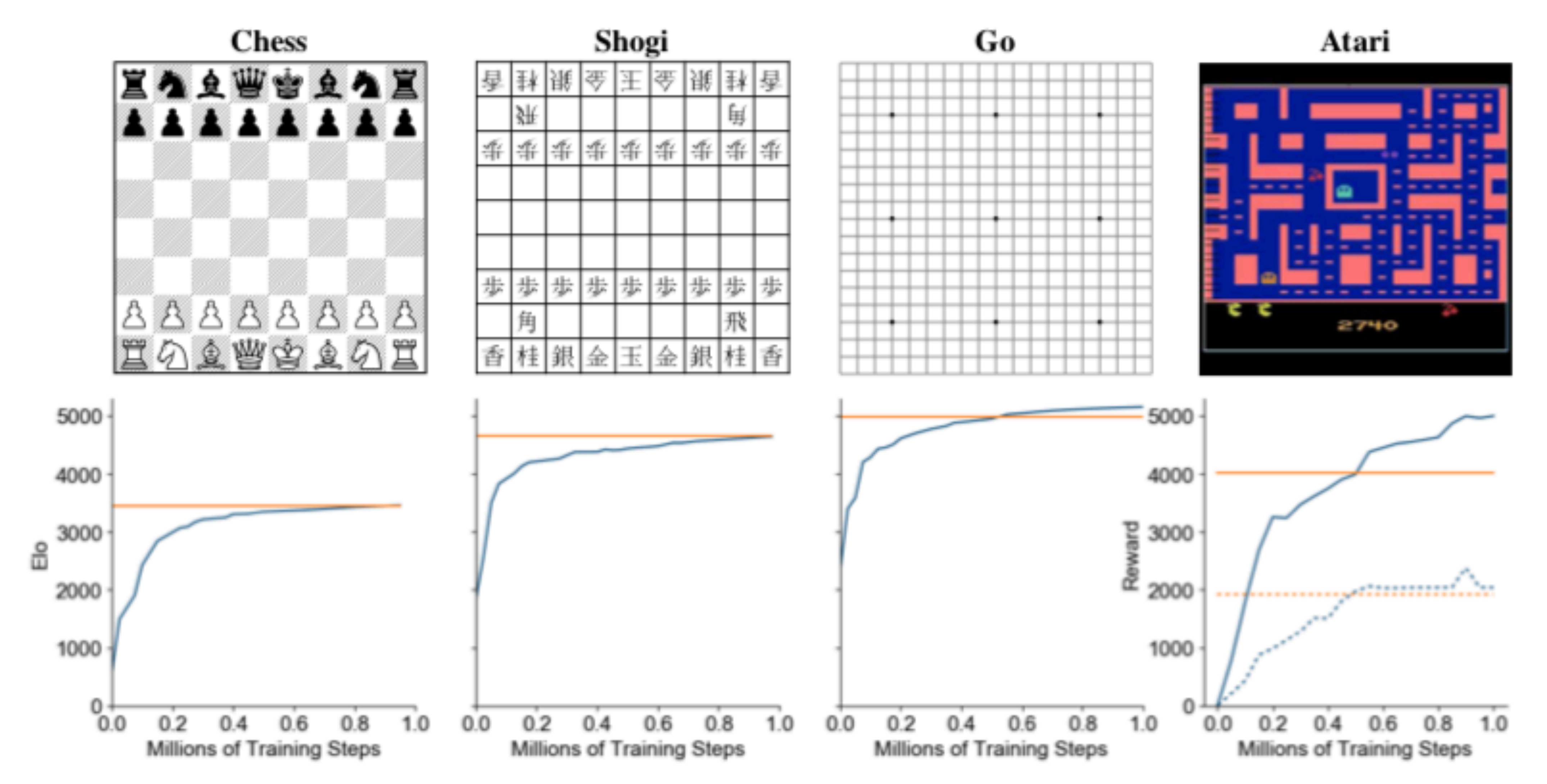


MuZero - How does It Work?

- Learned model has errors
- Errors compound with depth
- Searches only a few steps (about 5) deep
- Still, super strong results



MuZero Results



AlphaGo and Us

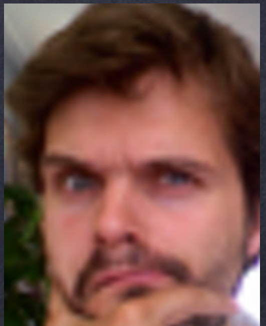
- AlphaGo was "Big Science"
- Dozens of developers, millions of dollars in hardware and computing costs
- What is the role of our university in all of this?
- We contributed lots of:
    1. Basic research
    2. Training











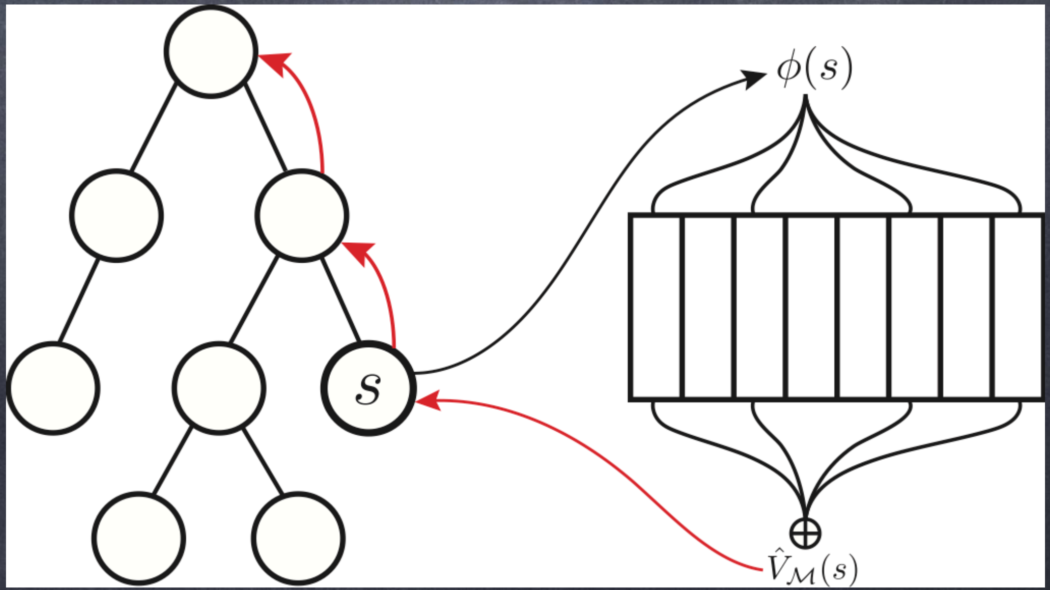
UAlberta Research and Training

- Citation list from first AlphaGo paper
- Papers with UofA people in yellow

1. Allis, L. V. *Searching for Solutions in Games and Artificial Intelligence*. PhD thesis, Univ. Limburg, Maastricht, The Netherlands (1994).
2. van den Herik, H., Uiterwijk, J. W. & van Rijswijck, J. Games solved: now and in the future. *Artif. Intell.* **134,** 277–311 (2002).
3. Schaeffer, J. The games computers (and people) play. *Advances in Computers* **52,** 189–266 (2000).
4. Campbell, M., Hoane, A. & Hsu, F. Deep Blue. *Artif. Intell.* **134,** 57–83 (2002).
5. Schaeffer, J. *et al.* A world championship caliber checkers program. *Artif. Intell.* **53,** 273–289 (1992).
6. Buro, M. From simple features to sophisticated evaluation functions. In *1st International Conference on Computers and Games*, 126–145 (1999).
7. Müller, M. Computer Go. *Artif. Intell.* **134,** 145–179 (2002).
8. Tesauro, G. & Galperin, G. On-line policy improvement using Monte-Carlo search. In *Advances in Neural Information Processing*, 1068–1074 (1996).
9. Sheppard, B. World-championship-caliber Scrabble. *Artif. Intell.* **134,** 241–275 (2002).
10. Bouzy, B. & Helmstetter, B. Monte-Carlo Go developments. In *10th Internationa* *Conference on Advances in Computer Games*, 159–174 (2003).
11. Coulom, R. Efficient selectivity and backup operators in Monte-Carlo tree search. In *5th International Conference on Computers and Games*, 72–83 (2006).
12. Kocsis, L. & Szepesvári, C. Bandit based Monte-Carlo planning. In *15th European Conference on Machine Learning*, 282–293 (2006).
13. Coulom, R. Computing Elo ratings of move patterns in the game of Go. *ICGA J.* **30,** 198–208 (2007).
14. Baudiš, P. & Gailly, J.-L. Pachi: State of the art open source Go program. In *Advances in Computer Games*, 24–38 (Springer, 2012).
15. Müller, M., Enzenberger, M., Arneson, B. & Segal, R. Fuego – an open-source framework for board games and Go engine based on Monte-Carlo tree search. *IEEE Trans. Comput. Intell. AI in Games* **2,** 259–270 (2010).
16. Gelly, S. & Silver, D. Combining online and offline learning in UCT. In *17th International Conference on Machine Learning*, 273–280 (2007).
17. Krizhevsky, A., Sutskever, I. & Hinton, G. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 1097–1105 (2012).
18. Lawrence, S., Giles, C. L., Tsoi, A. C. & Back, A. D. Face recognition: a convolutional neural-network approach. *IEEE Trans. Neural Netw.* **8,** 98–113 (1997).
19. Mnih, V. *et al.* Human-level control through deep reinforcement learning. *Nature* **518,** 529–533 (2015).
20. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521,** 436–444 (2015
21. Stern, D., Herbrich, R. & Graepel, T. Bayesian pattern ranking for move prediction in the game of Go. In *International Conference of Machine Learning* 873–880 (2006).
22. Sutskever, I. & Nair, V. Mimicking Go experts with convolutional neural networks. In *International Conference on Artificial Neural Networks*, 101–110 (2008).
23. Maddison, C. J., Huang, A., Sutskever, I. & Silver, D. Move evaluation in Go us deep convolutional neural networks. *3rd International Conference on Learnin* *Representations* (2015).
24. Clark, C. & Storkey, A. J. Training deep convolutional neural networks to play go. In *32nd International Conference on Machine Learning*, 1766–1774 (2015).
25. Williams, R. J. Simple statistical gradient-following algorithms for connectior reinforcement learning. *Mach. Learn.* **8,** 229–256 (1992).
26. Sutton, R., McAllester, D., Singh, S. & Mansour, Y. Policy gradient methods fo reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, 1057–1063 (2000).
27. Sutton, R. & Barto, A. *Reinforcement Learning: an Introduction* (MIT Press, 199
28. Schraudolph, N. N., Dayan, P. & Sejnowski, T. J. Temporal difference learning of position evaluation in the game of Go. *Adv. Neural Inf. Process. Syst.* **6,** 817–824 (1994).
29. Enzenberger, M. Evaluation in Go by a neural network using soft segmentati In *10th Advances in Computer Games Conference*, 97–108 (2003). 267.
30. Silver, D., Sutton, R. & Müller, M. Temporal-difference search in computer Go *Mach. Learn.* **87,** 183–219 (2012).
31. Levinovitz, A. The mystery of Go, the ancient game that computers still can't win. *Wired Magazine* (2014).
32. Mechner, D. All Systems Go. *The Sciences* **38,** 32–37 (1998).
33. Mandziuk, J. Computational intelligence in mind games. In *Challenges for Computational Intelligence*, 407–442 (2007).
34. Berliner, H. A chronology of computer chess and its literature. *Artif. Intell.* **10** 201–214 (1978).
35. Browne, C. *et al.* A survey of Monte-Carlo tree search methods. *IEEE Trans. Comput. Intell. AI in Games* **4,** 1–43 (2012).
36. Gelly, S. *et al.* The grand challenge of computer Go: Monte Carlo tree search and extensions. *Commun. ACM* **55,** 106–113 (2012).
37. Coulom, R. Whole-history rating: A Bayesian rating system for players of time-varying strength. In *International Conference on Computers and Games*, 113–124 (2008).
38. KGS. Rating system math. http://www.gokgs.com/help/rmath.html.
39. Littman, M. L. Markov games as a framework for multi-agent reinforcem learning. In *11th International Conference on Machine Learning*, 157–163 (1994).
40. Knuth, D. E. & Moore, R. W. An analysis of alpha-beta pruning. *Artif. Intel* 293–326 (1975).
41. Sutton, R. Learning to predict by the method of temporal differences. *Mach. Learn.* **3,** 9–44 (1988).
42. Baxter, J., Tridgell, A. & Weaver, L. Learning to play chess using temporal differences. *Mach. Learn.* **40,** 243–263 (2000).
43. Veness, J., Silver, D., Blair, A. & Uther, W. Bootstrapping from game tree s In *Advances in Neural Information Processing Systems* (2009).
44. Samuel, A. L. Some studies in machine learning using the game of checkers II - recent progress. *IBM J. Res. Develop.* **11,** 601–617 (1967).
45. Schaeffer, J., Hlynka, M. & Jussila, V. Temporal difference learning applied to high-performance game-playing program. In *17th International Joint Conference on Artificial Intelligence*, 529–534 (2001).
46. Tesauro, G. TD-gammon, a self-teaching backgammon program, achieves master-level play. *Neural Comput.* **6,** 215–219 (1994).
47. Dahl, F. Honte, a Go-playing program using neural nets. In *Machines that lear* *to play games*, 205–223 (Nova Science, 1999).
48. Rosin, C. D. Multi-armed bandits with episode context. *Ann. Math. Artif. Intell.* **61,** 203–230 (2011).
49. Lanctot, M., Winands, M. H. M., Pepels, T. & Sturtevant, N. R. Monte Carlo tree search with heuristic evaluations using implicit minimax backups. In *IEEE Conference on Computational Intelligence and Games*, 1–8 (2014).
50. Gelly, S., Wang, Y., Munos, R. & Teytaud, O. Modification of UCT with patterns Monte-Carlo Go. Tech. Rep. 6062, INRIA (2006).
51. Silver, D. & Tesauro, G. Monte-Carlo simulation balancing. In *26th Internation* *Conference on Machine Learning*, 119 (2009).
52. Huang, S.-C., Coulom, R. & Lin, S.-S. Monte-Carlo simulation balancing in practice. In *7th International Conference on Computers and Games*, 81–92 (Springer-Verlag, 2011).
53. Baier, H. & Drake, P. D. The power of forgetting: improving the last-good-repl policy in Monte Carlo Go. *IEEE Trans. Comput. Intell. AI in Games* **2,** 303–309 (2010).
54. Huang, S. & Müller, M. Investigating the limits of Monte-Carlo tree search methods in computer Go. In *8th International Conference on Computers and Games*, 39–48 (2013).
55. Segal, R. B. On the scalability of parallel UCT. *Computers and Games* **6515,** 36–47 (2011).
56. Enzenberger, M. & Müller, M. A lock-free multithreaded Monte-Carlo tree search algorithm. In *12th Advances in Computer Games Conference*, 14–20 (2009).
57. Huang, S.-C., Coulom, R. & Lin, S.-S. Time management for Monte-Carlo tree search applied to the game of Go. In *International Conference on Technologie* *and Applications of Artificial Intelligence*, 462–466 (2010).
58. Gelly, S. & Silver, D. Monte-Carlo tree search and rapid action value estimatio in computer Go. *Artif. Intell.* **175,** 1856–1875 (2011).

What's Next?

- Research continues
- Examples from our group:
- 3-head neural net
- Memory-augmented MCTS
- Exploration in SAT solving
- RL and search in even more general settings



Interested?

- UofA is the right place
- Many related undergrad courses: 250, 296, 350, 355, 366, 382, 397, 450, 455, 366, 497
- Many faculty work in games and/or RL
- Companies/nonprofits in town: DeepMind, Amii, Huawei, Borealis, …
- More coming…

Summary

- Overview of Computer Go and especially DeepMind's programs
- From human engineered to machine-learned solutions
- Search plays a key role for both learning and actual use
- Huge success in games
- Much work remains to apply methods in the real world