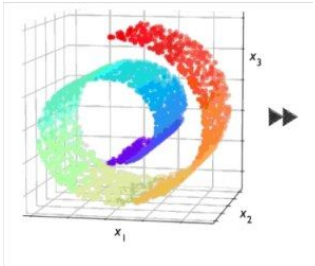
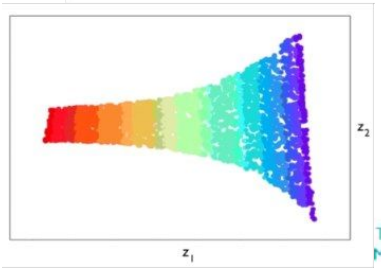
01/29/2020

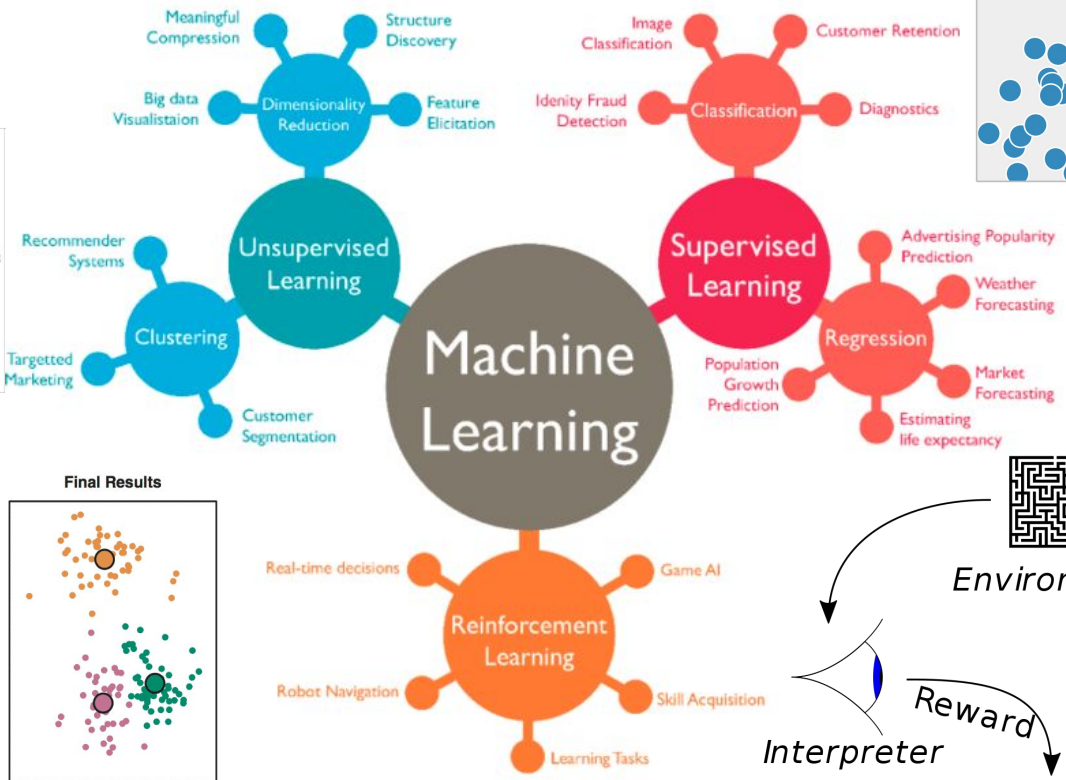
Recent Advances in Reinforcement Learning (with a focus on AlphaGo)

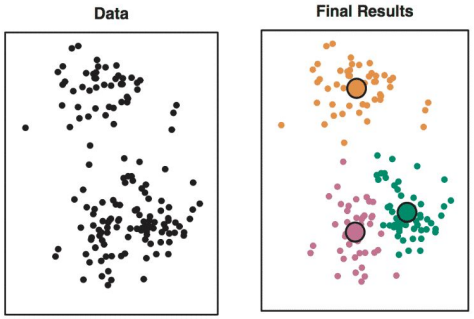
Patrick Scholz
Division of Computer Assisted Medical Interventions

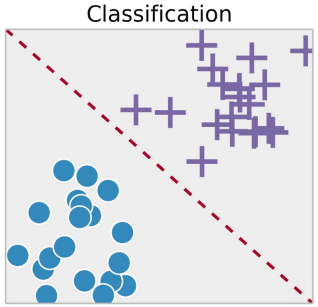
Taxonomic position of RL

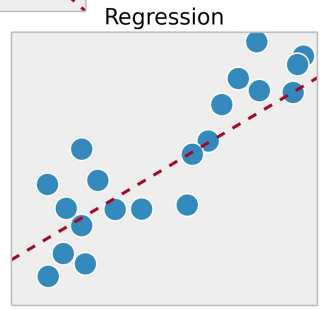


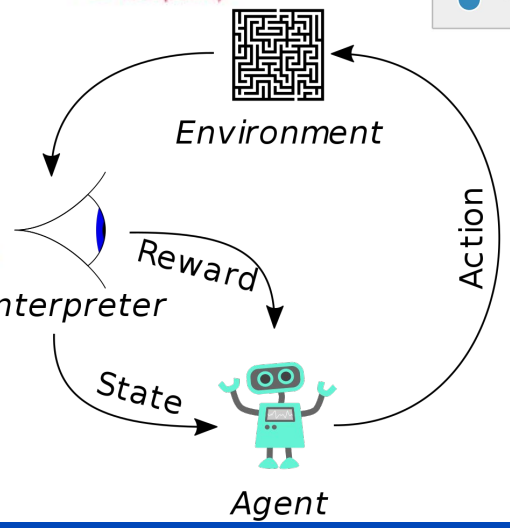










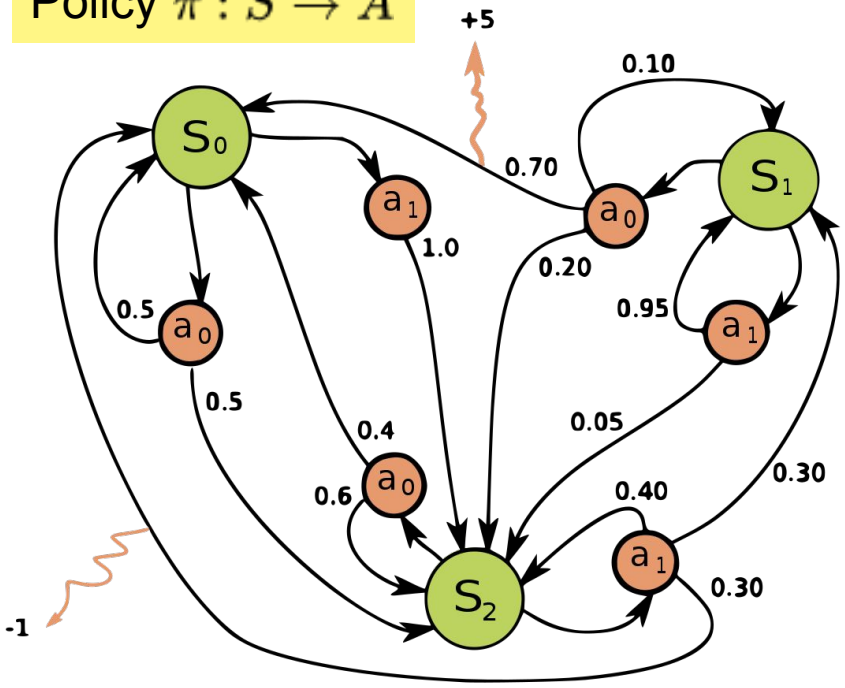


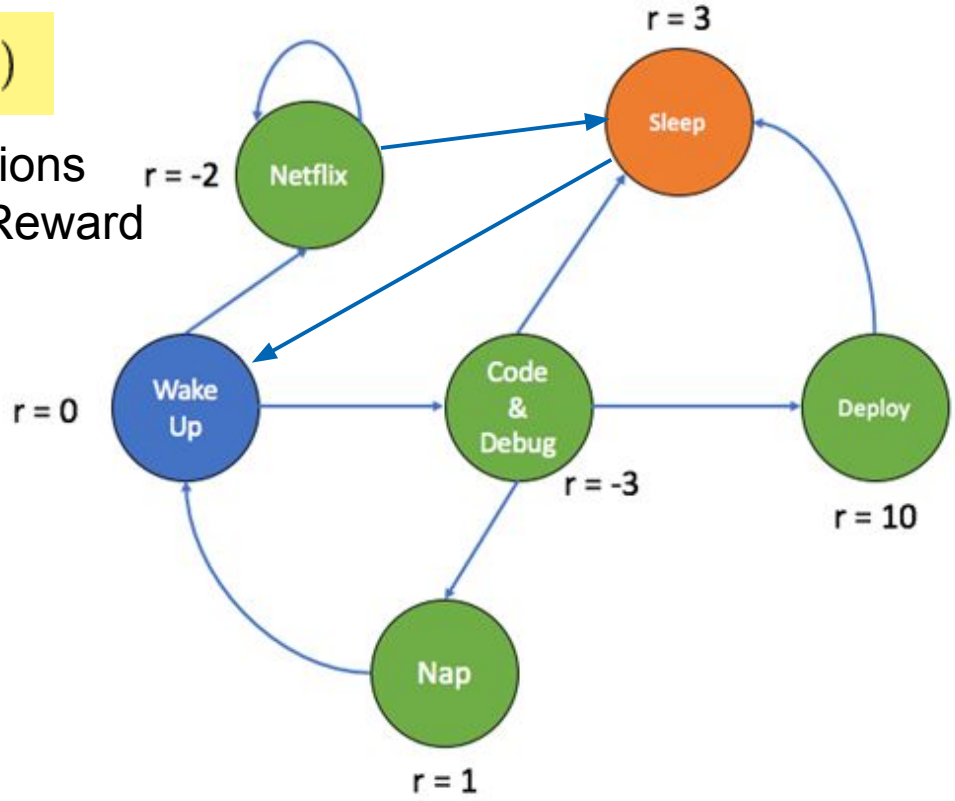
Basics of RL

Markov Decision Process (S, A, P_a, R_a)

S – States
A – Possible Actions
P – Transition Probability
R – Immediate Reward

Policy $\pi : S \rightarrow A$





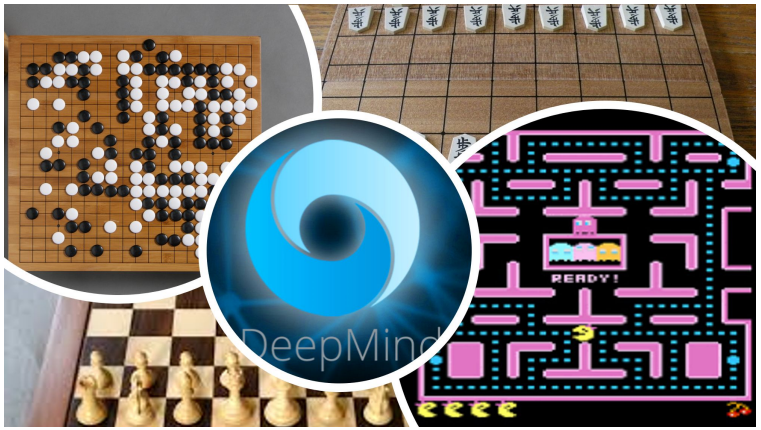
Cumulative reward

$$E[\sum_{t=0}^{\infty} \gamma^t R_{a_t}(s_t, s_{t+1})]$$

Deep RL within the last years wrt AlphaGo

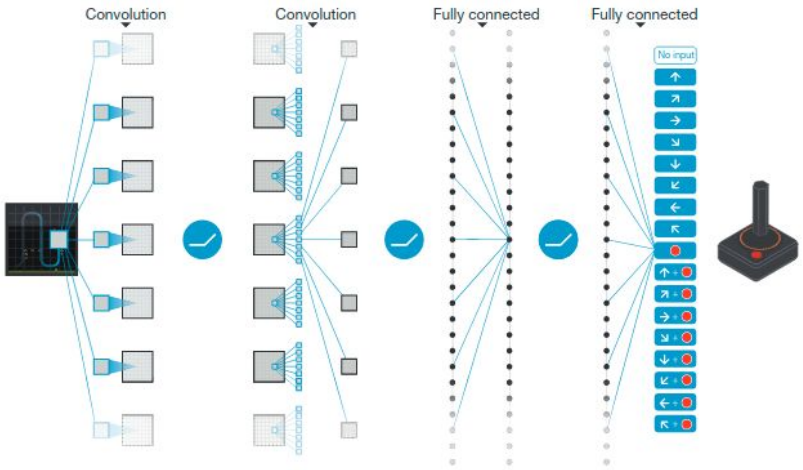
AlphaGo | AlphaGo Zero | AlphaZero | MuZero

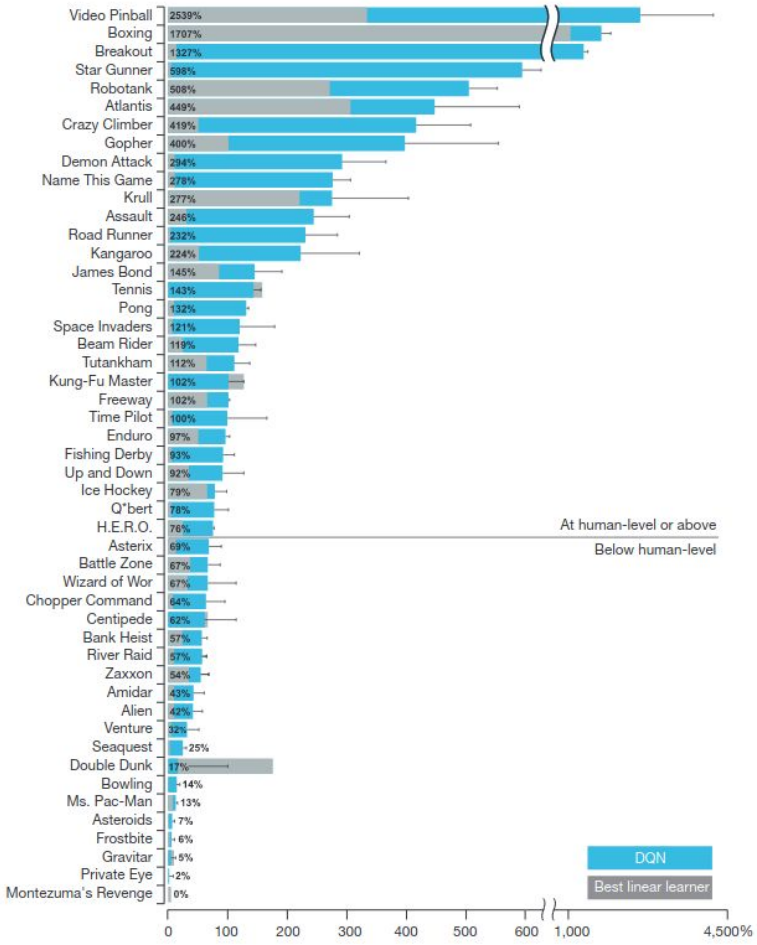
2015 | 2016 | 2017 | 2018 | 2019



dkfz.

“Deep” Learning and Reinforcement learning



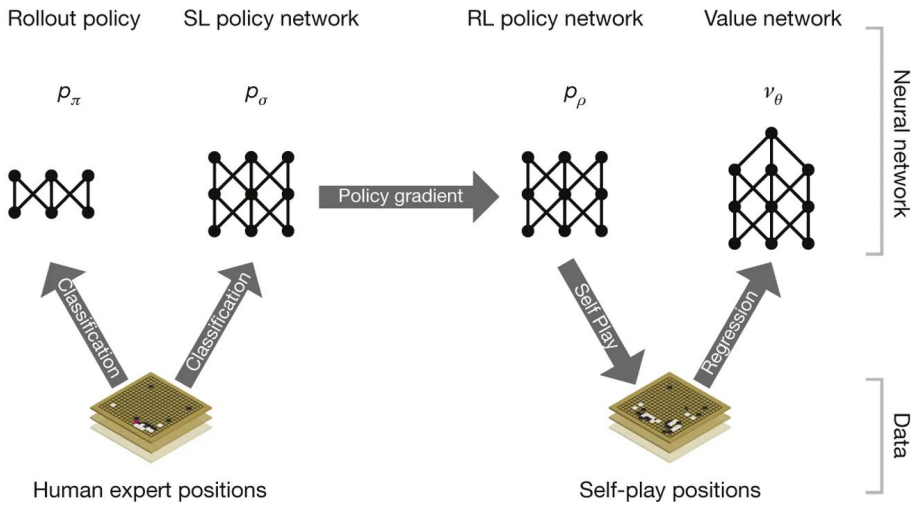


Mnih, V., Kavukcuoglu, K., Silver, D. et al. ‘Human-level control through deep reinforcement learning’. Nature 518, 529–533 (2015). https://doi.org/10.1038/nature14236

„Go“ as the next holy grail

Using expert moves for supervised learning

Playing against earlier versions to generate data

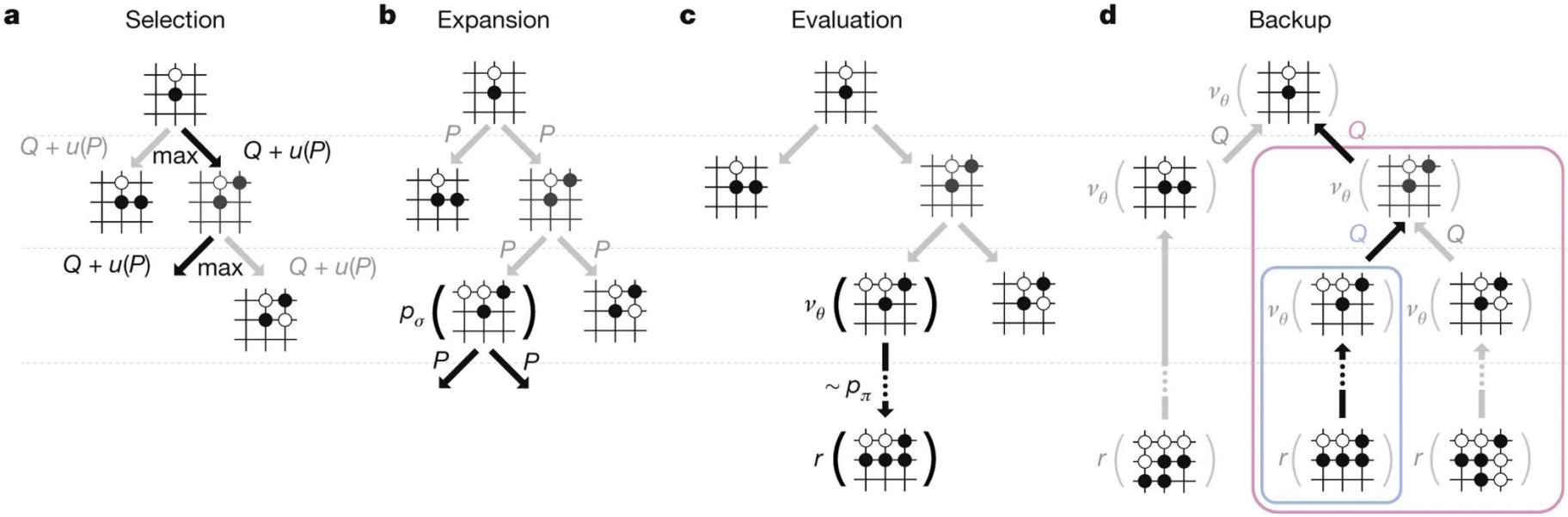


Defeated Lee Sedol (world champion) in a regular match 4:1 (using 48 TPUs)

Silver, D., Huang, A., Maddison, C. et al. ‘Mastering the game of Go with deep neural networks and tree search’. Nature 529, 484–489 (2016). https://doi.org/10.1038/nature16961

„Go“ as the next holy grail

Monte Carlo Tree Search



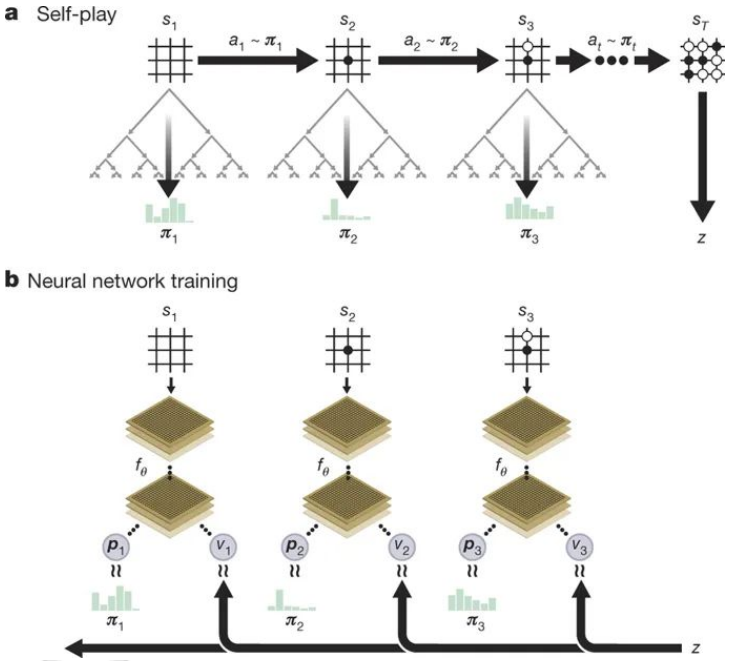
Silver, D., Huang, A., Maddison, C. et al. ‘Mastering the game of Go with deep neural networks and tree search’. Nature 529, 484–489 (2016). https://doi.org/10.1038/nature16961

Dropping initial human input

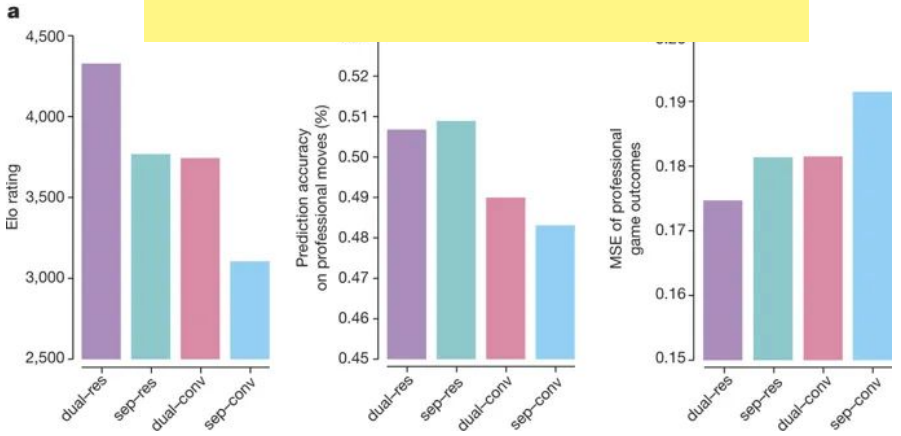
Major design changes:

- using MCTS action distribution to train
- combining policy and value network
- switching to ResNet architecture
- no hand-crafted input features any more

Feature	# of planes	Description
Stone colour	3	Player stone / opponent stone / empty
Ones	1	A constant plane filled with 1
Turns since	8	How many turns since a move was played
Liberties	8	Number of liberties (empty adjacent points)
Capture size	8	How many opponent stones would be captured
Self-atari size	8	How many of own stones would be captured
Liberties after move	8	Number of liberties after this move is played
Ladder capture	1	Whether a move at this point is a successful ladder capture
Ladder escape	1	Whether a move at this point is a successful ladder escape
Sensibleness	1	Whether a move is legal and does not fill its own eyes
Zeros	1	A constant plane filled with 0
Play		



Defeated AlphaGo after 72h under same conditions 100:0 (using 4 TPUs)



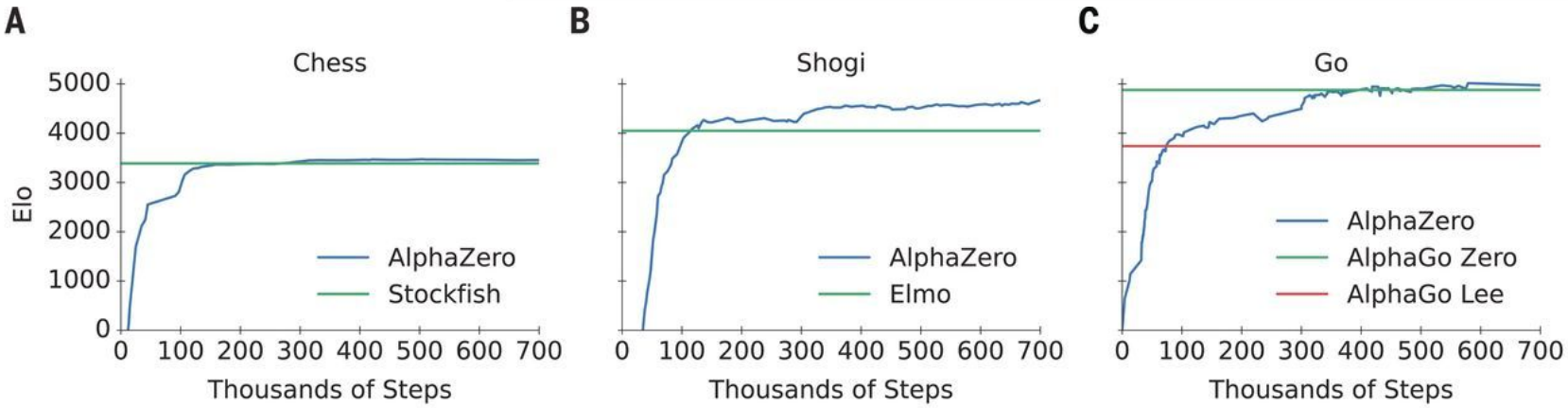
Silver, D., Schrittwieser, J., Simonyan, K. et al. 'Mastering the game of Go without human knowledge'. Nature 550, 354–359 (2017). https://doi.org/10.1038/nature24270

Generalizing input/output representation

Major design changes:

- including draws
- no augmentation exploitation any more
- continuously updating instead of choosing a winner after iteration
- always same hyper-parameters

Go		Chess		Shogi	
Feature	Planes	Feature	Planes	Feature	Planes
P1 stone	1	P1 piece	6	P1 piece	14
P2 stone	1	P2 piece	6	P2 piece	14
		Repetitions	2	Repetitions	3
				P1 prisoner count	7
				P2 prisoner count	7
Colour	1	Colour	1	Colour	1
		Total move count	1	Total move count	1
		P1 castling	2		
		P2 castling	2		
		No-progress count	1		
Total	17	Total	119	Total	362

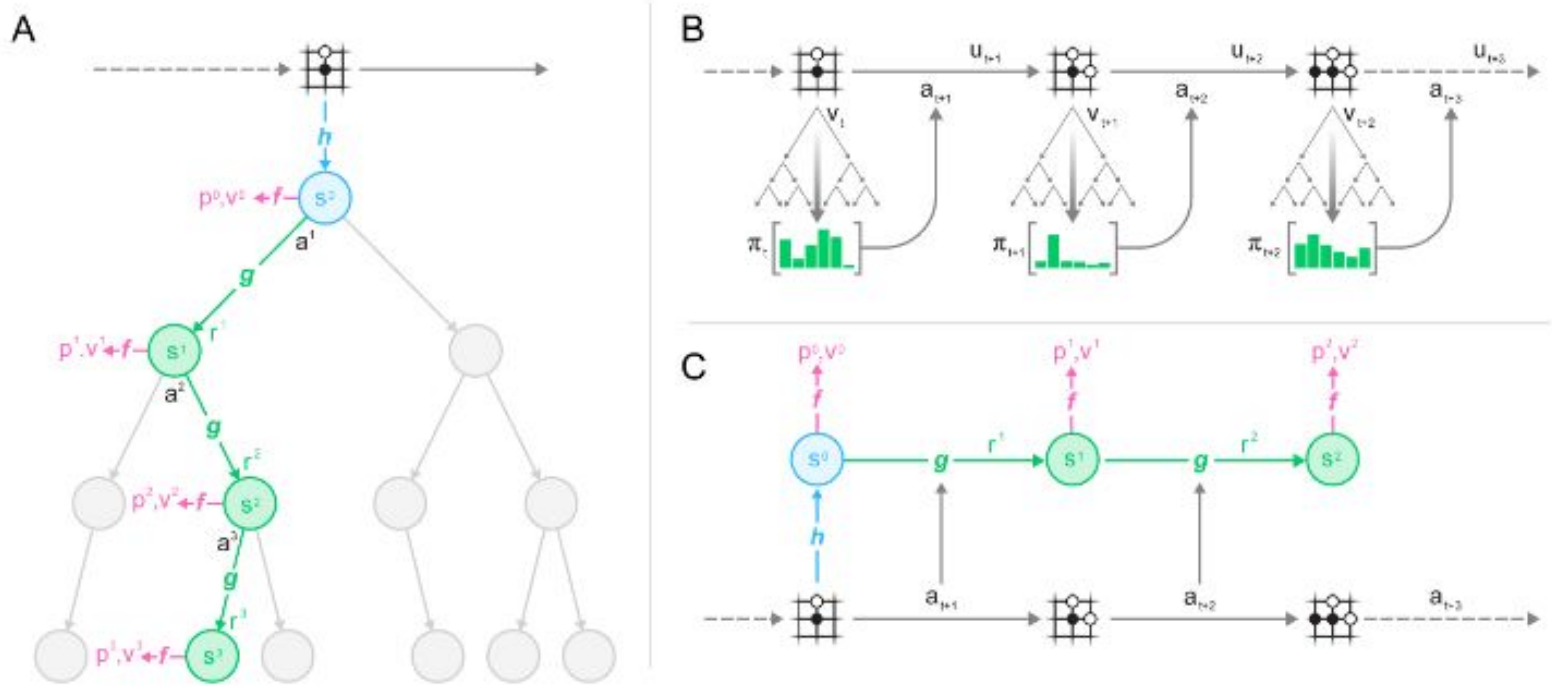


Silver, David, et al. 'A General Reinforcement Learning Algorithm That Masters Chess, Shogi, and Go through Self-Play'. Science, vol. 362, no. 6419, Dec. 2018, pp. 1140–44.

Leaving perfect information environments

representation function *h* | prediction function *f* | dynamics function *g*

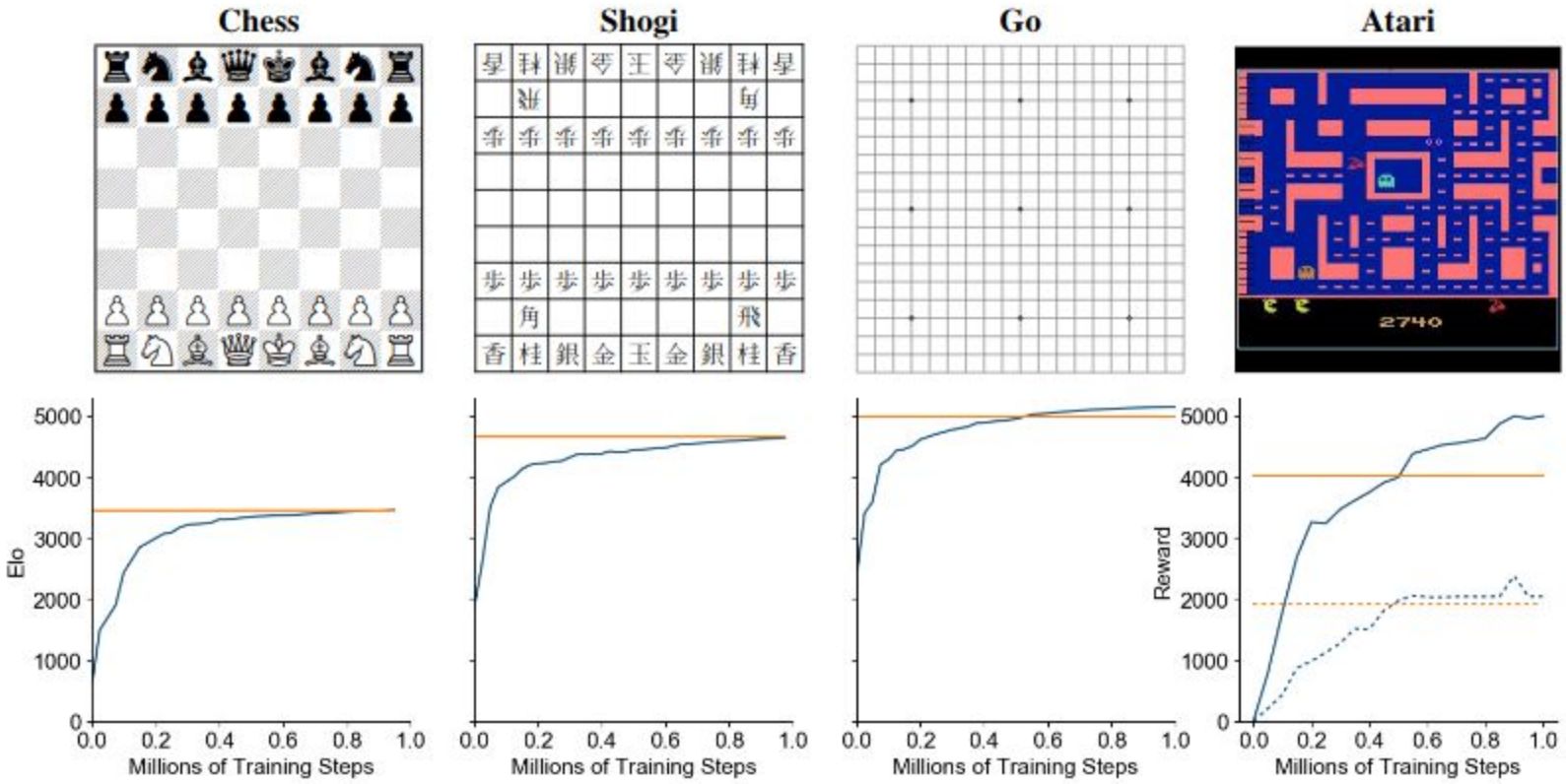
A: planning
B: acting
C: training



Schrittwieser, Julian, et al. ‘Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model’. ArXiv:1911.08265 [Cs, Stat], Nov. 2019. arXiv.org, http://arxiv.org/abs/1911.08265.

Leaving perfect information environments

learns all game rules on its own

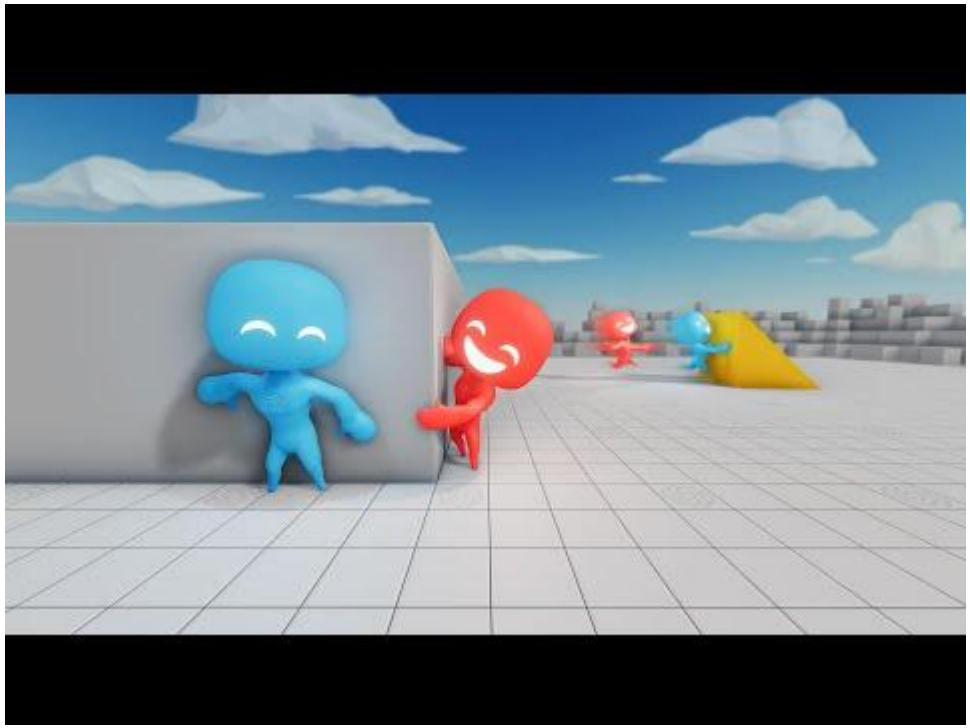


Compared against: Stockfish (chess), Elmo (Shogi), AlphaZero (Go), R2D2 (Atari)

Schrittwieser, Julian, et al. ‘Mastering Atari, Go, Chess and Shogi by Planning with a Learned Model’. ArXiv:1911.08265 [Cs, Stat], Nov. 2019. arXiv.org, http://arxiv.org/abs/1911.08265.

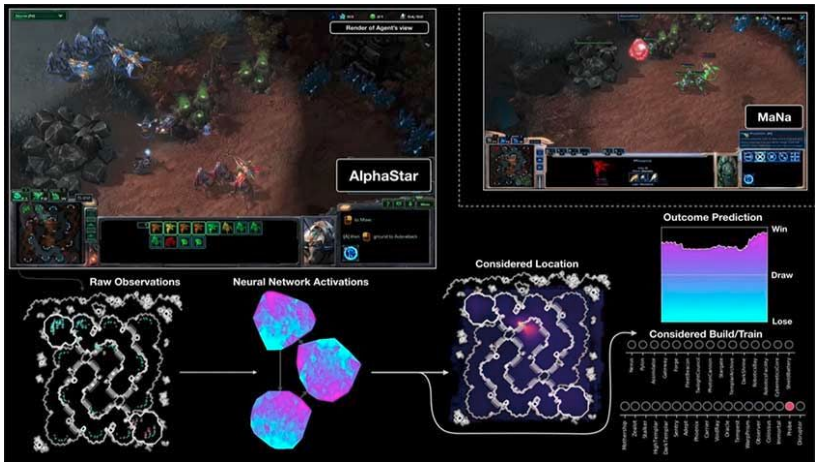
Some other advances

Hide and Seek



Multiple agents in an open environment

AlphaStar



approx. values	**Chess**	**Go**	**Starcraft II**
breadth	35	250	10^{26}
depth	80	150	1000s

Thank you for your attention!

Any questions?