Reinforcment Learning for Sequential Decision Making

Peter Stone*

Director, Learning Agents Research Group (LARG)
Department of Computer Science
The University of Texas at Austin

Joint work with members of LARG
past and present

Also, Cogitai Inc.

Who Am I?

Who Am I?

To what degree can autonomous intelligent agents learn in the presence of teammates and/or adversaries in real-time, dynamic domains?

Who Am I?

To what degree can autonomous intelligent agents learn in the presence of teammates and/or adversaries in real-time, dynamic domains?

Research Areas

- Autonomous agents
- Robotics
- Multiagent systems

Who Am I?

To what degree can autonomous intelligent agents learn in the presence of teammates and/or adversaries in real-time, dynamic domains?

Research Areas

- Autonomous agents
- Robotics
- Multiagent systems
- Machine learning

Who Am I?

To what degree can autonomous intelligent agents learn in the presence of teammates and/or adversaries in real-time, dynamic domains?

Research Areas

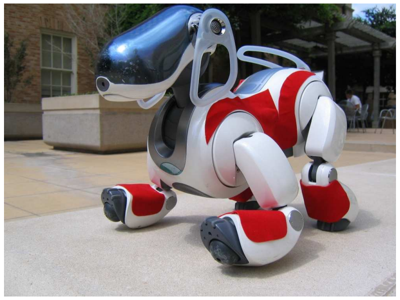
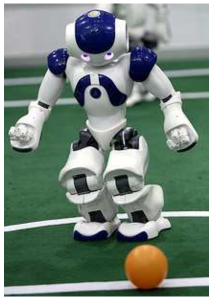
- Autonomous agents
- Robotics
- Multiagent systems
- Machine learning
  - **Reinforcement Learning**

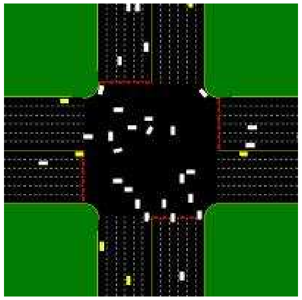
Who Am I?

To what degree can autonomous intelligent agents learn in the presence of teammates and/or adversaries in real-time, dynamic domains?

Research Areas

- Autonomous agents
- Robotics
- Multiagent systems
- Machine learning
  - **Reinforcement Learning**



AlphaGo [Silver et al., '15]: An AI milestone

AlphaGo [Silver et al., ’15]: An AI milestone

- Integrated two **existing** AI technologies

AlphaGo [Silver et al., ’15]: An AI milestone

- Integrated two **existing** AI technologies
  - Supervised learning (Deep learning)

AlphaGo [Silver et al., '15]: An AI milestone

- Integrated two **existing** AI technologies
  - Supervised learning (Deep learning)
  - **Reinforcement learning**

AlphaGo [Silver et al., '15]: An AI milestone

- Integrated two **existing** AI technologies
  - Supervised learning (Deep learning)
  - **Reinforcement learning**
- Does **not** solve AI

AlphaGo [Silver et al., ’15]: An AI milestone

- Integrated two **existing** AI technologies
  - Supervised learning (Deep learning)
  - **Reinforcement learning**
- Does **not** solve AI
  - The end of the line for:

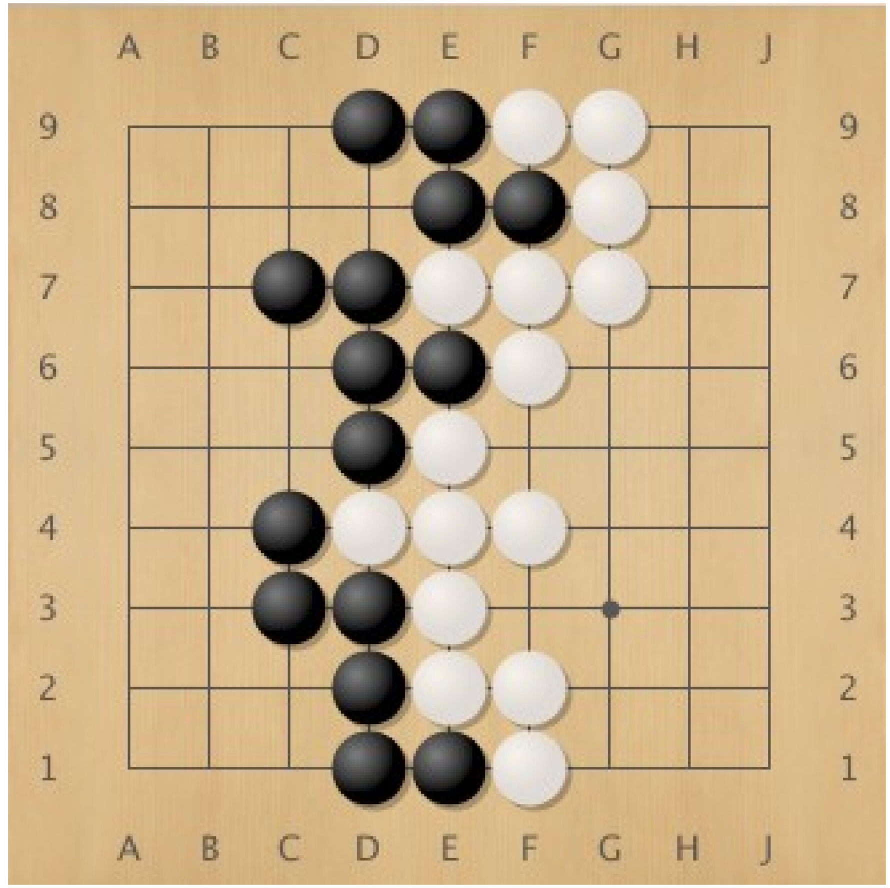
AlphaGo [Silver et al., '15]: An AI milestone

- Integrated two **existing** AI technologies
  - Supervised learning (Deep learning)
  - **Reinforcement learning**
- Does **not** solve AI
  - The end of the line for:

  2-player zero-sum discrete finite deterministic games of perfect information

A brief introduction to Go

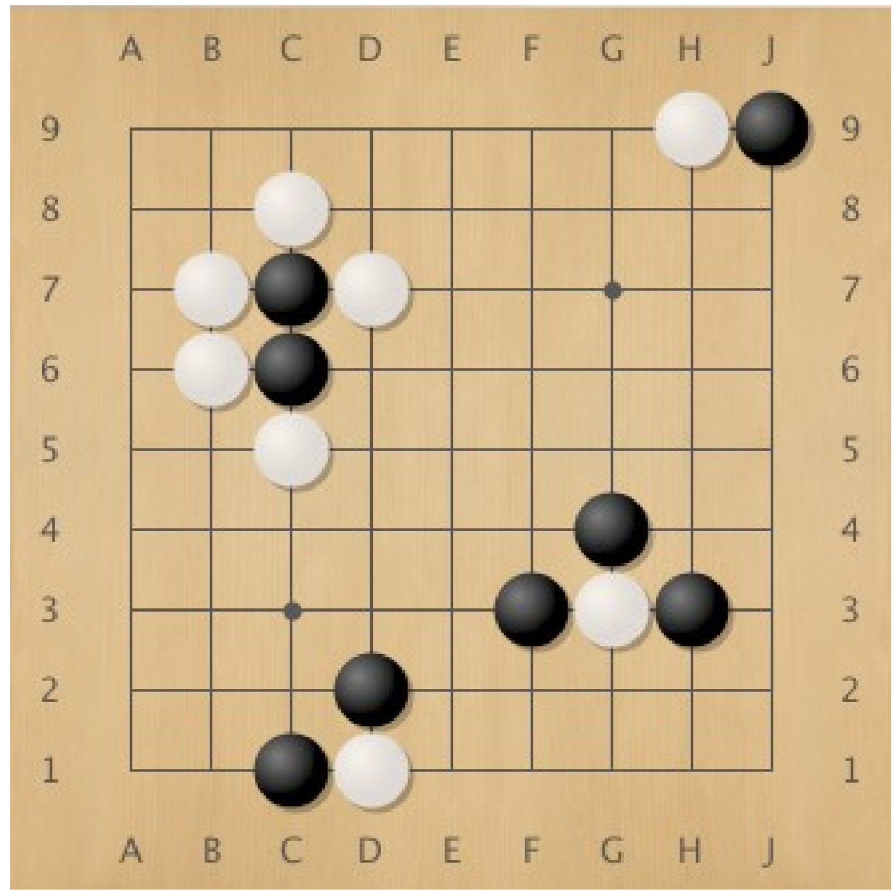
- Black and white take turns to place down stones
- Once played, a stone cannot move
- The aim is to surround the most territory
- Usually played on 19x19 board



Slide created by David Silver

Capturing

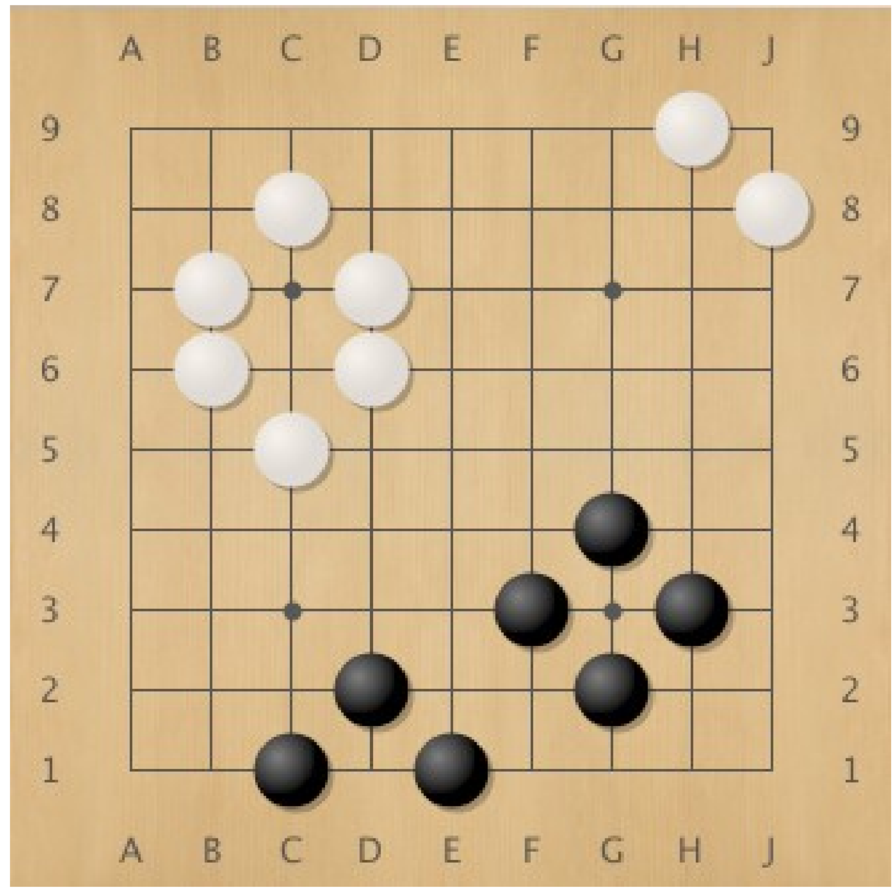
- The lines radiating from a stone are called *liberties*
- If a connected group of stones has all of its liberties removed then it is captured
- Captured stones are removed from the board



Slide created by David Silver

Capturing

- The lines radiating from a stone are called *liberties*
- If a connected group of stones has all of its liberties removed then it is captured
- Captured stones are removed from the board



Slide created by David Silver

2-player zero-sum discrete finite deterministic games of perfect information

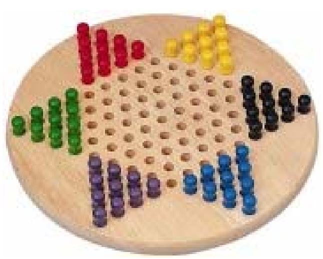
What do these terms mean?

- Two player: Duh!
- Zero-sum: In any outcome of any game, Player A's gains equal player B's losses. (Doesn't mean fairness: "On average, two equal players will win or lose equal amounts" not necessary for zero-sum.)
- Discrete: All game states and decisions are discrete values.
- Finite: Only a finite number of states and decisions.
- Deterministic: No chance (no die rolls).
- Games: See next page
- Perfect information: Both players can see the state, and each decision is made sequentially (no simultaneous moves).

Slide created by Andrew Moore

Which of these are: 2-player zero-sum discrete finite deterministic games of perfect information

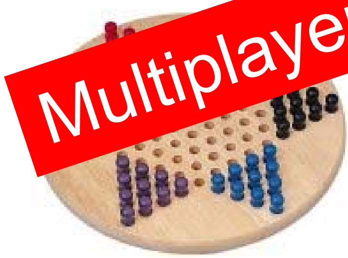
- Two player: Duh!
- Zero-sum: In any outcome of any game, Player A's gains equal player B's losses.
- Discrete: All game states and decisions are discrete values.
- Finite: Only a finite number of states and decisions.
- Deterministic: No chance (no die rolls).
- Games: See next page
- Perfect information: Both players can see the state, and each decision is made sequentially (no simultaneous moves).



Slide created by Andrew Moore

Which of these are: 2-player zero-sum discrete finite deterministic games of perfect information

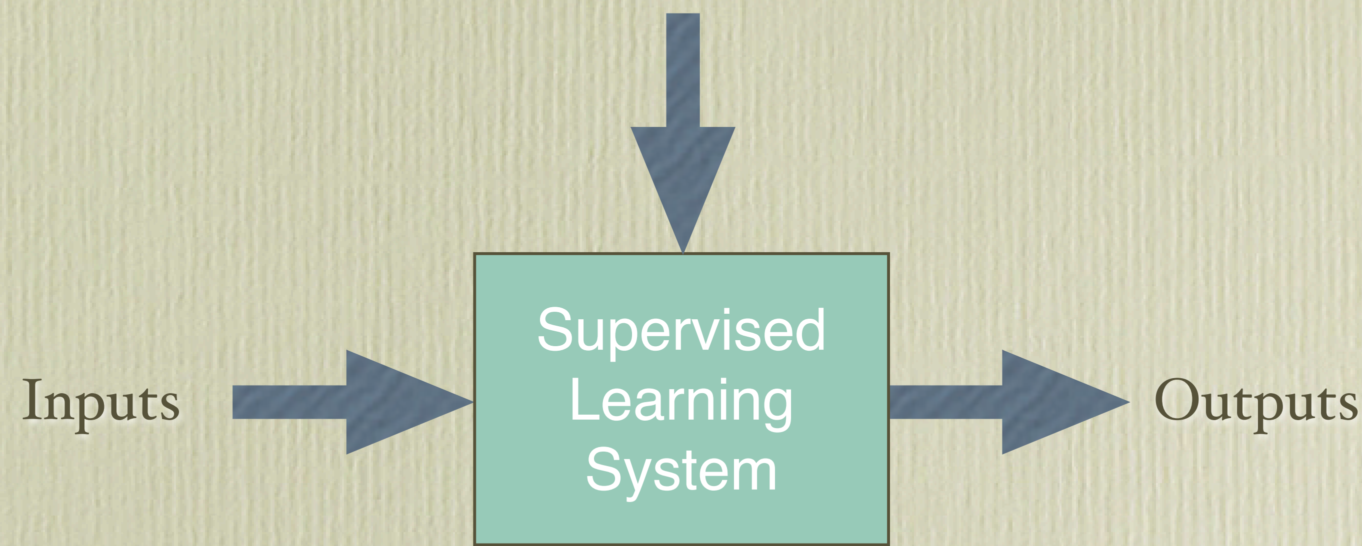
- Two player: Duh!
- Zero-sum: In any outcome of any game, Player A's gains equal player B's losses.
- Discrete: All game states and decisions are discrete values.
- Finite: Only a finite number of states and decisions.
- Deterministic: No chance (no die rolls).
- Games: See next page
- Perfect information: Both players can see the state, and each decision is made sequentially (no simultaneous moves).



Slide created by Andrew Moore

Supervised Learning

Training Info = desired (target) outputs

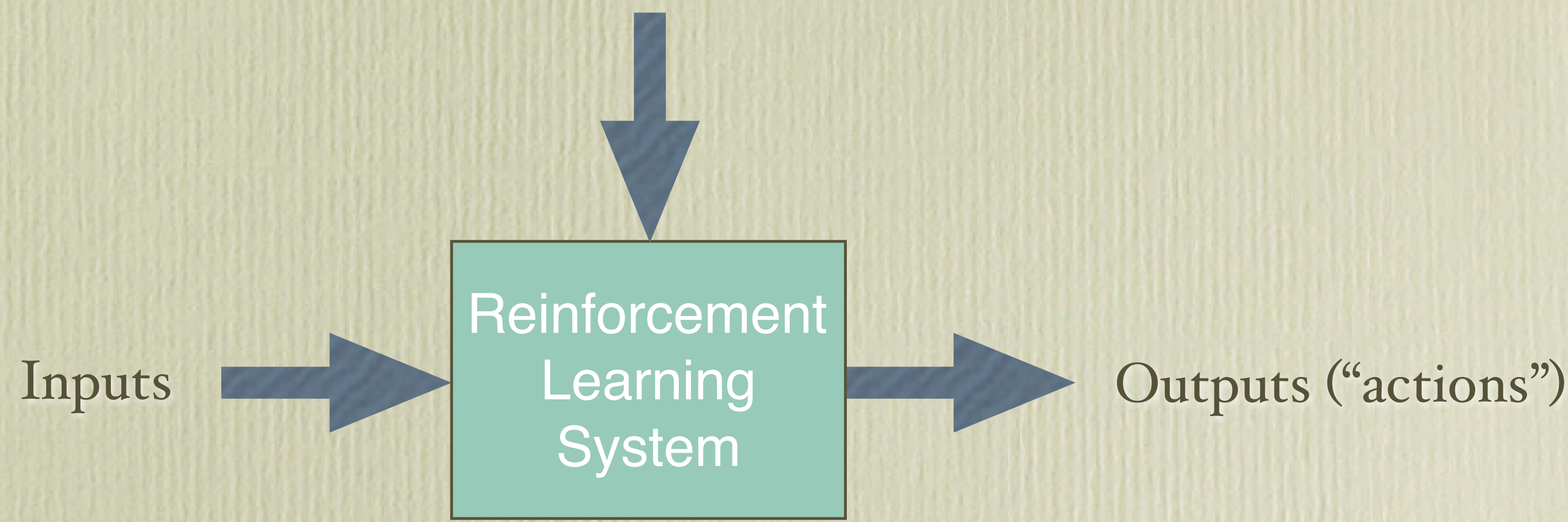


Error = (target output − actual output)

Slide created by Andrew Barto

Reinforcement Learning

Training Info = evaluations (“rewards” / “penalties”)



Objective: get as much reward as possible

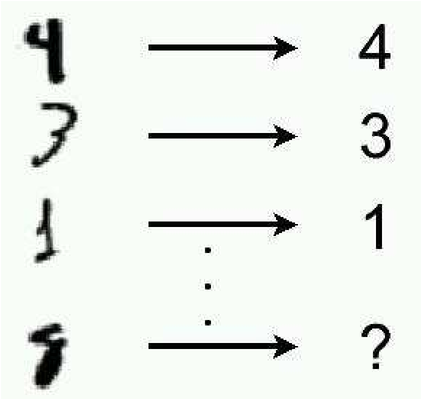
Slide created by Andrew Barto

Key Features of RL

- Learner is not told which actions to take
- Trial-and-Error search
- Possibility of delayed reward
  - Sacrifice short-term gains for greater long-term gains
- The need to ***explore*** and ***exploit***
- Considers the whole problem of a goal-directed agent interacting with an uncertain environment

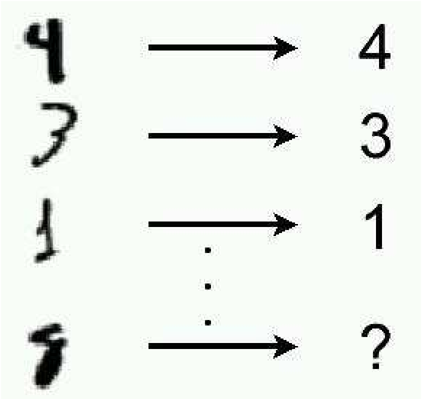
Slide created by Andrew Barto

Reinforcement Learning



Supervised learning mature [WEKA]

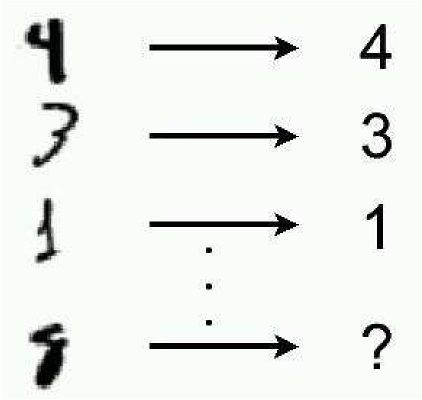
Reinforcement Learning



Supervised learning mature [WEKA]

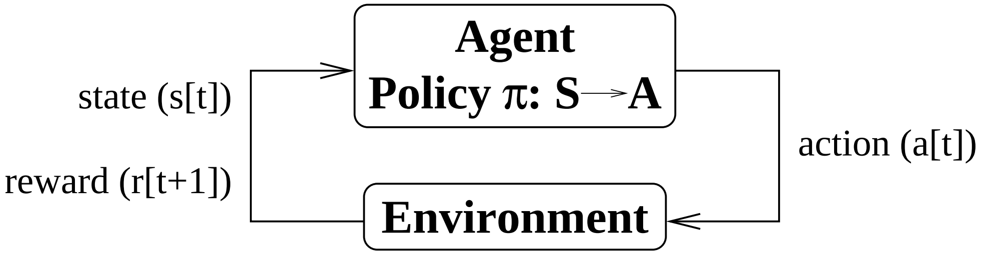
For agents, reinforcement learning most appropriate

Reinforcement Learning

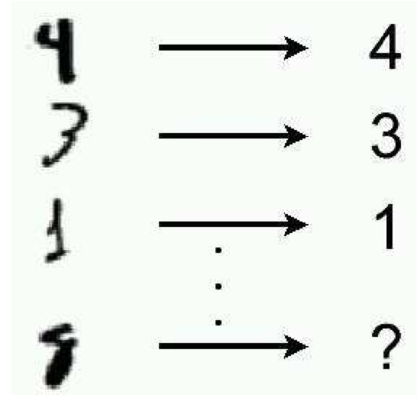


Supervised learning mature [WEKA]

For agents, reinforcement learning most appropriate

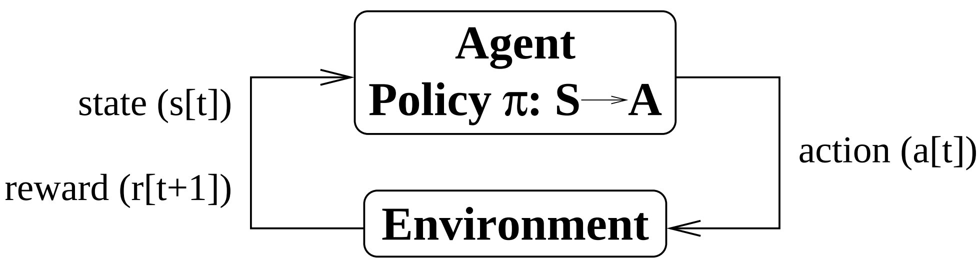


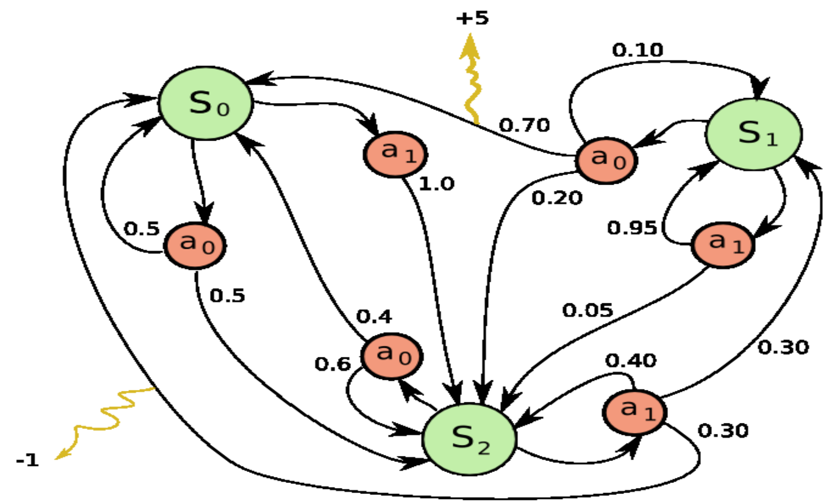
Reinforcement Learning



Supervised learning mature [WEKA]

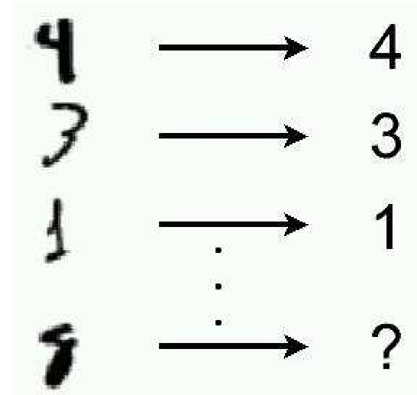
For agents, reinforcement learning most appropriate



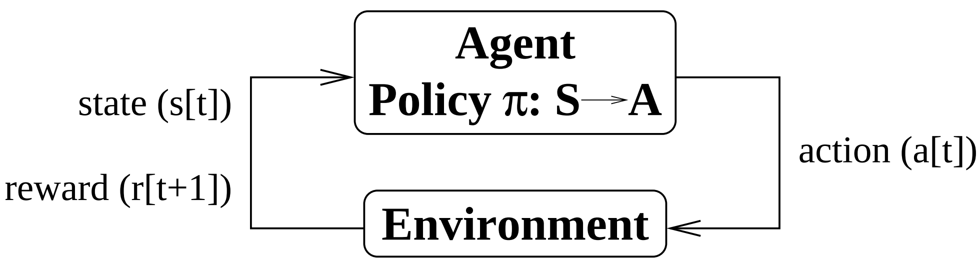


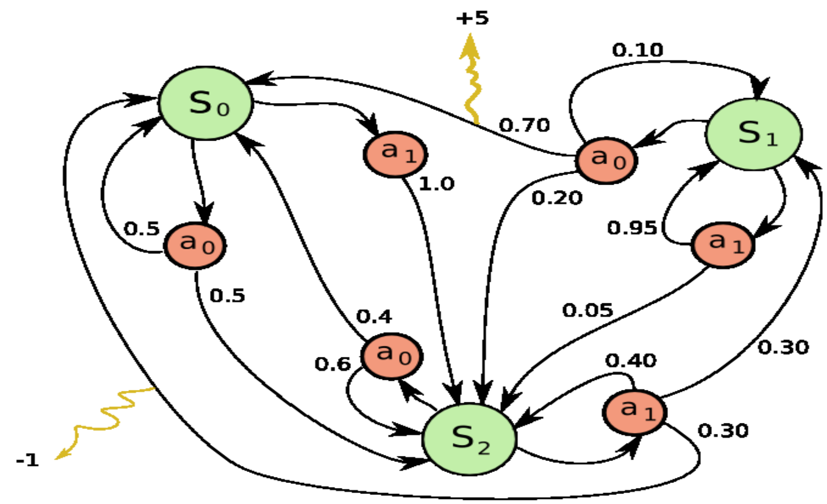
Reinforcement Learning

Supervised learning mature [WEKA]



For agents, reinforcement learning most appropriate

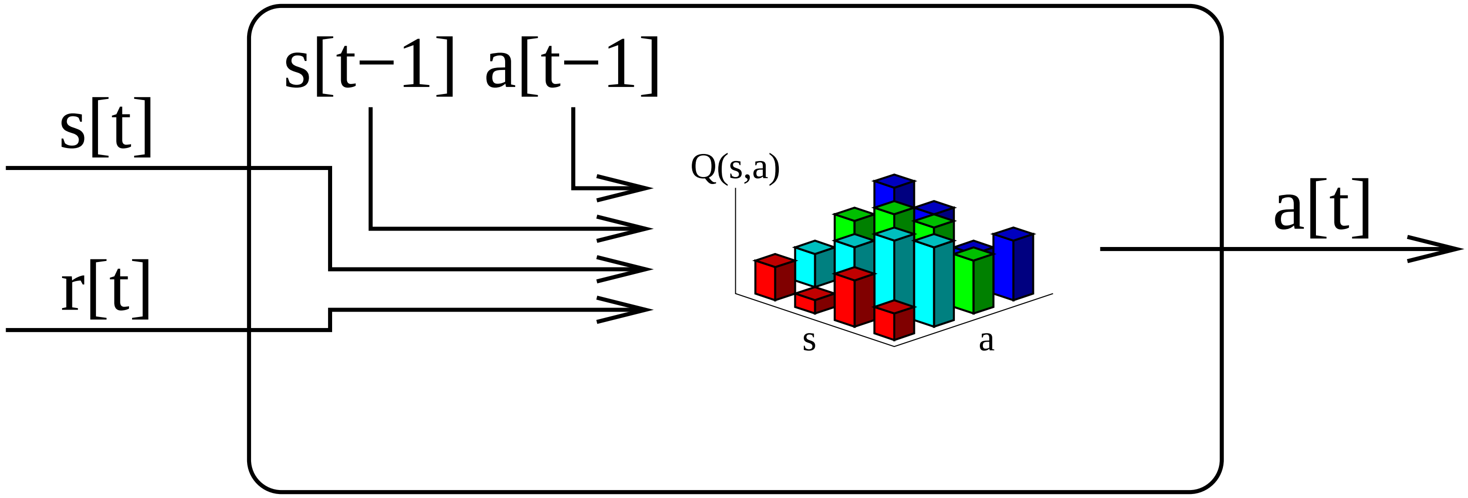




– Foundational theoretical results
– Applications require innovations to scale up

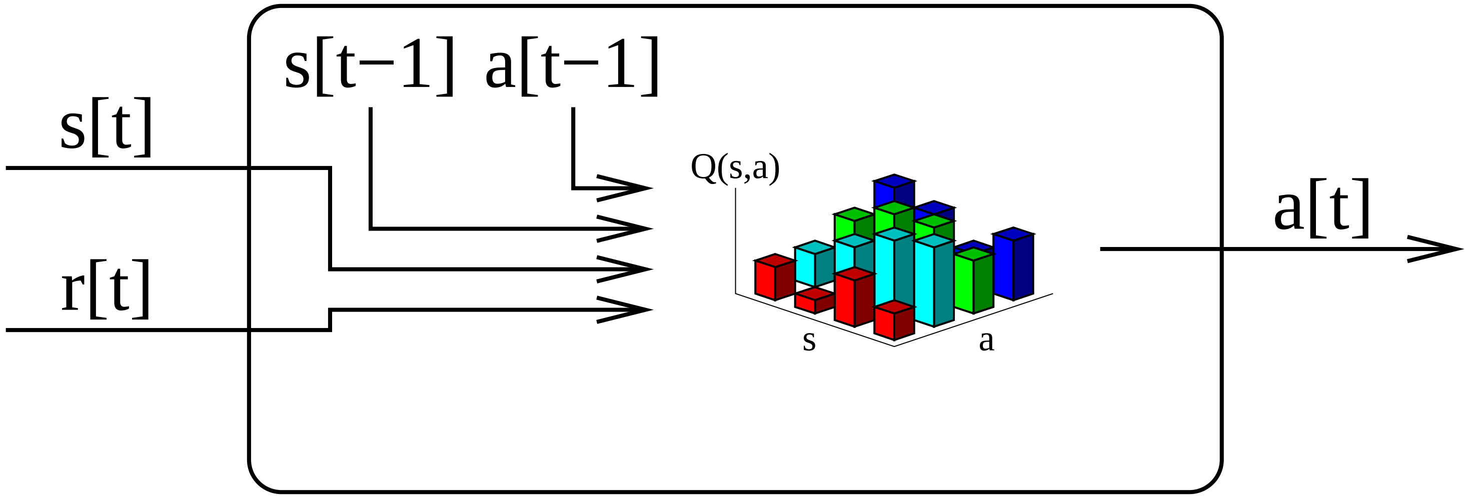
RL Theory

Success story: Q-learning converges to π^* [Watkins, 89]



RL Theory

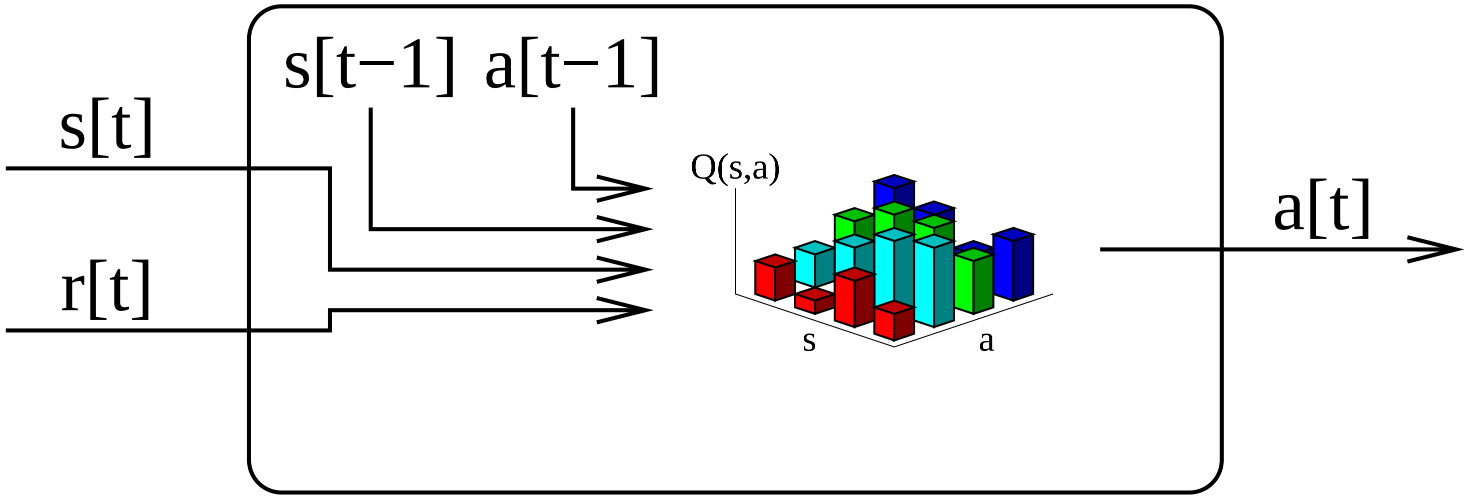
Success story: Q-learning converges to π^* [Watkins, 89]



– Table-based representation
– Visit every state infinitely often

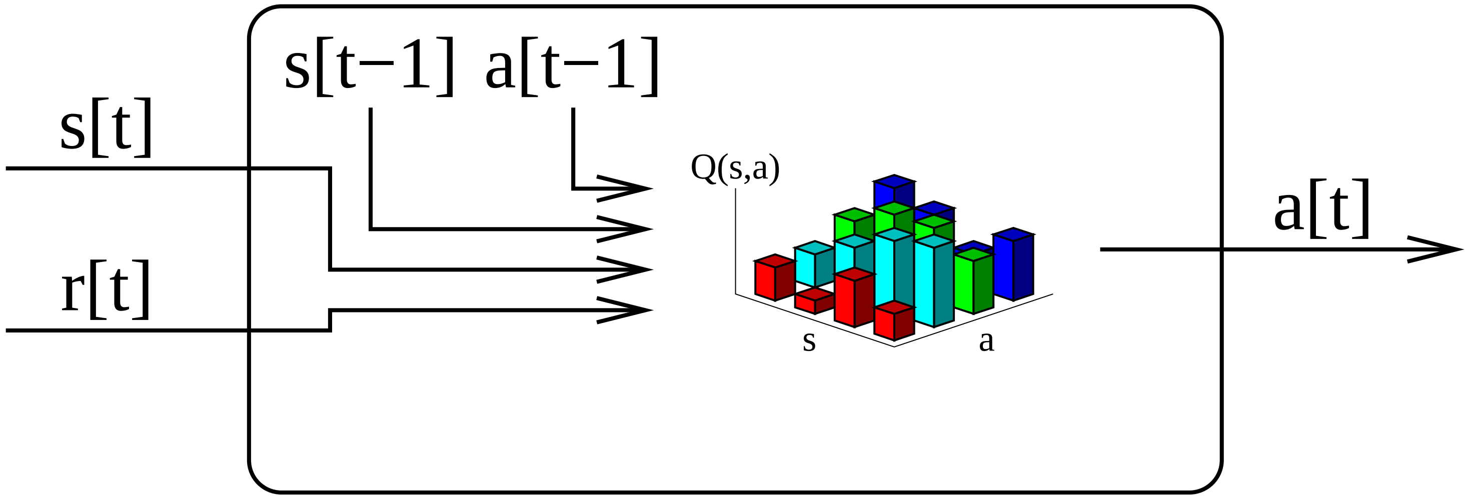
Function Approximation

In practice, visiting every state impossible

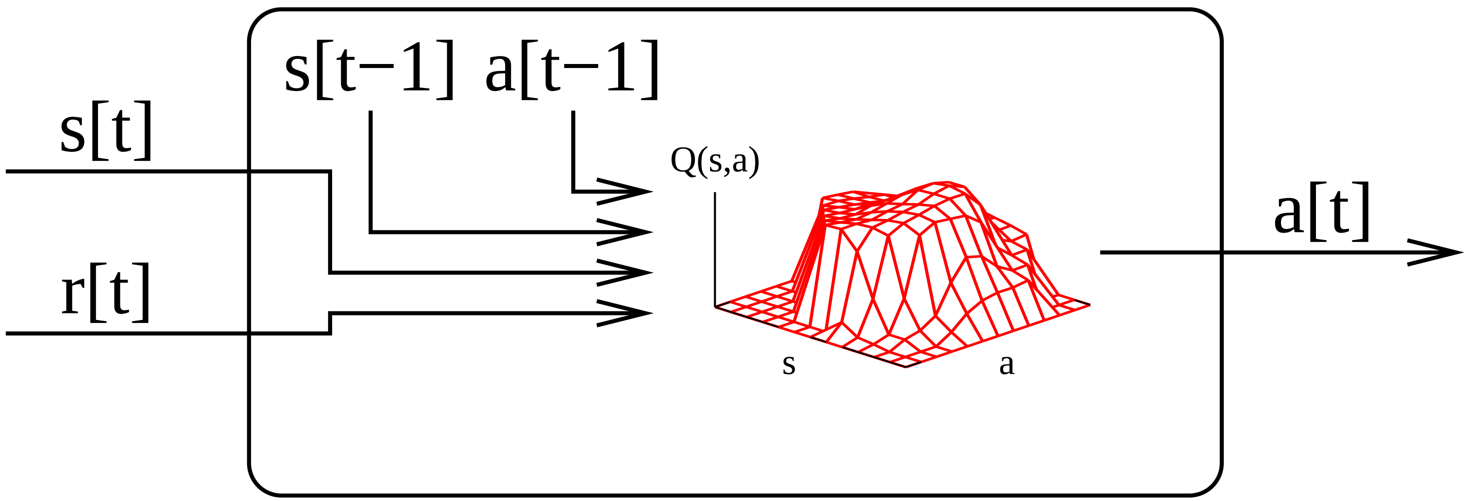


Function Approximation

In practice, visiting every state impossible



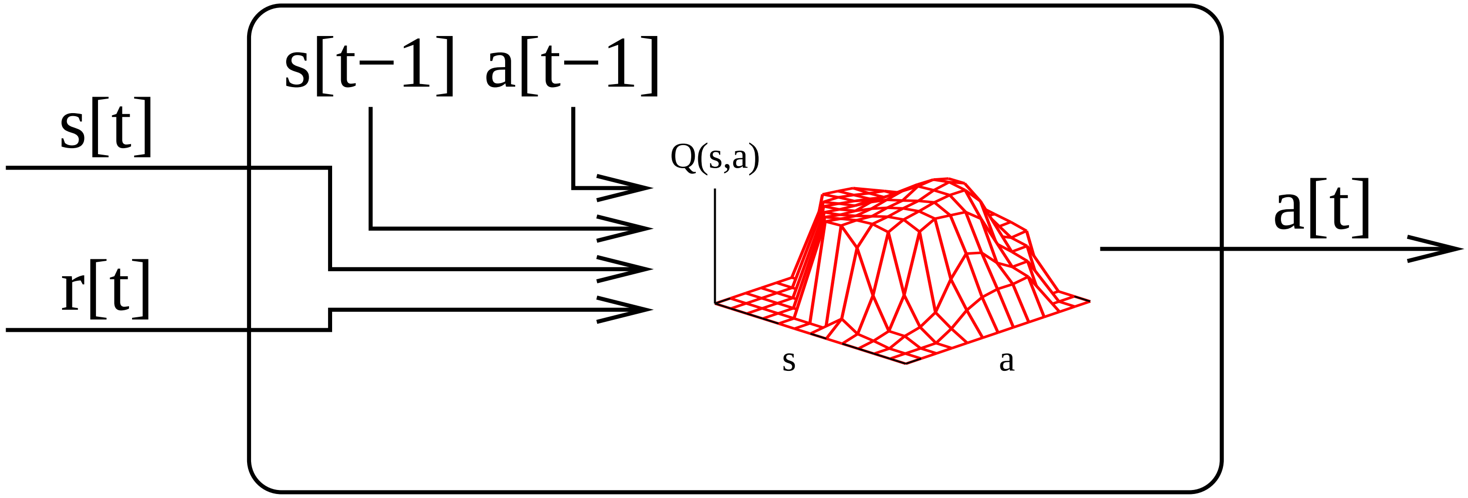
Function approximation of value function



Theoretical guarantees harder to come by

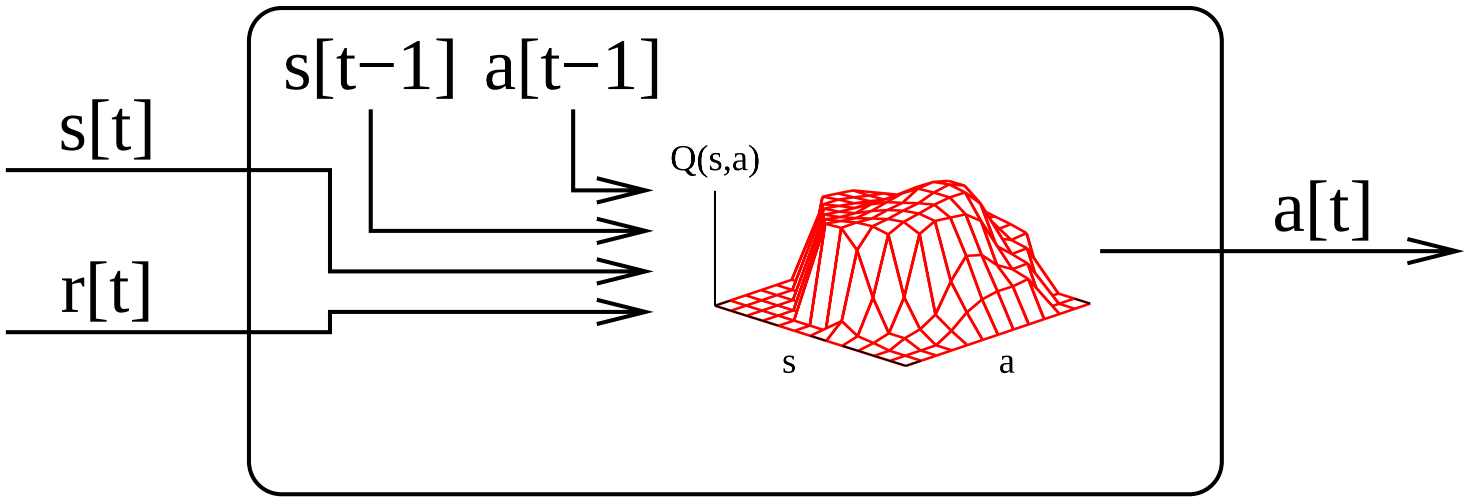
Batch Methods

In practice, often experience is scarce

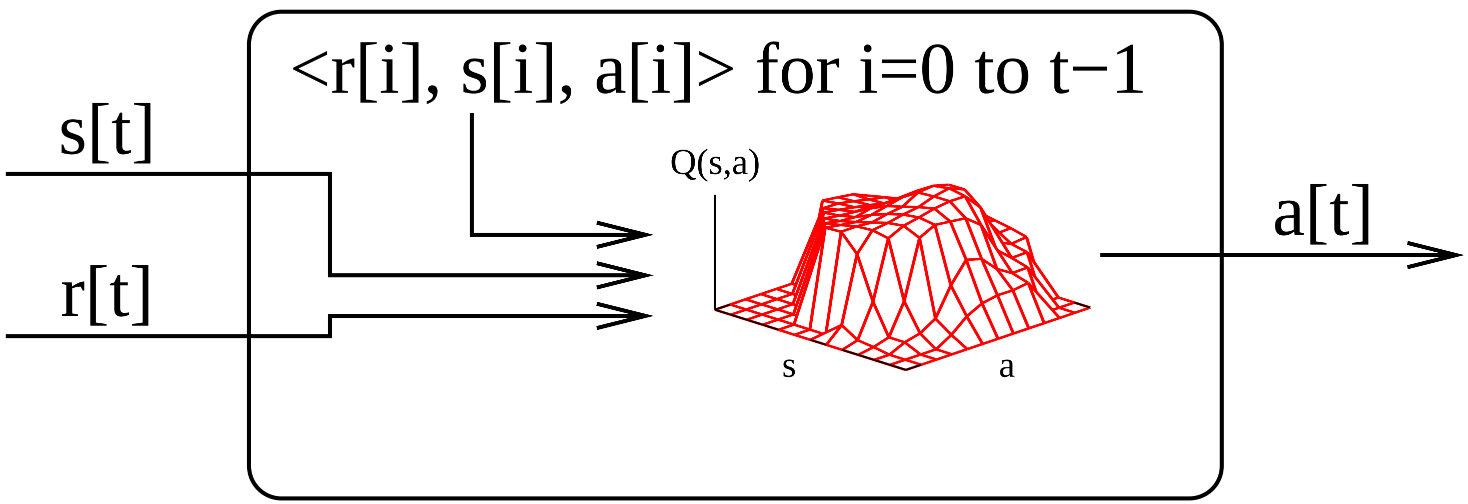


Batch Methods

In practice, often experience is scarce

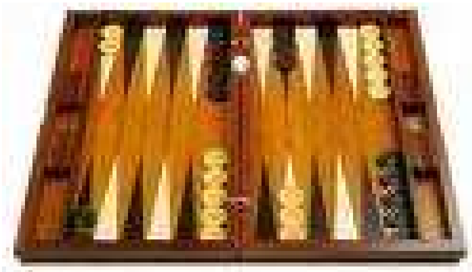


Save transitions:



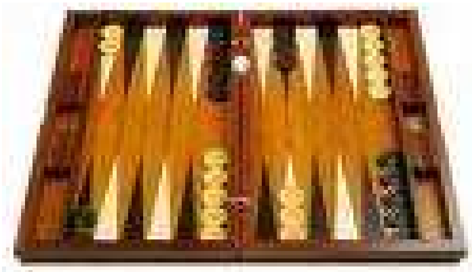
Applications: Towards a Useful Tool

- Backgammon [Tesauro, '94]
- Helicopter control [Ng et al., '03]

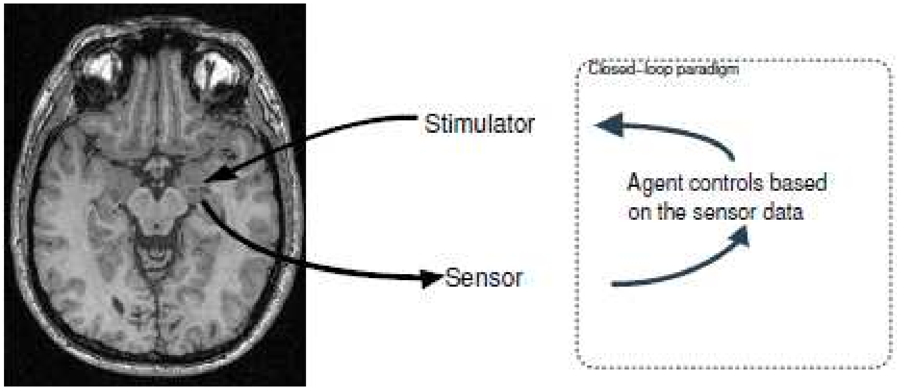


Applications: Towards a Useful Tool

- Backgammon [Tesauro, '94]
- Helicopter control [Ng et al., '03]



- Adaptive treatment of epilepsy [Pineau et al., '08]

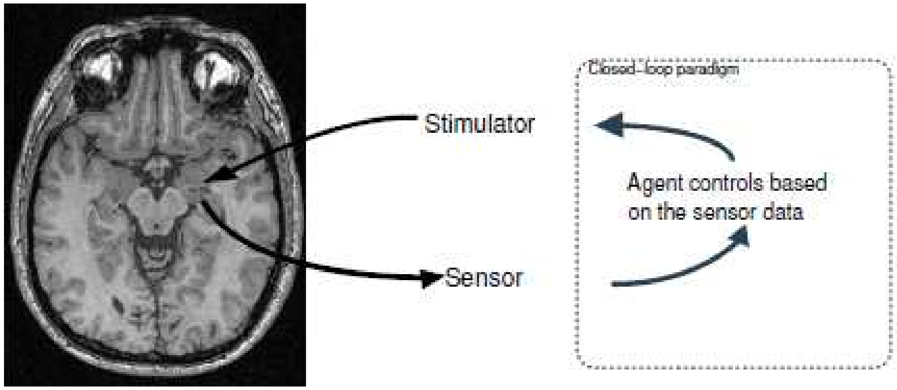


- Invasive species management, wildfire suppression [Dietterich et al., '13]

Applications: Towards a Useful Tool

- Backgammon [Tesauro, '94]
- Helicopter control [Ng et al., '03]

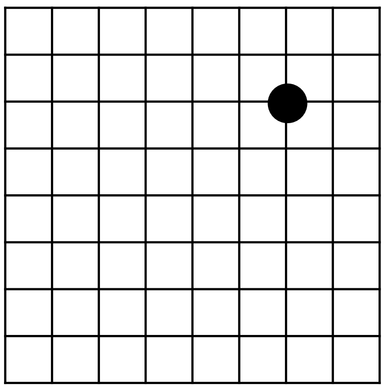
- Adaptive treatment of epilepsy [Pineau et al., '08]



- Invasive species management,
  wildfire suppression [Dietterich et al., '13]
- Google DeepMind beats human go champion, [Silver et al., '16]

Computer Go AI – Definition

d = 1



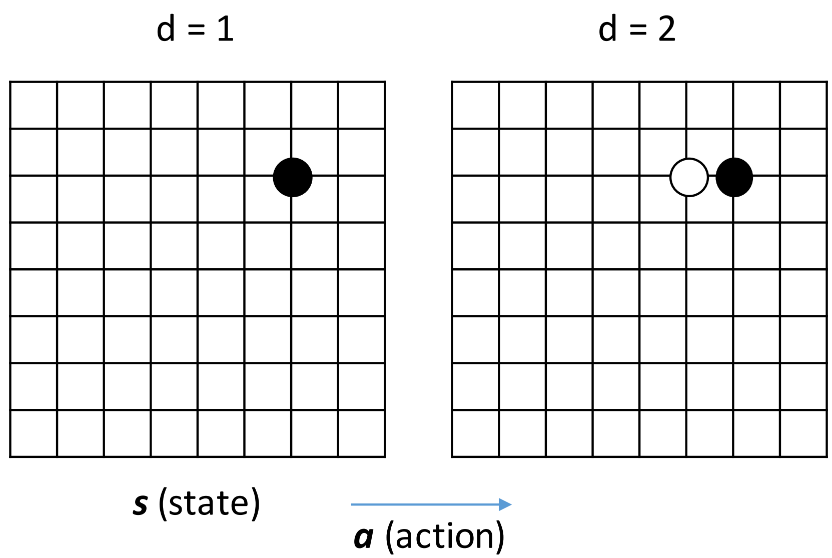
s (state)

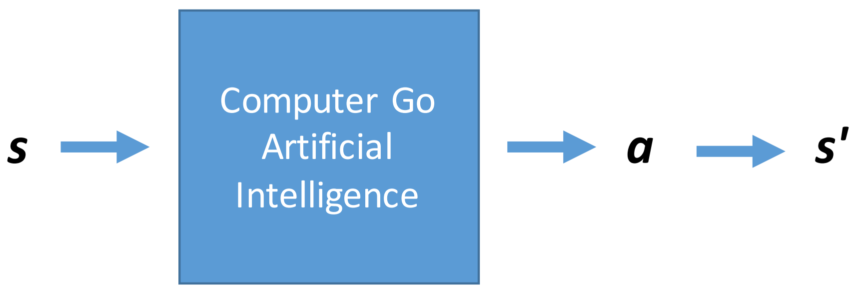
$$= \begin{pmatrix} 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&\color{red}{\mathbf{1}}&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \end{pmatrix}$$

(e.g. we can represent the board into a matrix-like form)

**** The actual model uses other features than board positions as well***

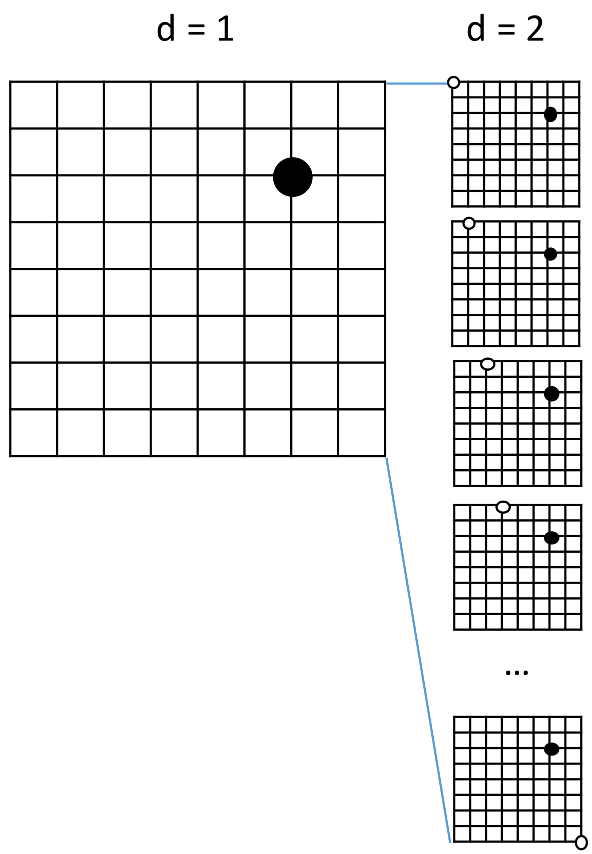
Computer Go AI – Definition





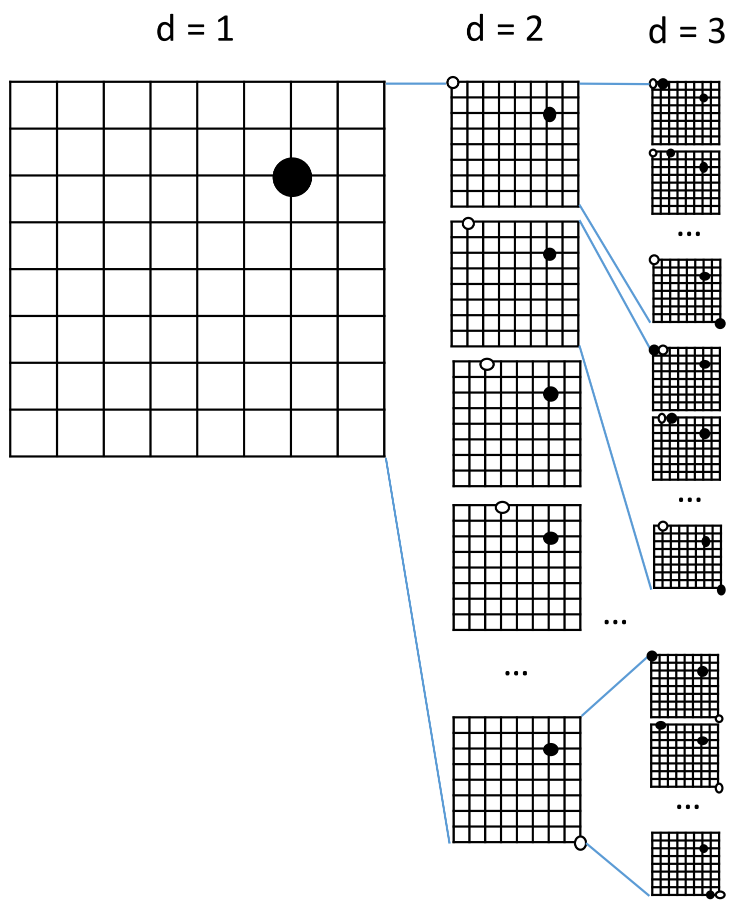
Given ***s***, pick the best ***a***

Computer Go AI – An Implementation Idea?

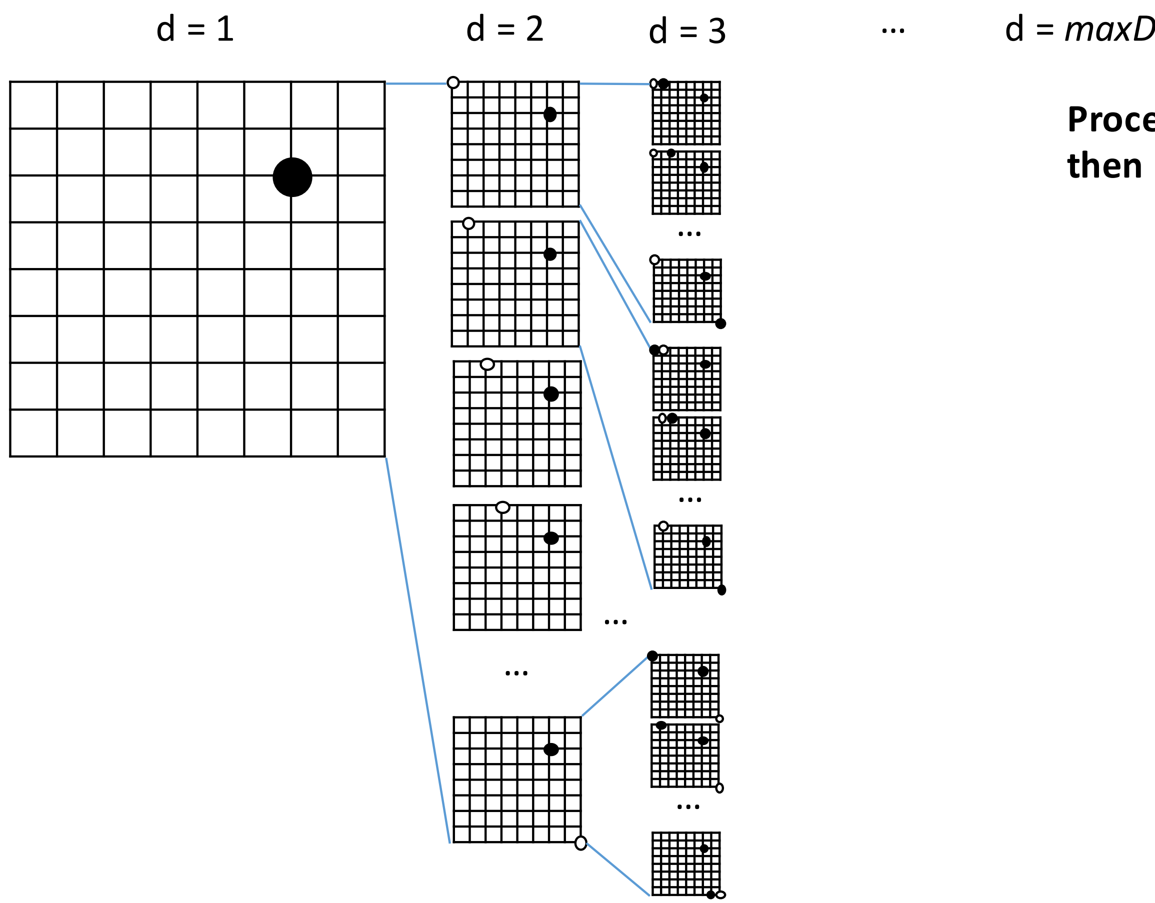


How about simulating all possible board positions?

Computer Go AI – An Implementation Idea?

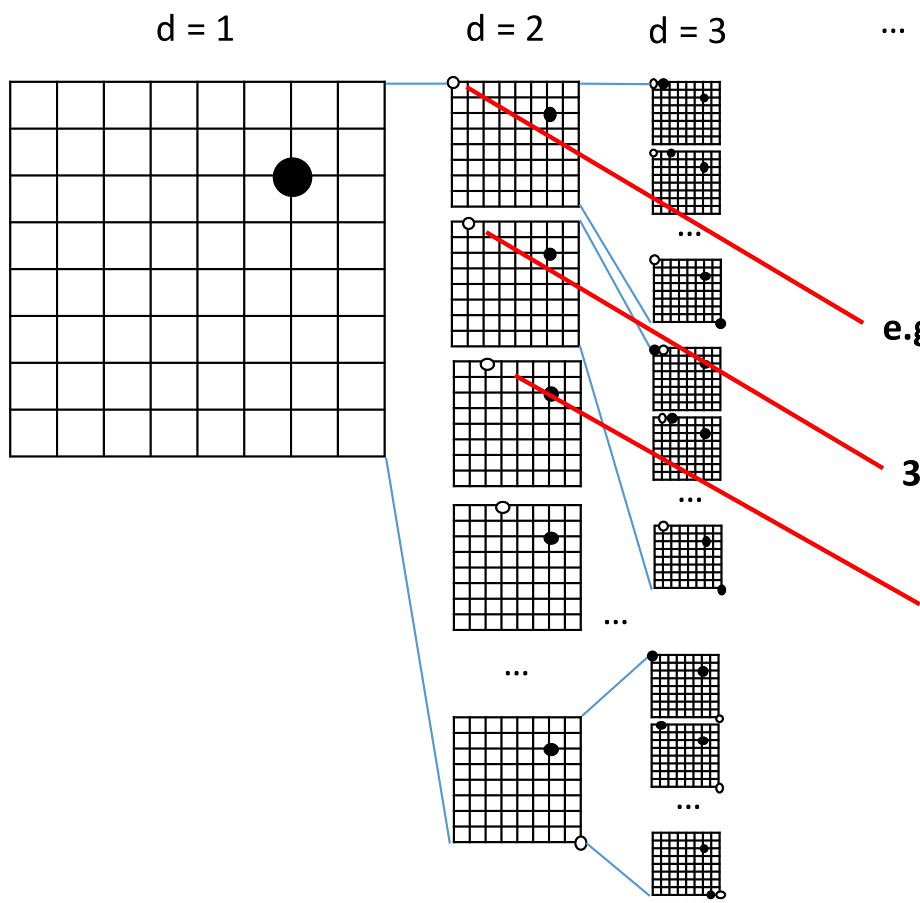


Computer Go AI – An Implementation Idea?



Process the simulation until the game ends, then report win / lose results

Computer Go AI – An Implementation Idea?



Process the simulation until the game ends, then report win / lose results

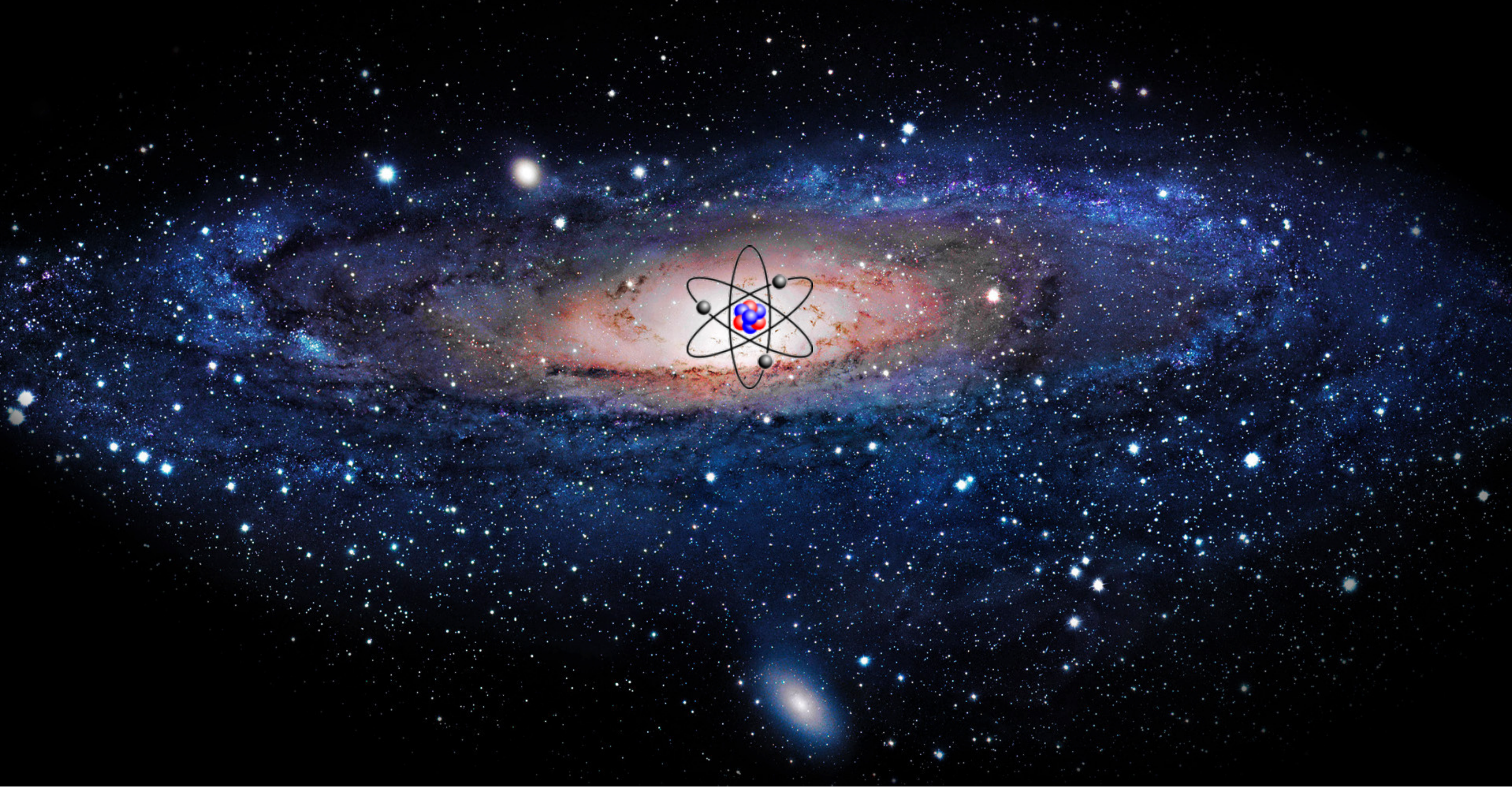
e.g. it wins 13 times if the next stone gets placed here

37,839 times

431,320 times

Choose the "next action / stone" that has the most win-counts in the full-scale simulation

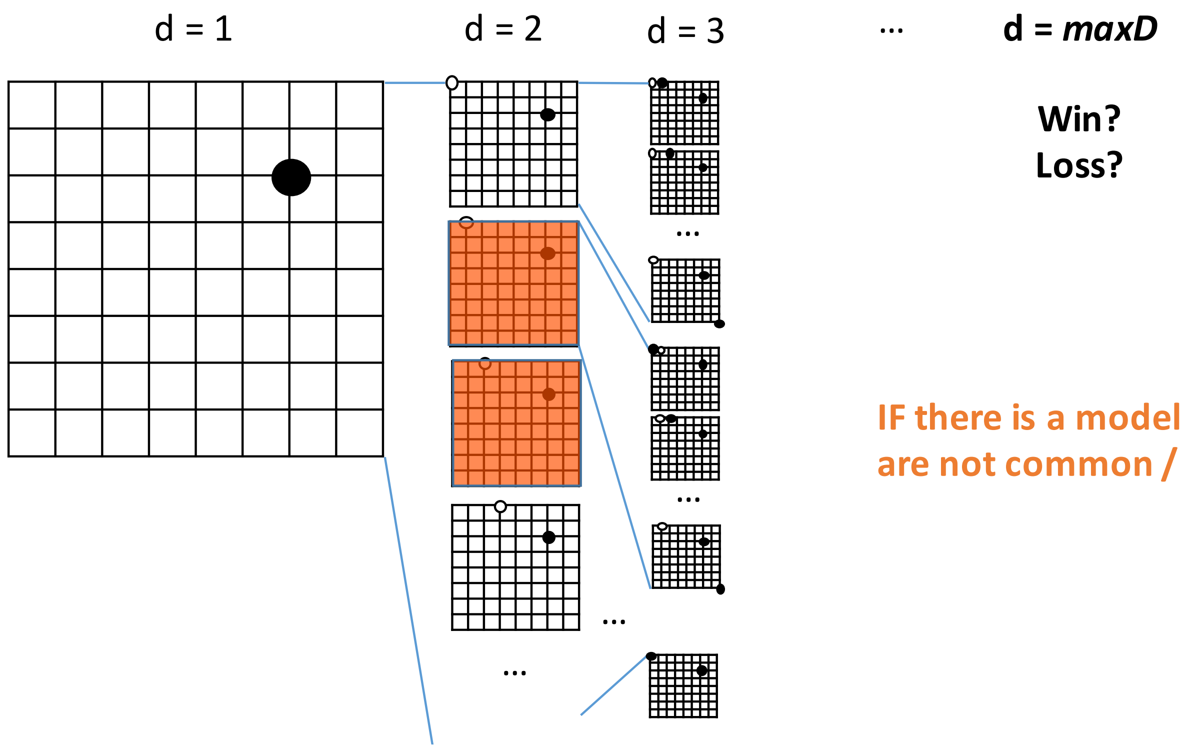
This is NOT possible; it is said the possible configurations of the board exceeds the number of atoms in the universe



Key: To Reduce Search Space

Reducing Search Space

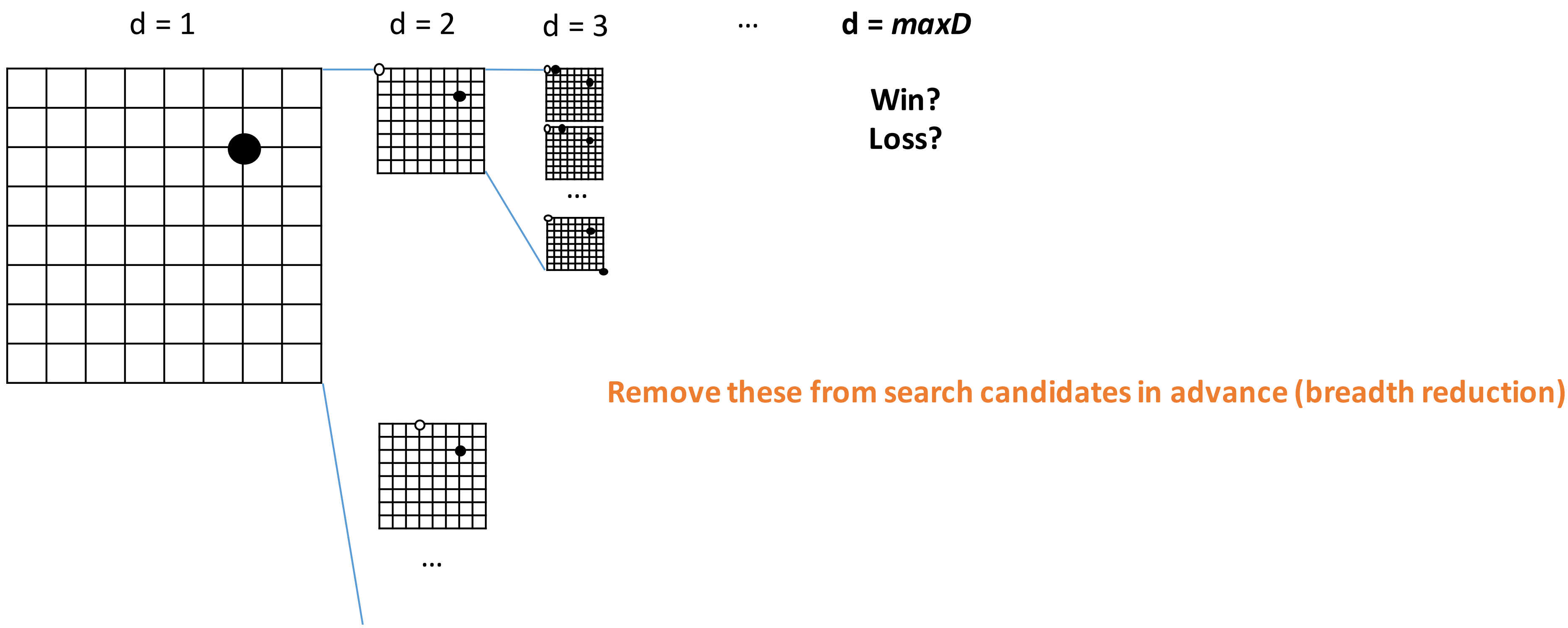
1. Reducing "action candidates" (Breadth Reduction)



IF there is a model that can tell you that these moves are not common / probable (e.g. by experts, etc.) …

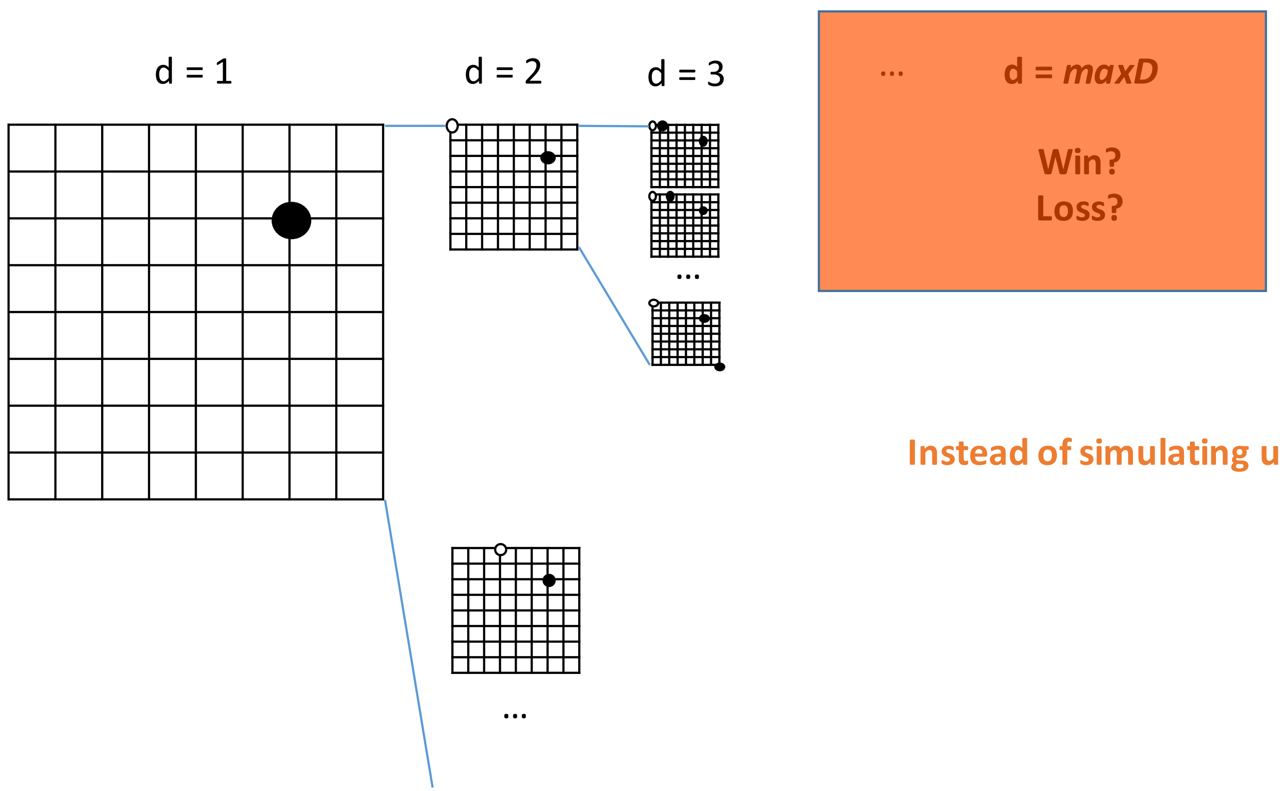
Reducing Search Space

1. Reducing "action candidates" (Breadth Reduction)



Reducing Search Space

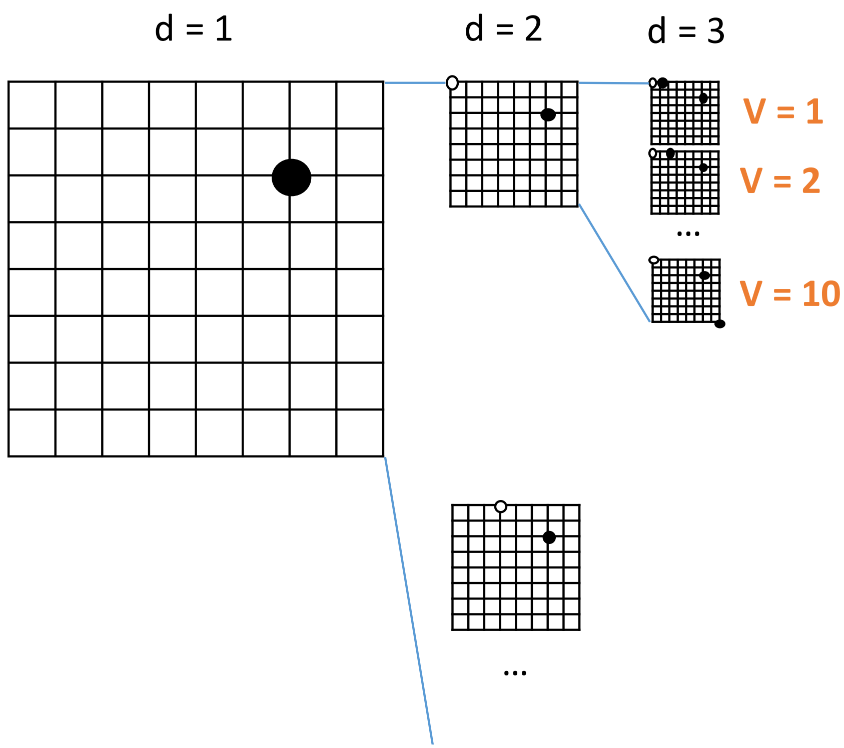
2. Position evaluation ahead of time (Depth Reduction)



Instead of simulating until the maximum depth ..

Reducing Search Space

2. Position evaluation ahead of time (Depth Reduction)



IF there is a function that can measure:
V(*s*): "board evaluation of state *s*"

Reducing Search Space

1. **Reducing “action candidates” (Breadth Reduction)**

2. **Position evaluation ahead of time (Depth Reduction)**

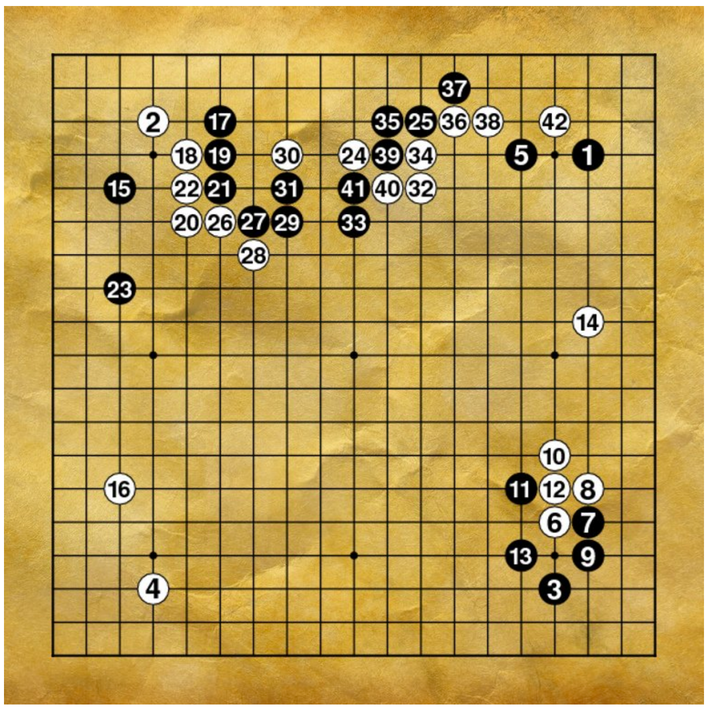
1. Reducing "action candidates"

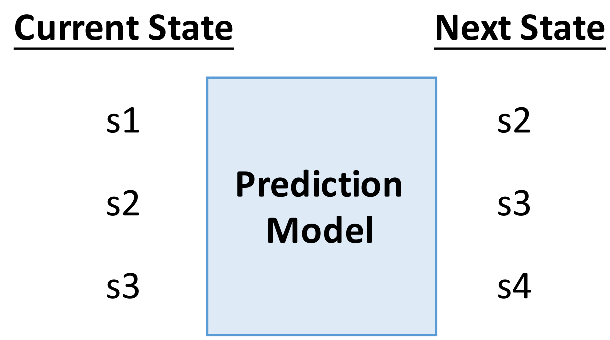
Learning: **P (next action | current state)**

= P (a | s)

1. Reducing "action candidates"

(1) Imitating expert moves (supervised learning)

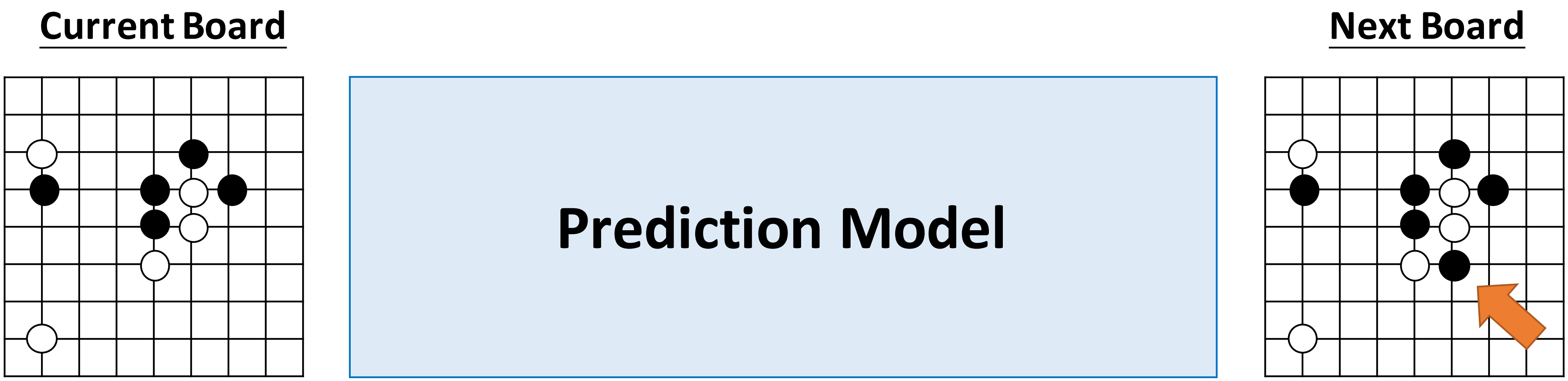




Data: Online Go experts (5~9 dan)
160K games, 30M board positions

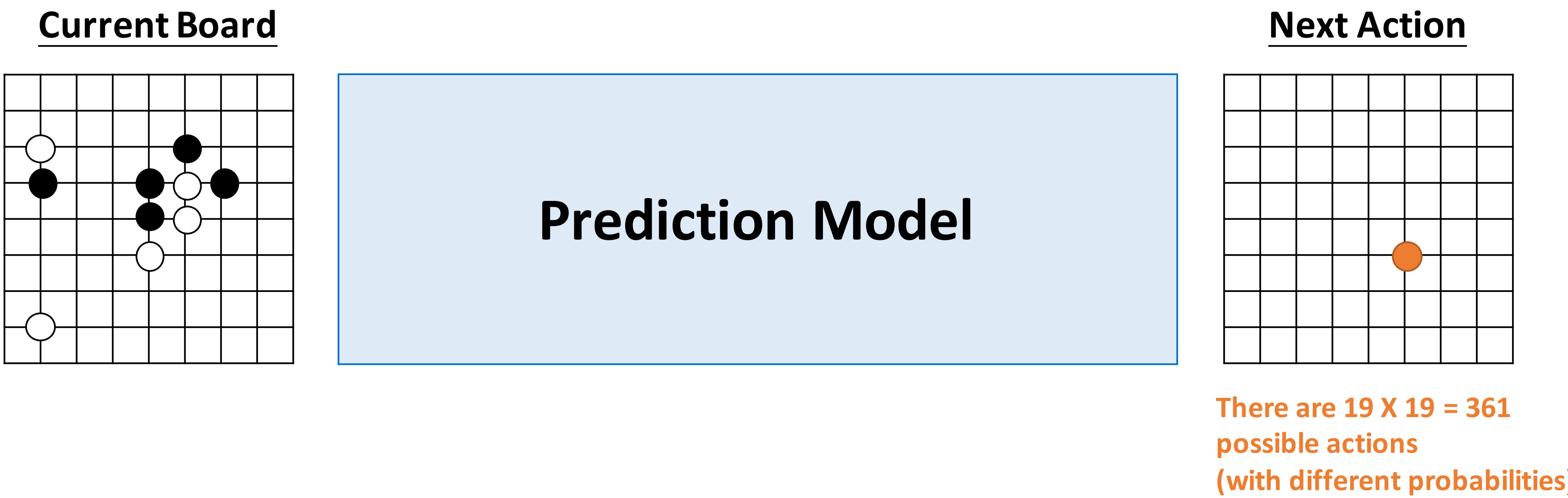
1. Reducing “action candidates”

(1) Imitating expert moves (supervised learning)



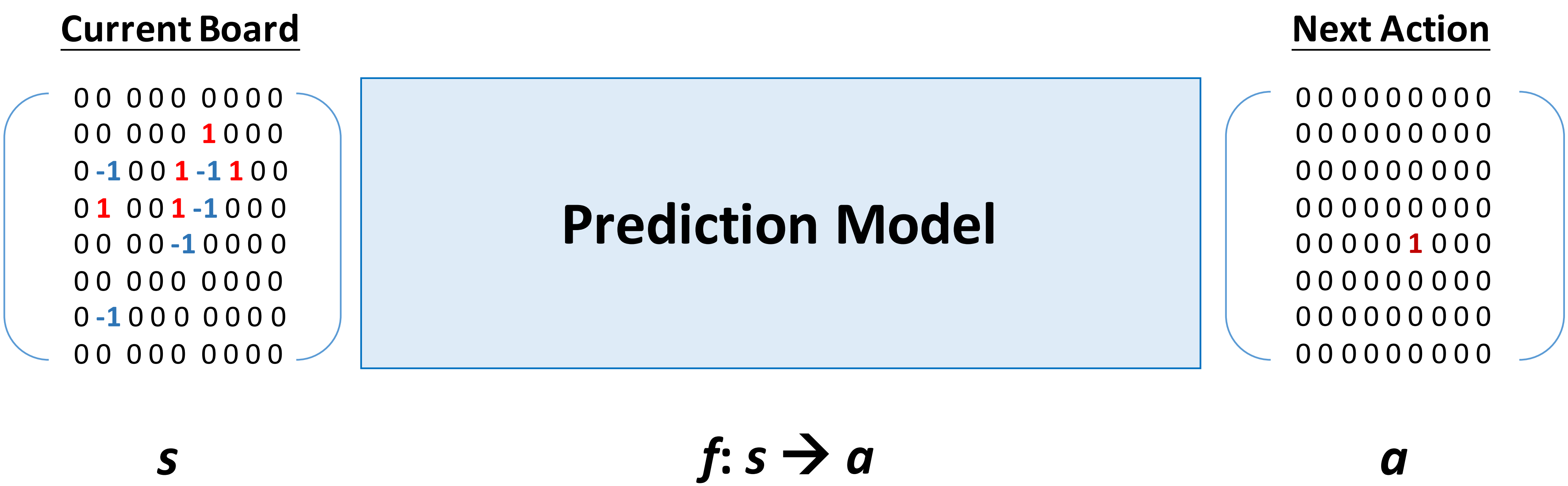
1. Reducing "action candidates"

(1) Imitating expert moves (supervised learning)



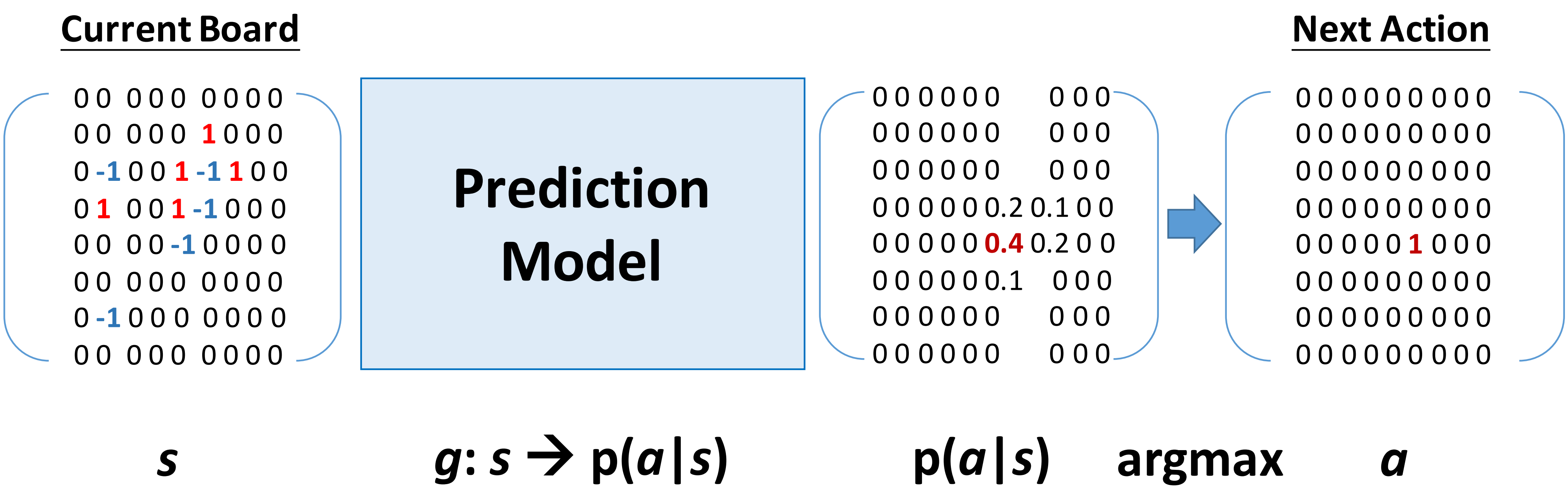
1. Reducing "action candidates"

(1) Imitating expert moves (supervised learning)



1. Reducing “action candidates”

(1) Imitating expert moves (supervised learning)



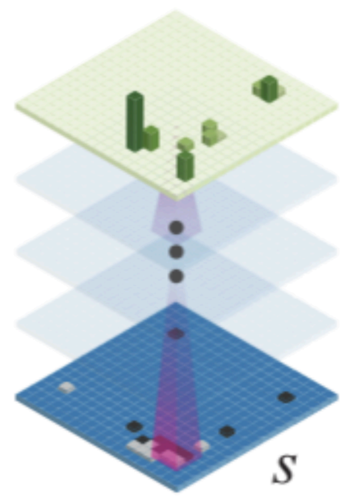
1. Reducing "action candidates"

(1) Imitating expert moves (supervised learning)

Current Board

$$\begin{pmatrix} 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0 \\ 0&-1&0&0&1&-1&1&0&0 \\ 0&1&0&0&1&-1&0&0&0 \\ 0&0&0&0&-1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&-1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \end{pmatrix}$$

Prediction Model



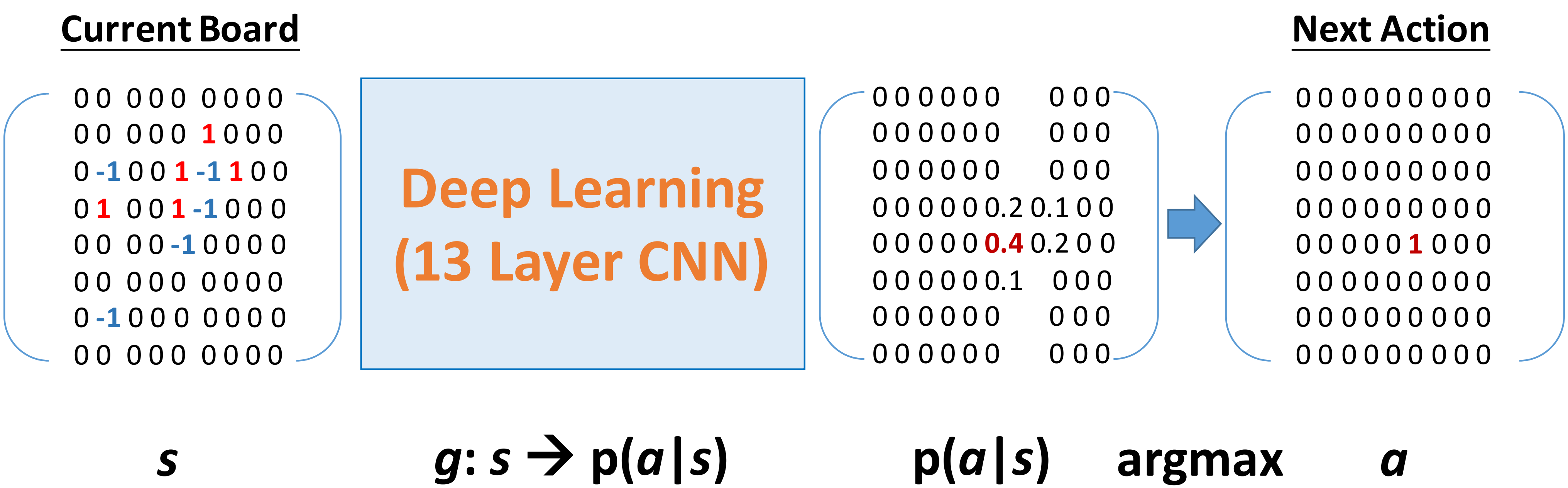
Next Action

$$\begin{pmatrix} 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \end{pmatrix}$$

s **$g: s \rightarrow$ p($a|s$)** **p($a|s$)** **argmax** a

1. Reducing “action candidates”

(1) Imitating expert moves (supervised learning)



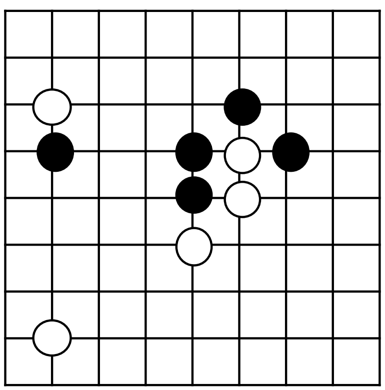
Go: abstraction is the key to win

CNN: abstraction is its *forte*

1. Reducing “action candidates”

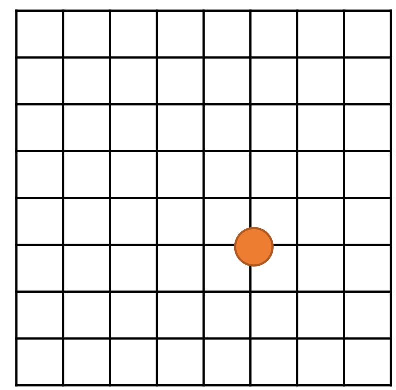
(1) Imitating expert moves (supervised learning)

Current Board



Expert Moves Imitator Model (w/ CNN)

Next Action



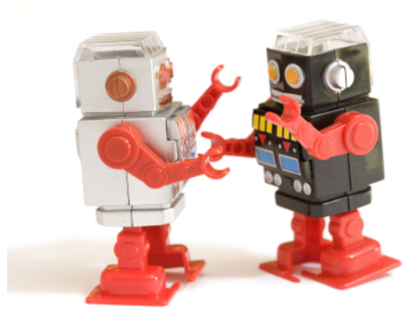
Training: $$\Delta\sigma \propto \frac{\partial \log p_\sigma(a|s)}{\partial \sigma}$$

1. Reducing “action candidates”

(2) Improving through self-plays (reinforcement learning)

Improving by playing against itself

Expert Moves Imitator Model (w/ CNN)	vs	Expert Moves Imitator Model (w/ CNN)



1. Reducing "action candidates"

(2) Improving through self-plays (reinforcement learning)

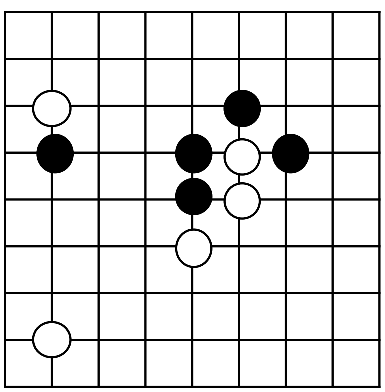
Expert Moves Imitator Model (w/ CNN) **VS** **Expert Moves Imitator Model (w/ CNN)**

Return: board positions, win/lose info

1. Reducing “action candidates”

(2) Improving through self-plays (reinforcement learning)

Board position



Expert Moves Imitator Model (w/ CNN)

win/loss

Loss

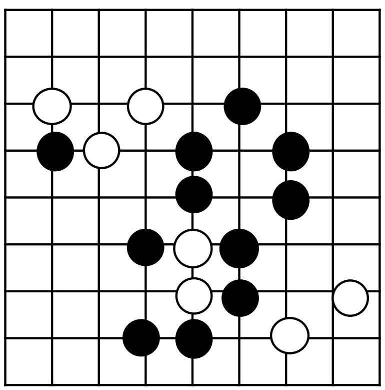
z = -1

Training: $\Delta\rho \propto \frac{\partial \log p_\rho(a_t|s_t)}{\partial \rho} z_t$.

1. Reducing "action candidates"

(2) Improving through self-plays (reinforcement learning)

Board position



Expert Moves Imitator Model (w/ CNN)

win/loss

Win

z = +1

Training: $\Delta\rho \propto \frac{\partial \log p_\rho(a_t|s_t)}{\partial \rho} z_t$.

1. Reducing “action candidates”

(2) Improving through self-plays (reinforcement learning)

Older models vs. newer models

Updated Model ver 1.1	VS	Updated Model ver 1.3

It uses the same topology as the expert moves imitator model, and just uses the updated parameters

→ **Return**: board positions, win/lose info

1. Reducing "action candidates"

(2) Improving through self-plays (reinforcement learning)

Updated Model ver 1.3 **VS** **Updated Model ver 1.7**

Return: board positions, win/lose info

1. Reducing “action candidates”

(2) Improving through self-plays (reinforcement learning)

Updated Model ver 1.5 **VS** **Updated Model ver 2.0**

Return: board positions, win/lose info

1. Reducing “action candidates”

(2) Improving through self-plays (reinforcement learning)

Updated Model ver 3204.1 **vs** **Updated Model ver 46235.2**

Return: board positions, win/lose info

1. Reducing “action candidates”

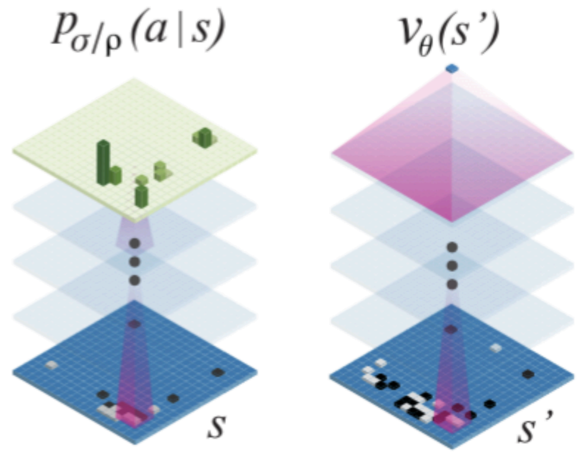
(2) Improving through self-plays (reinforcement learning)

Expert Moves Imitator Model	**VS**	**Updated Model ver 1,000,000**

The final model wins 80% of the time when playing against the first model

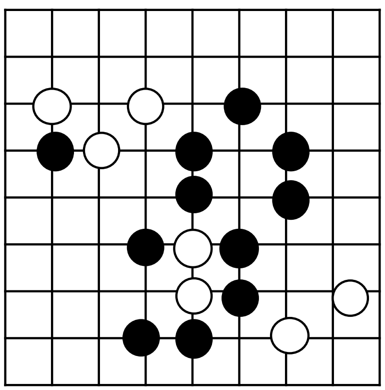
2. Board Evaluation

2. Board Evaluation



Adds a regression layer to the model
Predicts values between 0~1
Close to 1: a good board position
Close to 0: a bad board position

Board Position



Updated Model ver 1,000,000

Value Prediction Model (Regression)

Win / Loss

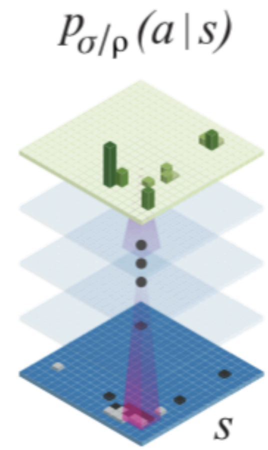
Win (0~1)

Training: $$\Delta\theta \propto \frac{\partial v_\theta(s)}{\partial \theta}(z - v_\theta(s))$$

Reducing Search Space

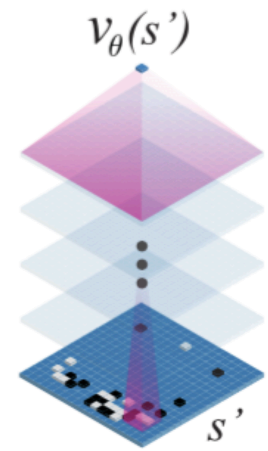
1. **Reducing "action candidates" (Breadth Reduction)**

   Policy Network

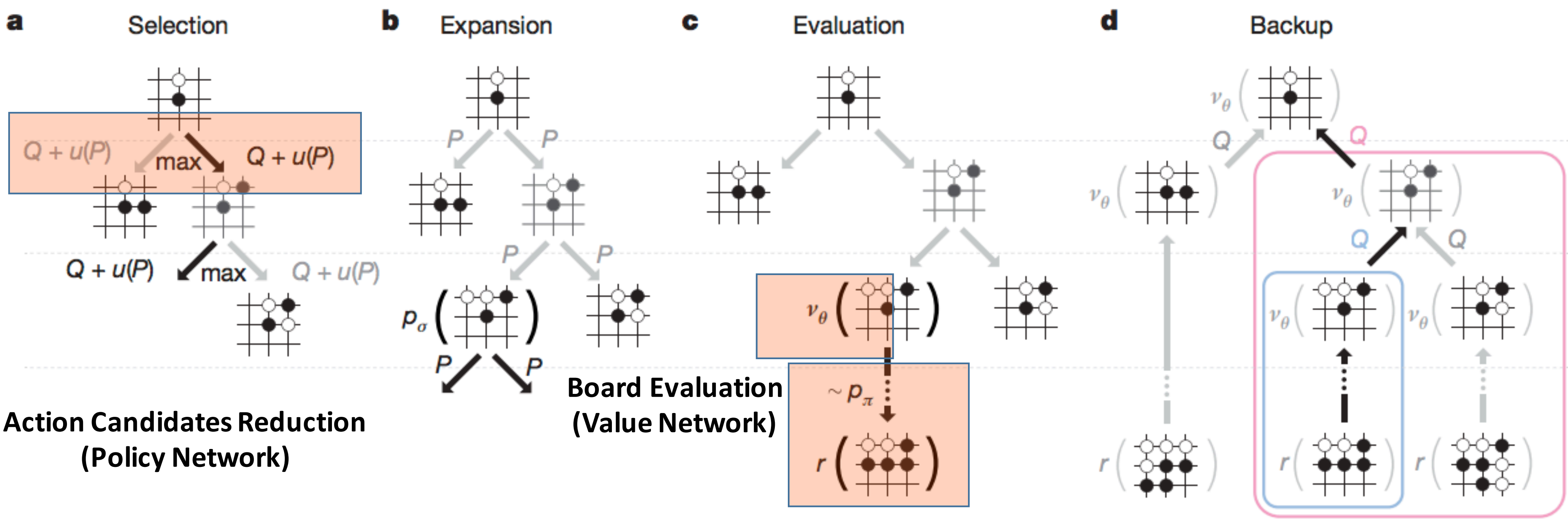


2. **Board Evaluation (Depth Reduction)**

   Value Network



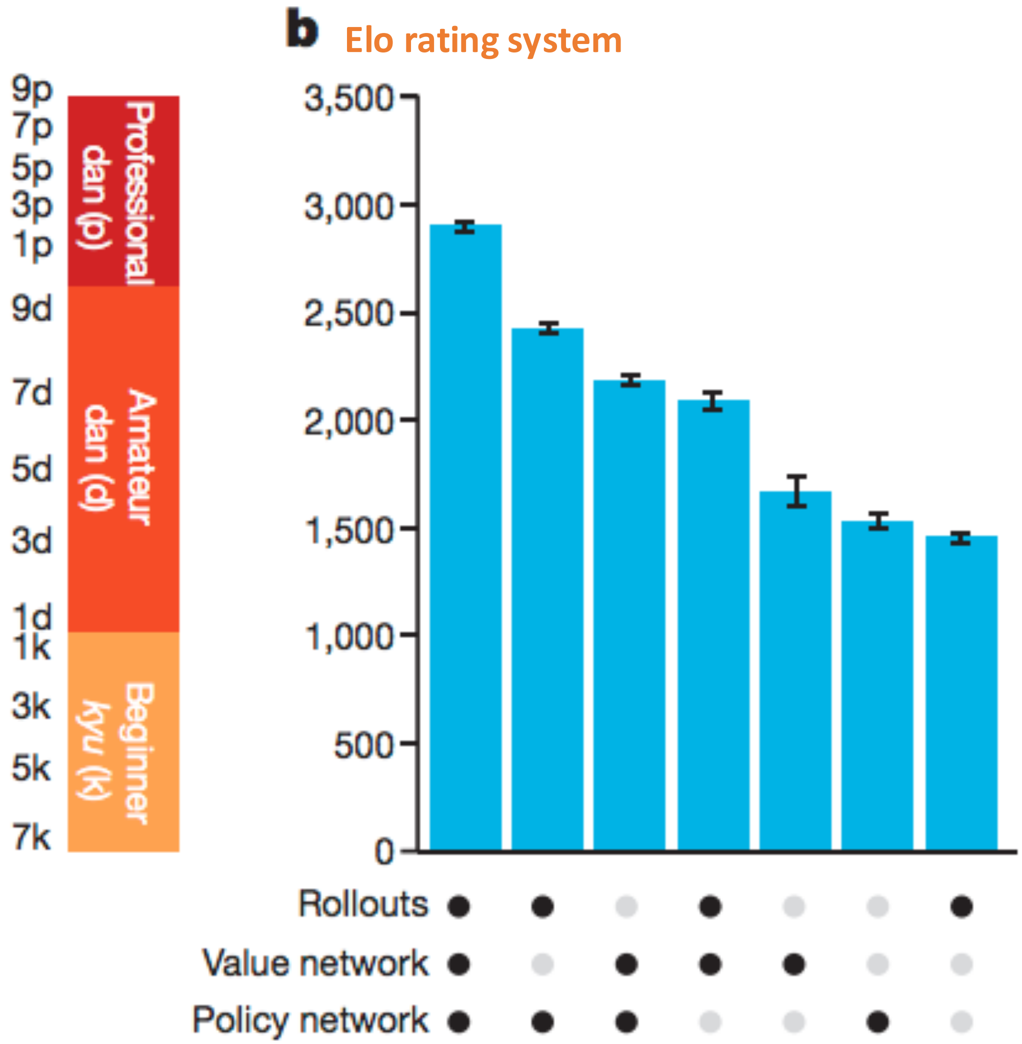
Looking ahead (w/ Monte Carlo Search Tree)



(Rollout): Faster version of estimating p(a|s)

→ uses shallow networks (3 ms → 2µs)

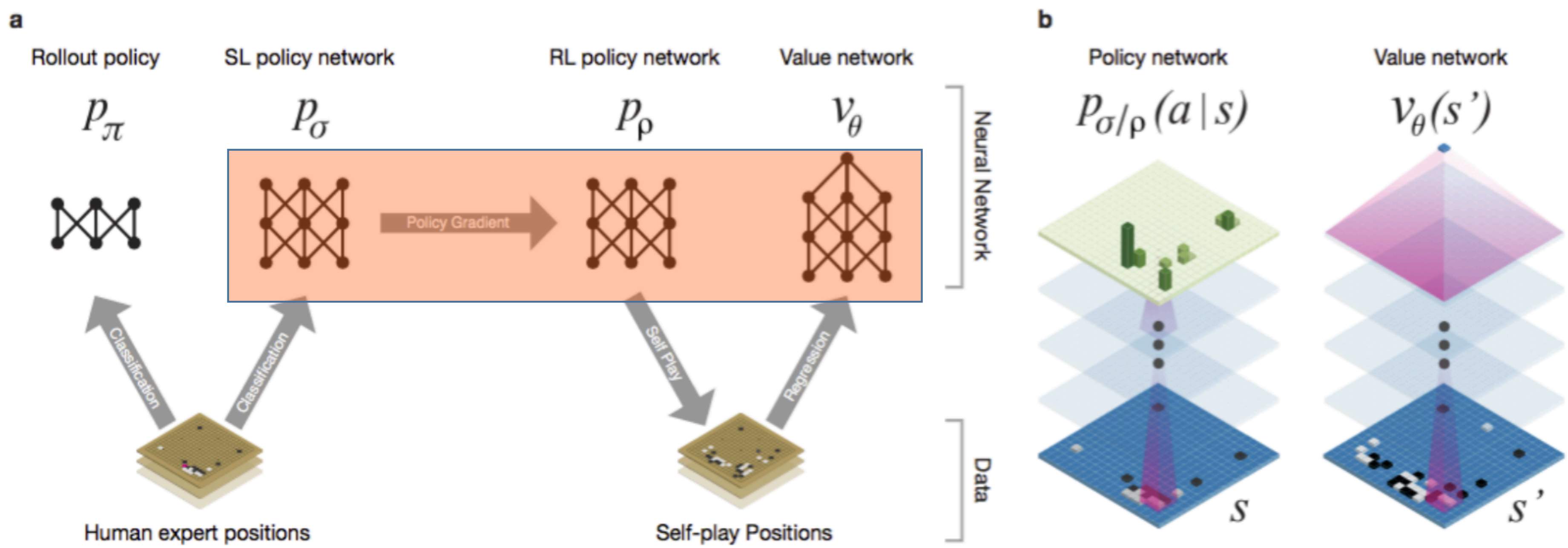
Results



Performance with different combinations of AlphaGo components

Takeaways

Use the networks trained for a certain task (with different loss objectives) for several other tasks

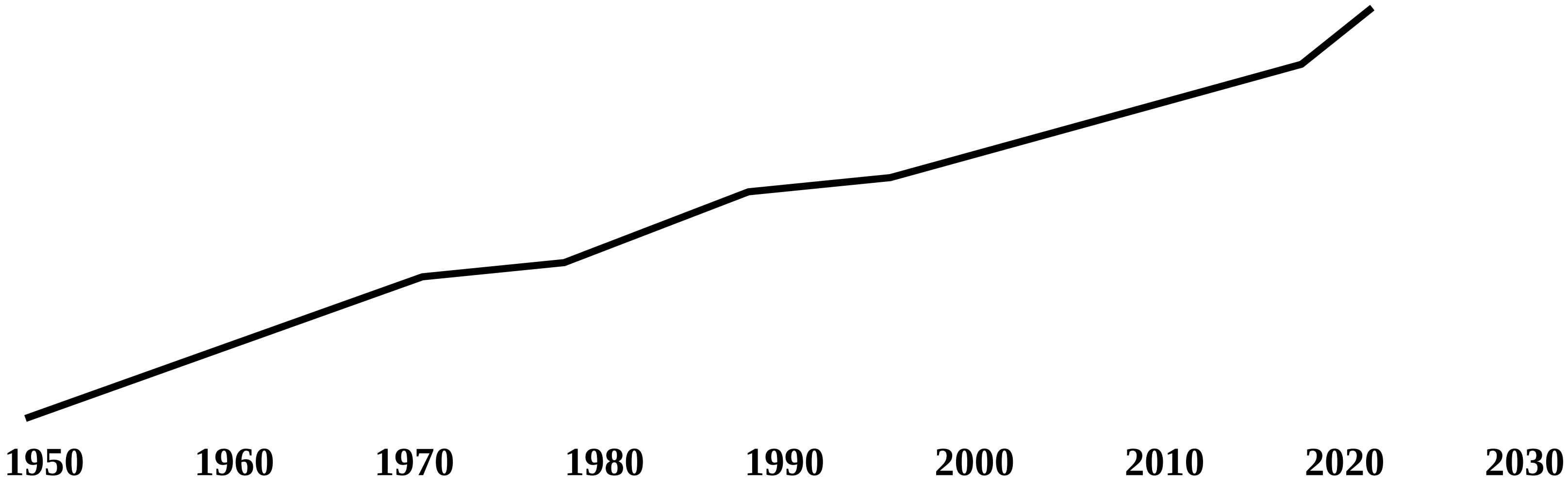


AlphaGo [Silver et al., ’15]: An AI milestone

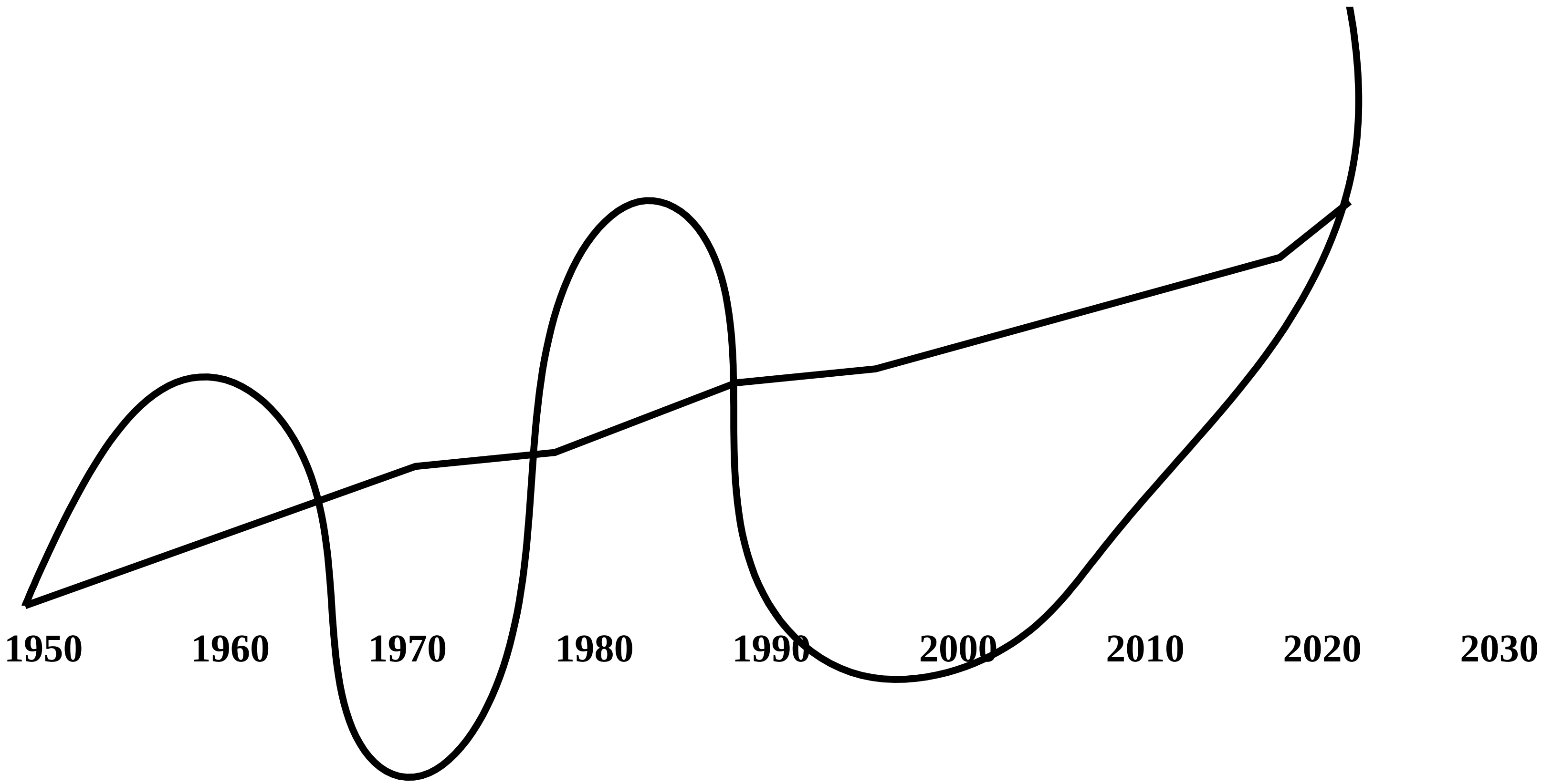
- Integrated two **existing** AI technologies
  - Supervised learning (Deep learning)
  - **Reinforcement learning**
- Does **not** solve AI
  - The end of the line for:

    2-player zero-sum discrete finite deterministic games of perfect information

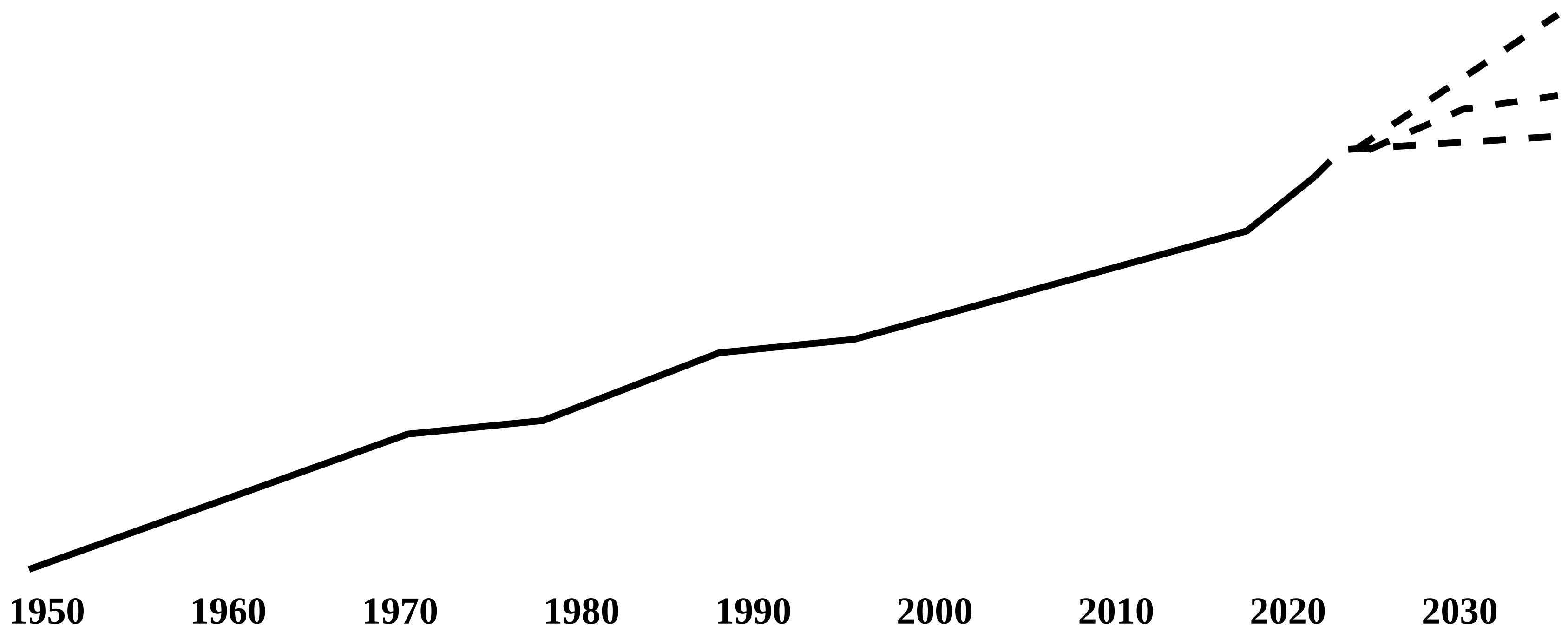
Reality



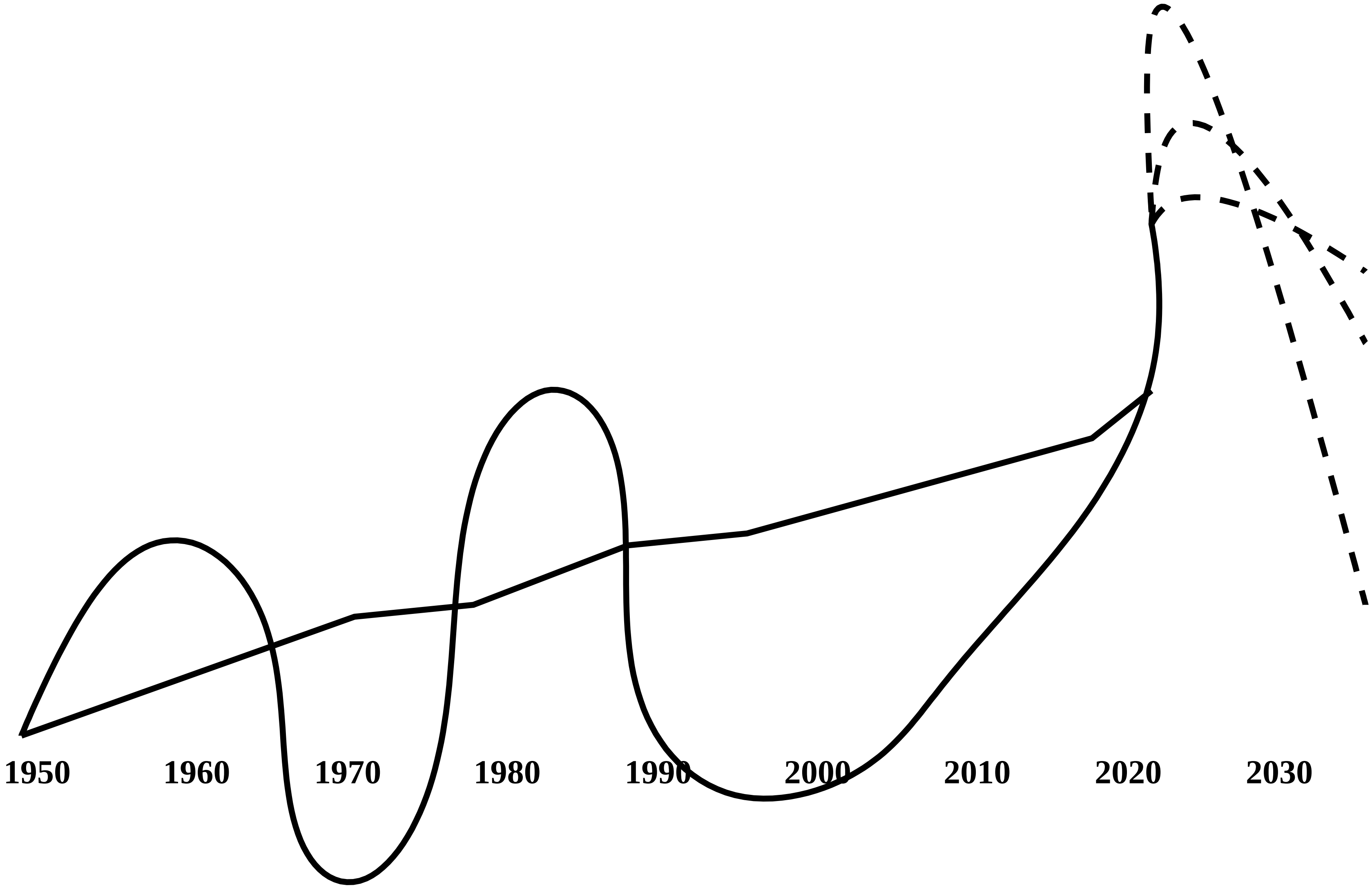
Perceptions



Uncertainty



Perception Uncertainty



Selected RL Contributions

- Human interaction

Selected RL Contributions

- Human interaction
  - Advice, Demonstration

Selected RL Contributions

- Human interaction
  - Advice, Demonstration
  - Positive/Negative Feedback

[Knox & Stone, '09]

Selected RL Contributions

- Human interaction
  - Advice, Demonstration
  - Positive/Negative Feedback
- Transfer learning for RL

[Knox & Stone, ’09]
[Taylor & Stone, ’07]

Selected RL Contributions

- Human interaction
  - Advice, Demonstration
  - Positive/Negative Feedback

[Knox & Stone, '09]

- Transfer learning for RL

[Taylor & Stone, '07]

- Adaptive/hierarchical representations

[Whiteson & Stone, '05], [Jong & Stone, '08]

Selected RL Contributions

- Human interaction
  - Advice, Demonstration
  - Positive/Negative Feedback

[Knox & Stone, '09]

- Transfer learning for RL [Taylor & Stone, '07]
- Adaptive/hierarchical representations

  [Whiteson & Stone, '05], [Jong & Stone, '08]
- **TEXPLORE** for Robot RL [Hester & Stone, '13]

Selected RL Contributions

- Human interaction
  - Advice, Demonstration
  - Positive/Negative Feedback

[Knox & Stone, ’09]

- Transfer learning for RL [Taylor & Stone, ’07]
- Adaptive/hierarchical representations

[Whiteson & Stone, ’05], [Jong & Stone, ’08]

- **TEXPLORE** for Robot RL [Hester & Stone, ’13]
  - Sample efficient; real-time

Selected RL Contributions

- Human interaction

  - Advice, Demonstration
  - Positive/Negative Feedback

[Knox & Stone, '09]

- Transfer learning for RL [Taylor & Stone, '07]

- Adaptive/hierarchical representations

[Whiteson & Stone, '05], [Jong & Stone, '08]

- **TEXPLORE** for Robot RL [Hester & Stone, '13]
  - Sample efficient; real-time
  - Continuous state; delayed effects

UT Austin Villa 2014 RoboCup 3D Simulation League Champion via Overlapping Layered Learning

Patrick MacAlpine, Mike Depinet, and Peter Stone

Layered Learning

- For domains too complex for tractably mapping state features $S \longmapsto$ outputs O
- Hierarchical subtask decomposition given: $\{L_1, L_2, \ldots, L_n\}$

Layered Learning

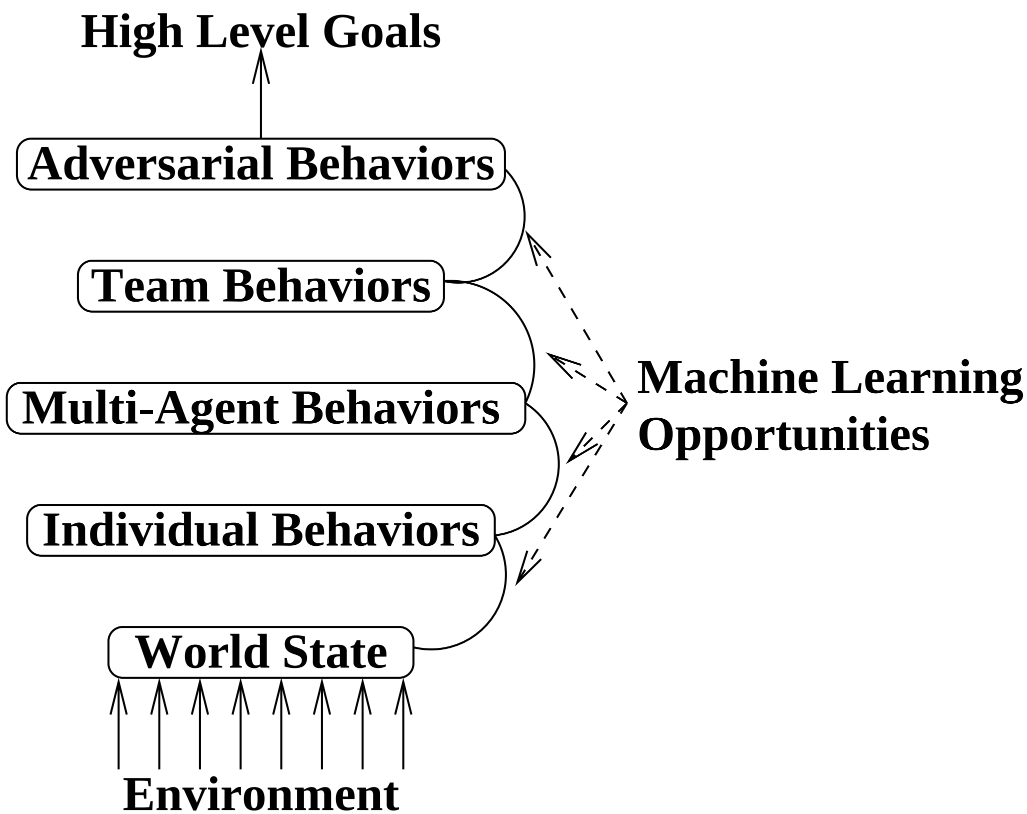
- For domains too complex for tractably mapping state features S $\longmapsto$ outputs O
- Hierarchical subtask decomposition given: $\{L_1, L_2, \ldots, L_n\}$
- Machine learning: exploit data to train, adapt

Layered Learning

- For domains too complex for tractably mapping state features $S \longmapsto$ outputs O
- Hierarchical subtask decomposition given: $\{L_1, L_2, \ldots, L_n\}$
- Machine learning: exploit data to train, adapt
- **Synthesis:** Learning in one layer feeds into next layer

Layered Learning

- For domains too complex for tractably mapping state features S $\longmapsto$ outputs O
- Hierarchical subtask decomposition given: $\{L_1, L_2, \ldots, L_n\}$
- Machine learning: exploit data to train, adapt
- **Synthesis:** Learning in one layer feeds into next layer



Layered Learning in Practice

First applied in simulated robot soccer [Stone & Veloso, ’97]

	Strategic Level	Example
L_1	individual	ball interception
L_2	multiagent	pass evaluation
L_3	**team**	pass selection

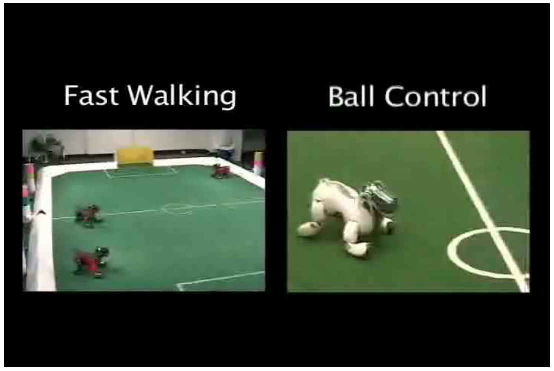
Layered Learning in Practice

First applied in simulated robot soccer [Stone & Veloso, '97]

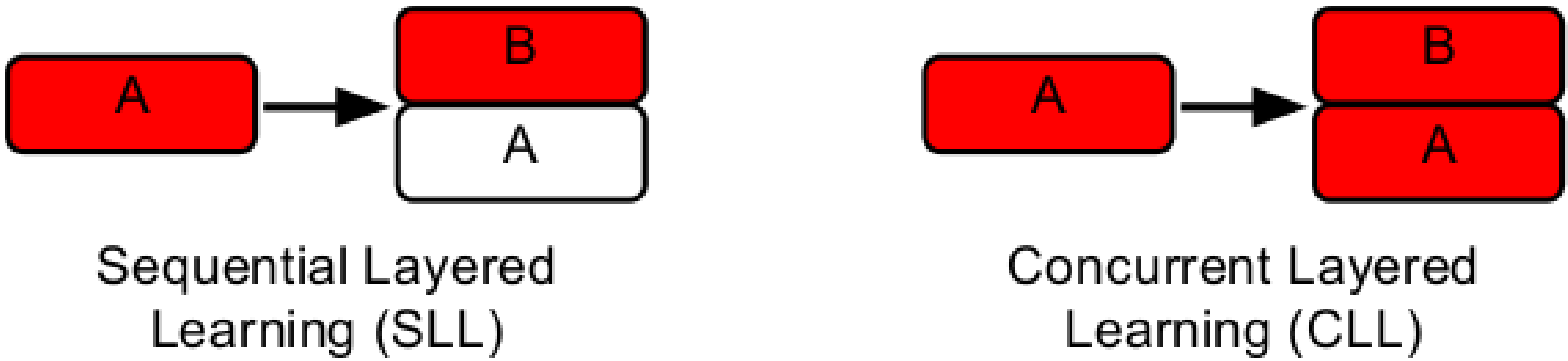
	Strategic Level	Example
L_1	individual	ball interception
L_2	multiagent	pass evaluation
L_3	**team**	pass selection

Later applied on real robots [Stone, Kohl, & Fidelman, '06]

	Strategic Level	Example
L_1	individual	fast walking
L_2	individual	ball control



Layered Learning Paradigms

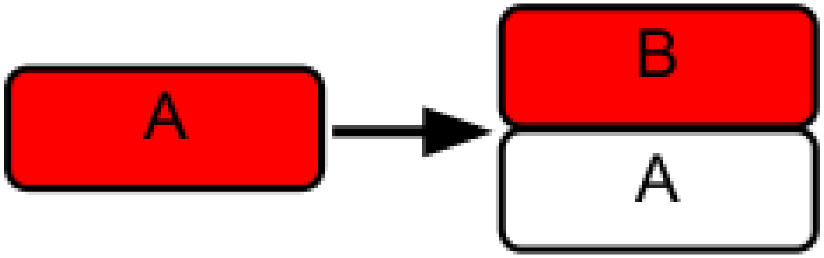


DESCRIPTIONS:

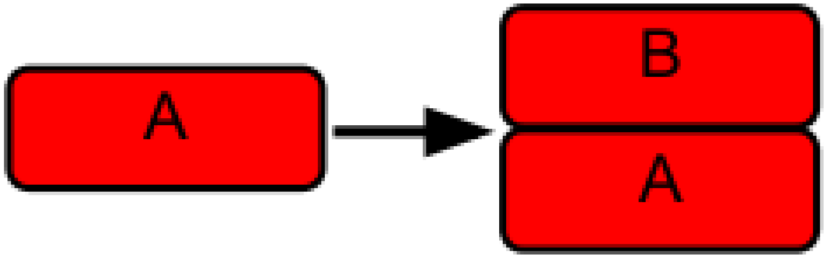
Sequential Layered Learning: Freeze parameters of layer after learning before learning of the next layer

Concurrent Layered Learning: Keep parameters of layer open during learning of the next layer

Layered Learning Paradigms



Sequential Layered Learning (SLL)



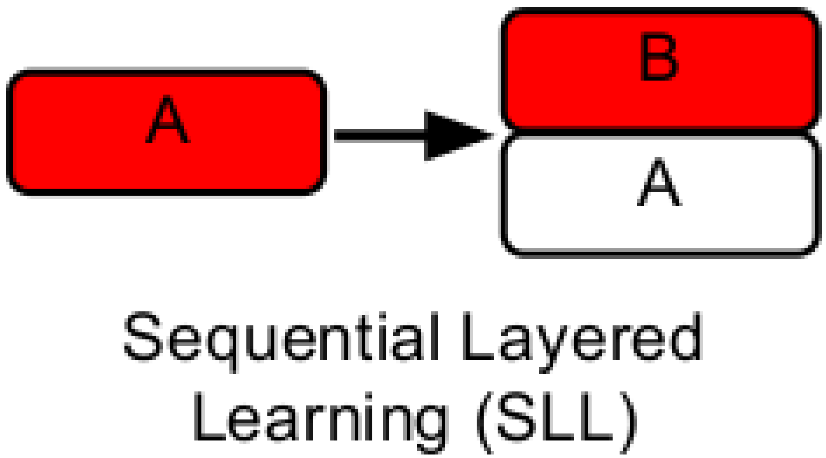
Concurrent Layered Learning (CLL)

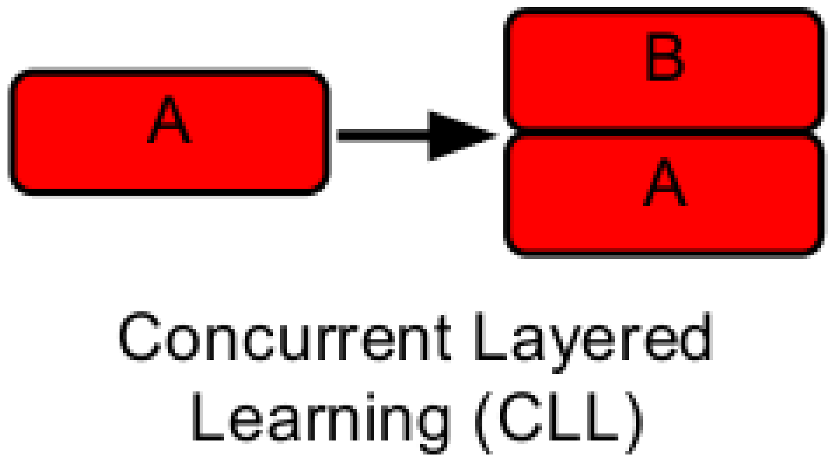
PROBLEMS:

Sequential Layered Learning: Can be too limiting in the joint layer policy seach space

Concurrent Layered Learning: The increased dimensionality can make learning harder or intractible

Layered Learning Paradigms



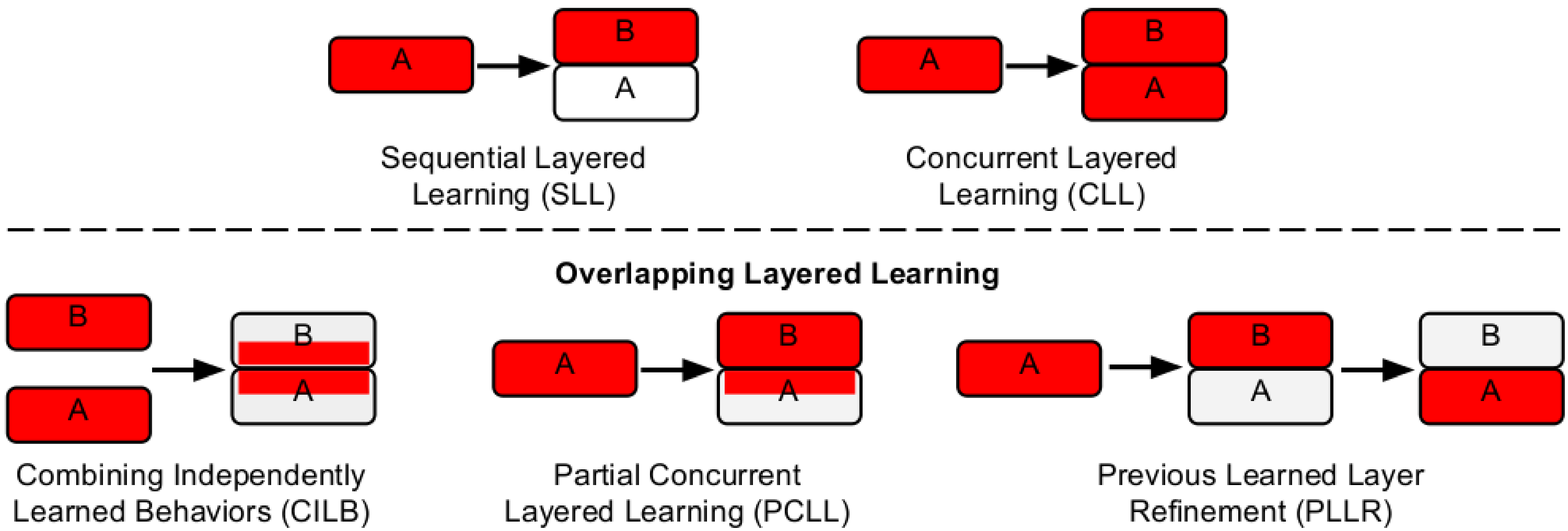


SOLUTION:

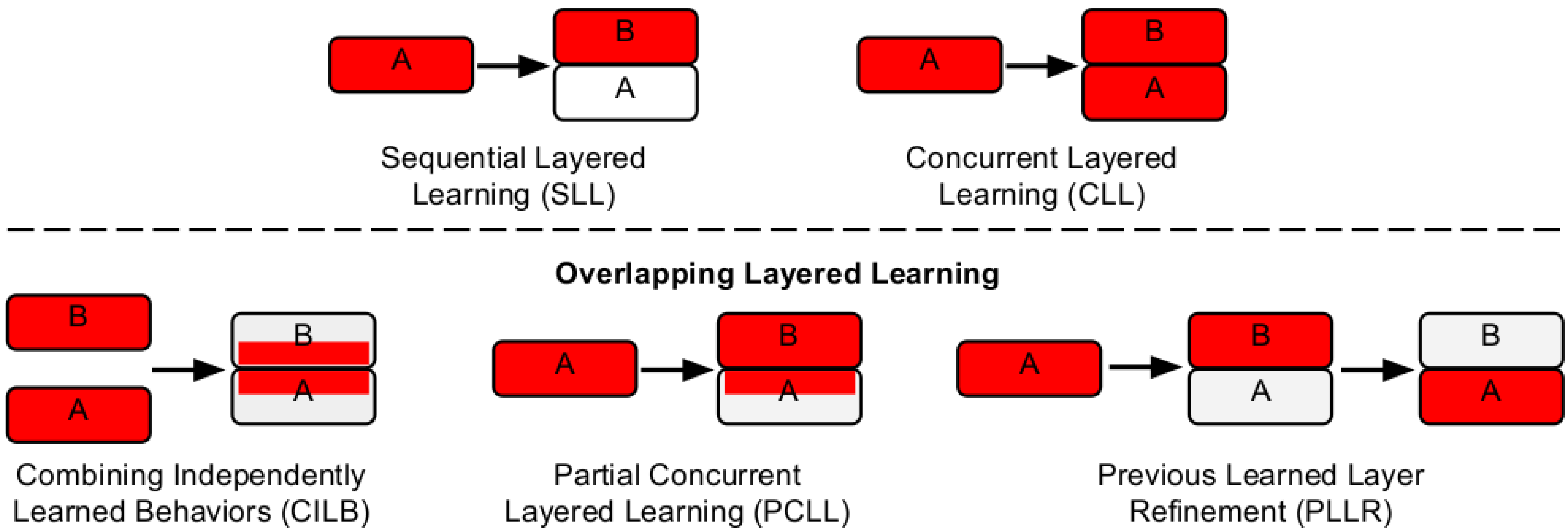
Overlapping Layered Learning: Tradeoff between freezing or keeping open previous learned layers

Optimizes "seam" or overlap between behaviors: keeps some parts of previously learned layers open during subsequent learning

Overlapping Layered Learning

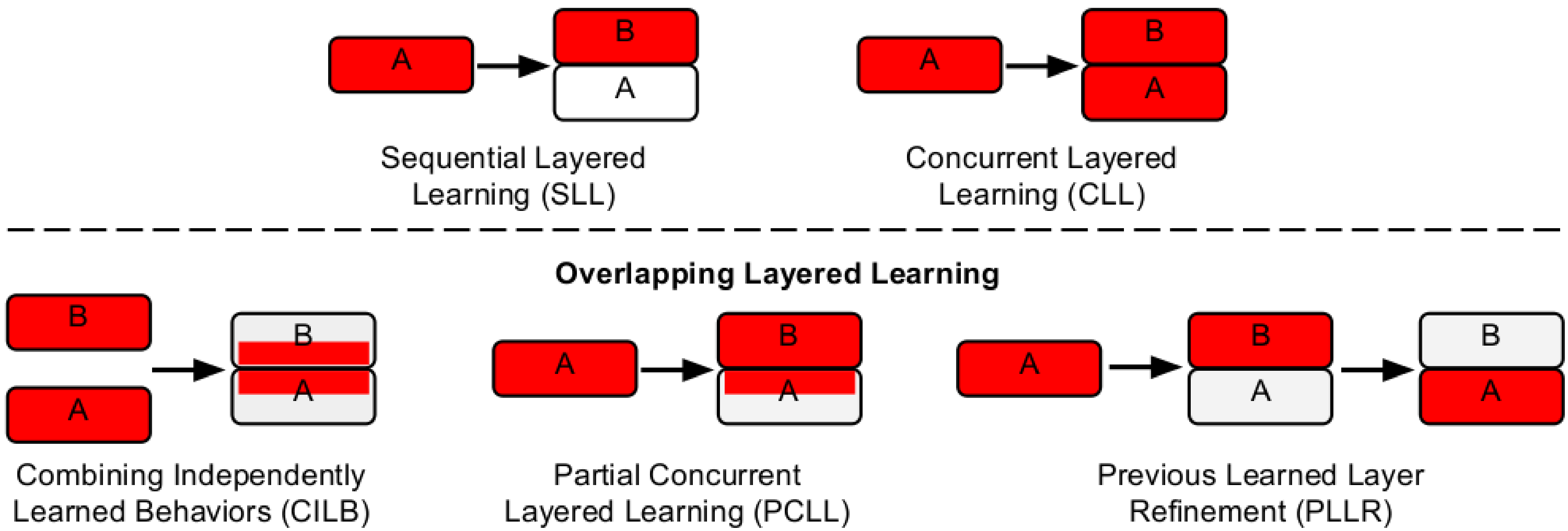


Overlapping Layered Learning



Combining Independently Learned Behaviors: Behaviors learned indpendently and then combined by relearning subset of behaviors' parameters

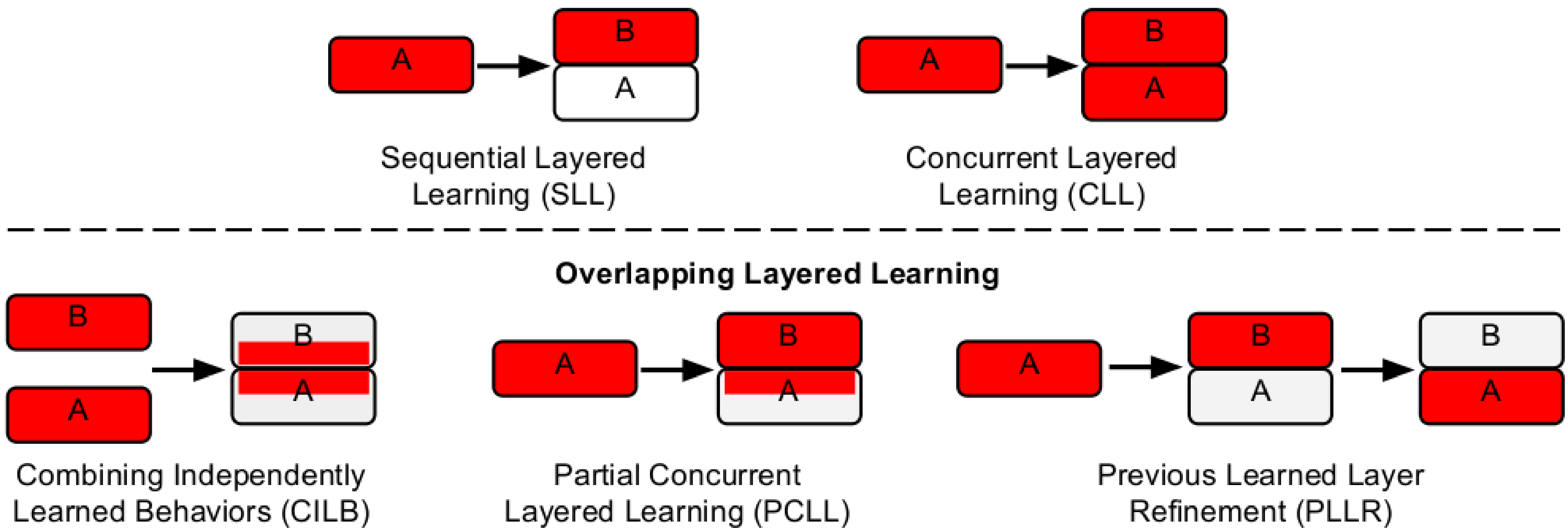
Overlapping Layered Learning



Combining Independently Learned Behaviors: Behaviors learned indpendently and then combined by relearning subset of behaviors' parameters

Partial Concurrent Layered Learning: Part, but not all, of a previously learned layer's behaviors are left open

Overlapping Layered Learning



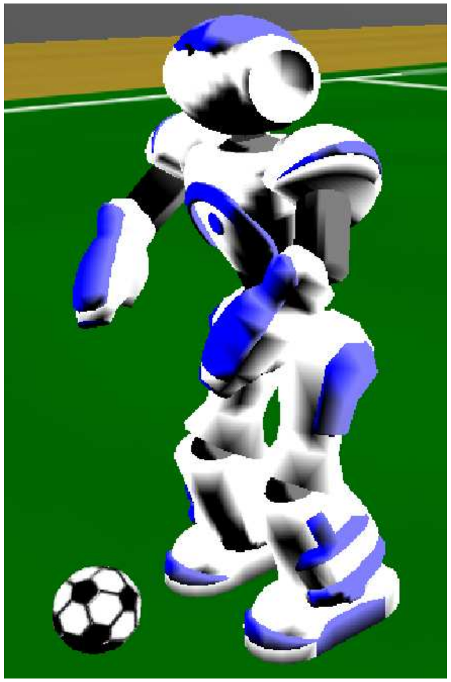
Combining Independently Learned Behaviors: Behaviors learned indpendently and then combined by relearning subset of behaviors' parameters

Partial Concurrent Layered Learning: Part, but not all, of a previously learned layer's behaviors are left open

Previous Learned Layer Refinement: After a pair of layers is learned, part or all of the initial layer is unfrozen

RoboCup 3D Simulation Domain

- Teams of 11 vs 11 autonomous robots play soccer
- Realistic physics using Open Dynamics Engine (ODE)
- Simulated robots modeled after Aldebaron Nao robot
- Robot receives noisy visual information about environment
- Robots can communicate over limited bandwidth channel



RoboCup Champions 2011, 2012

RoboCup Champions 2011, 2012

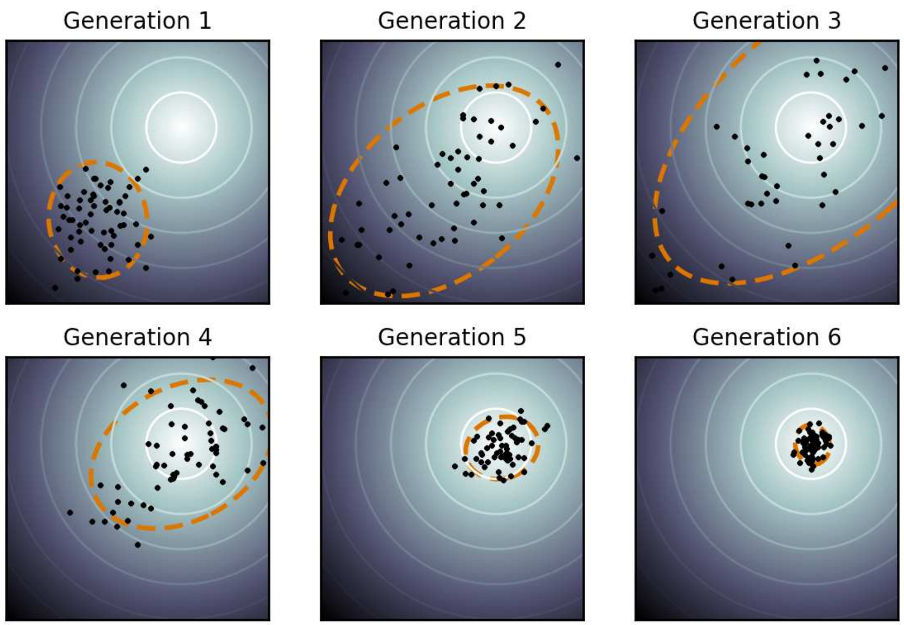
Humanoid Walk Learning via Layered Learning and CMA-ES

- Parameterized **double linear inverted pendulum model**

RoboCup Champions 2011, 2012

Humanoid Walk Learning via Layered Learning and CMA-ES

- Parameterized **double linear inverted pendulum model**



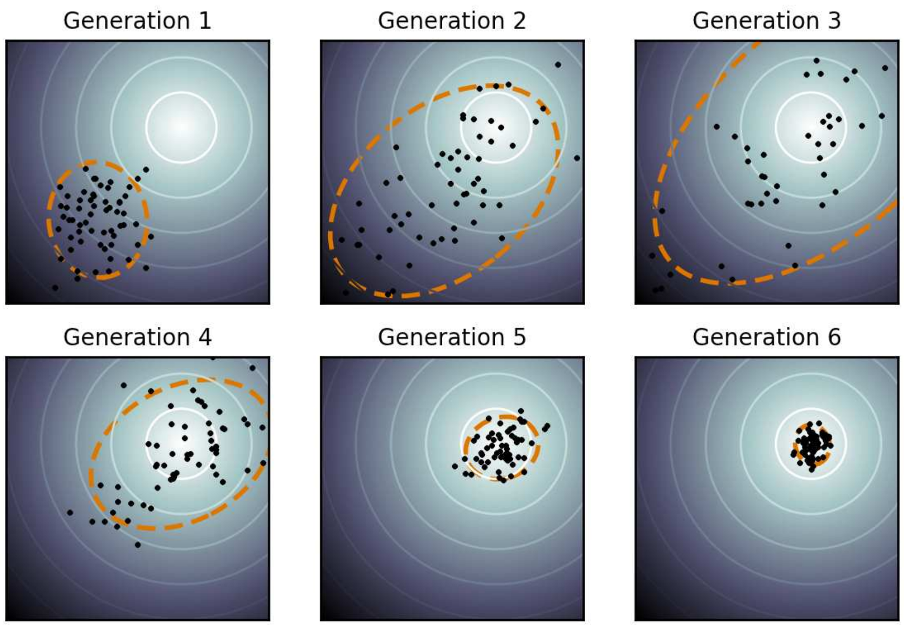
CMA-ES
[Hansen, ’09]

- Stochastic, derivative-free, numerical optimization method
- Candidates sampled from multidimensional Gaussian

RoboCup Champions 2011, 2012

Humanoid Walk Learning via Layered Learning and CMA-ES

- Parameterized **double linear inverted pendulum model**



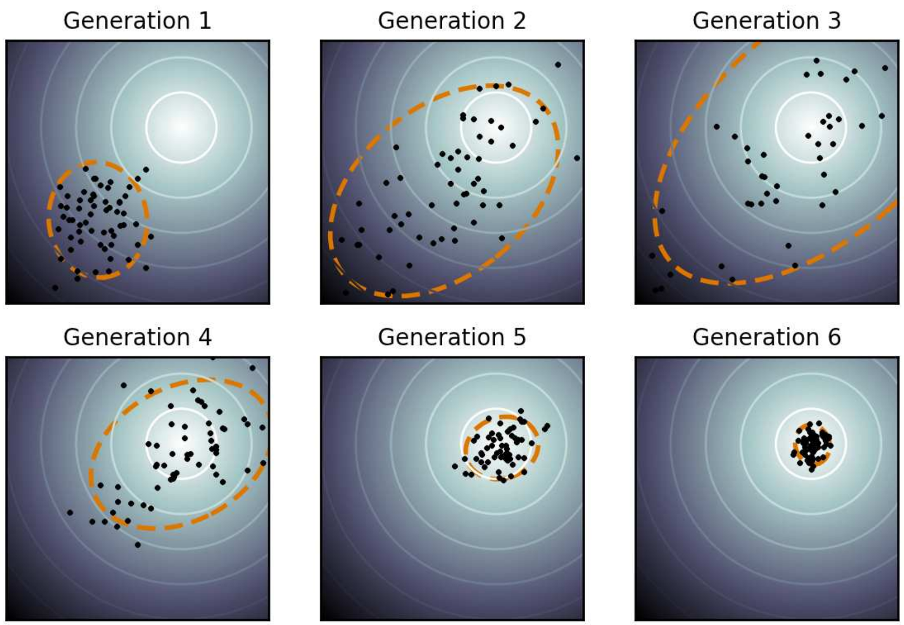
CMA-ES
[Hansen, '09]

- Stochastic, derivative-free, numerical optimization method
- Candidates sampled from multidimensional Gaussian
  - Mean maximizes likelihood of previous successes
  - Covariance update controls search step sizes

RoboCup Champions 2011, 2012

Humanoid Walk Learning via Layered Learning and CMA-ES

- Parameterized **double linear inverted pendulum model**

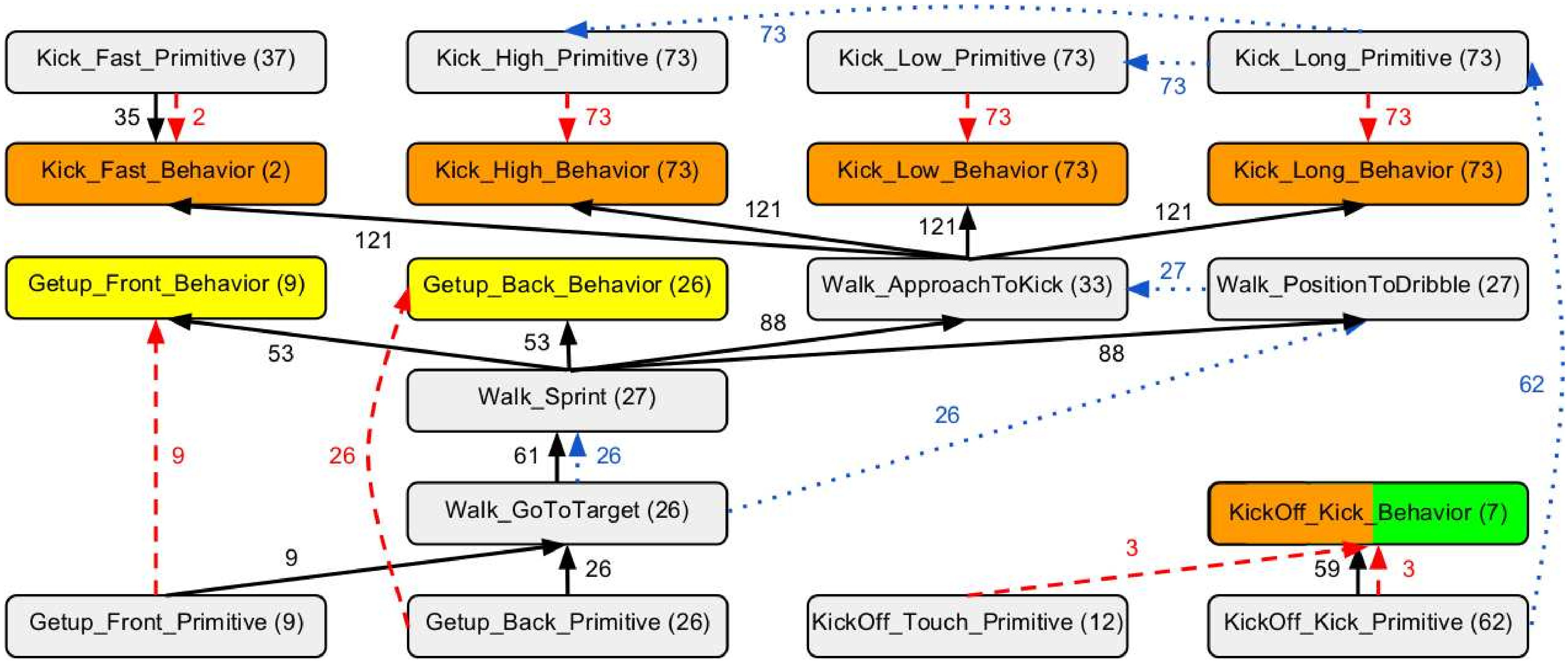


CMA-ES
[Hansen, '09]

- Stochastic, derivative-free, numerical optimization method
- Candidates sampled from multidimensional Gaussian
  - Mean maximizes likelihood of previous successes
  - Covariance update controls search step sizes

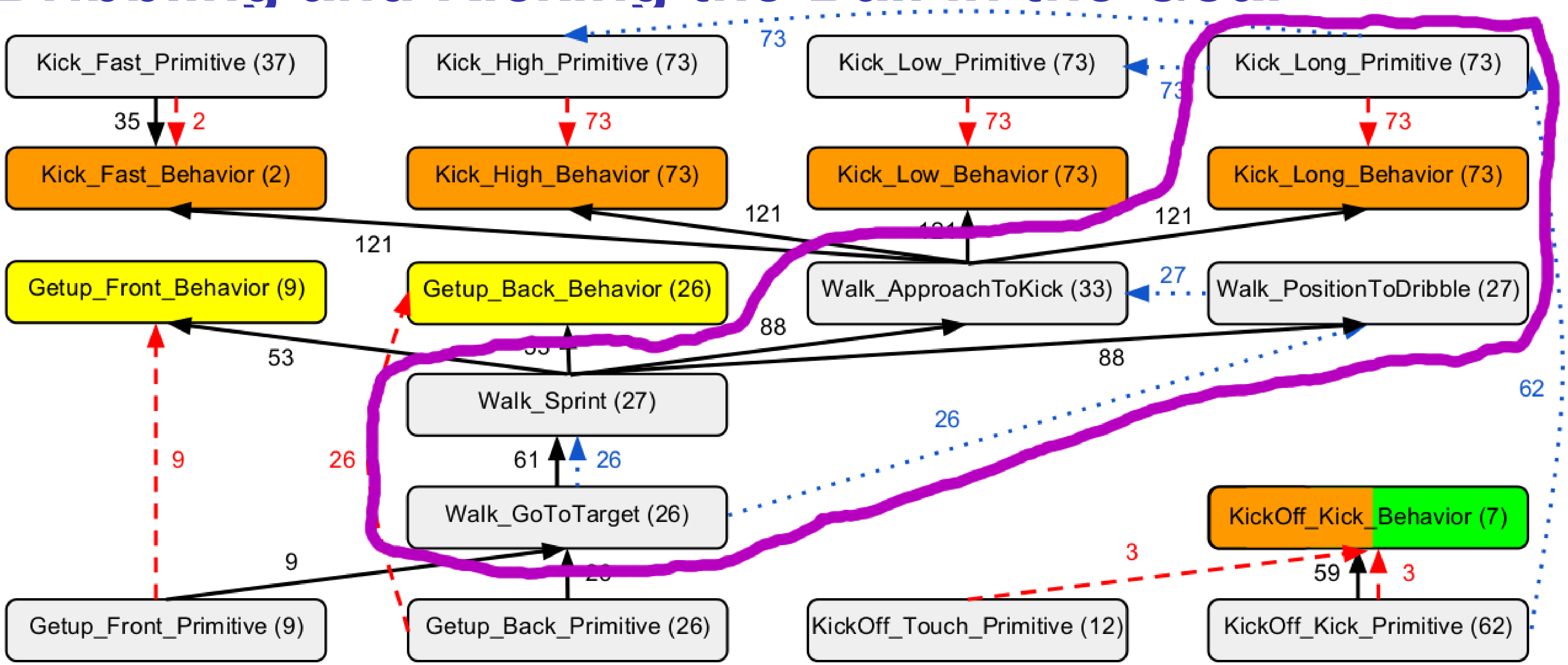
Initial walk **No layered learning** **2 layers**
3 layers **Final walk** **Champs*2**

Learned Layers



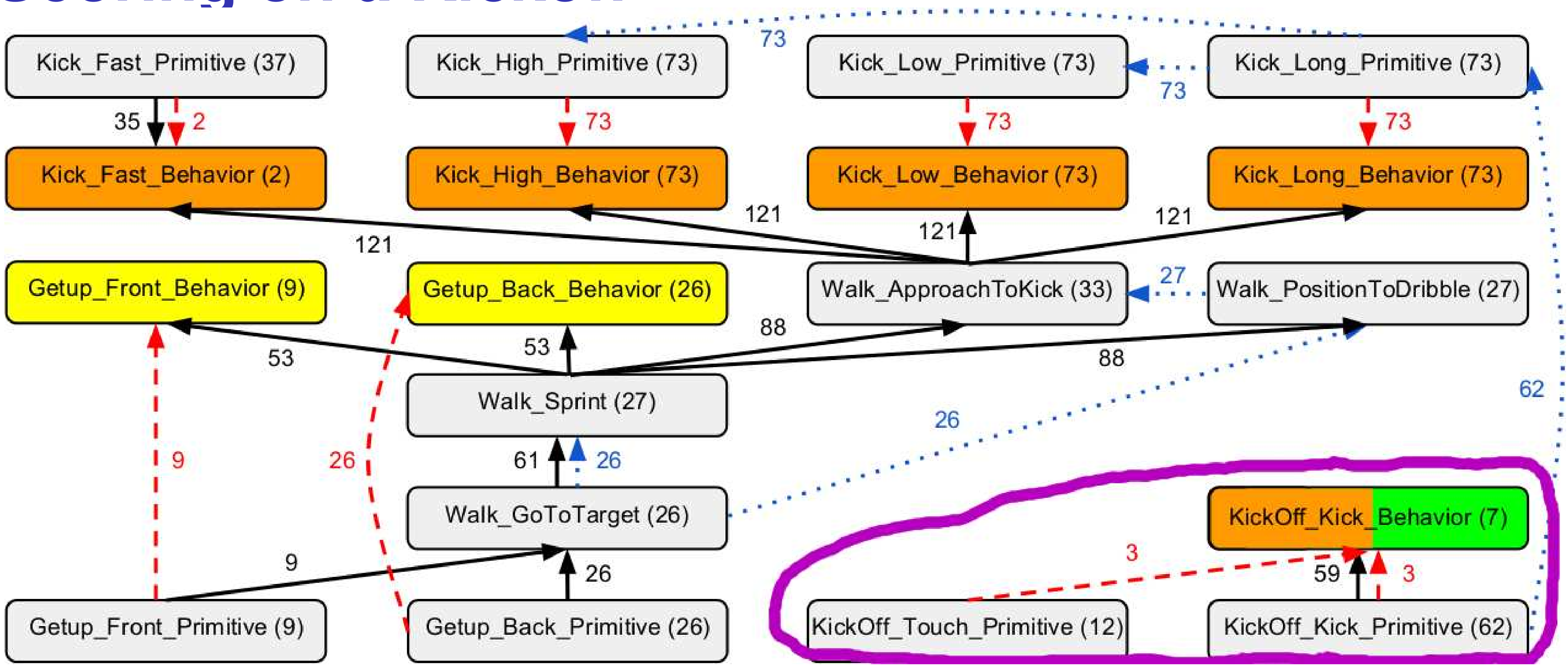
- 19 learned behaviors for standing up, walking, and kicking
  - CILB, PCLL, PLLR
- Over 500 parameters optimized during the course of learning
  - frozen, passed, seeded

Dribbling and Kicking the Ball in the Goal



- Four different walk parameter sets
  - Target/sprint/position + **approach ball to kick**
- Learn **fixed kick**
- Combine **kick with walk**: combine independent layers (CILB)
  - Overlap kick parameters for positioning
- Final **walk and kick**

Scoring on a Kickoff



- Kickoffs indirect (2 players must touch to score)
- Learn **fixed kick**
- Learn **touch** behavior **interferes**
- Combine **kick with touch**
  - Relearn position patterns: combine independent layers (CILB)
  - Learn new timing parameter: partial concurrent (PCLL)

Impact of Overlapping Layered Learning

1000 games vs. top 3 teams from 2013

Impact of Overlapping Layered Learning

1000 games vs. top 3 teams from 2013

	Average Goal Difference		
Opponent	**Full Team**	**No Kickoff**	**Dribble Only**
apollo3d	2.703 (0.041)	2.062 (0.038)	1.861 (0.034)
UTAustinVilla2013	1.589 (0.036)	1.225 (0.033)	0.849 (0.025)
fcportugal3d	3.991 (0.051)	3.189 (0.048)	1.584 (0.030)

No Kickoff: On kickoff, kick ball deep into opponent's end

Dribble Only: No kicking

Repetition on Different Robot Types

Type 0: Standard Nao model

Type 1: Longer legs and arms

Type 2: Quicker moving legs

Type 3: Wider hips and longest legs and arms

Type 4: Added toes to foot

Repetition on Different Robot Types

Type 0: Standard Nao model

Type 1: Longer legs and arms

Type 2: Quicker moving legs

Type 3: Wider hips and longest legs and arms

Type 4: Added toes to foot

	Avg. Goal Difference per Robot Type				
Opponent	**Type 0**	**Type 1**	**Type 2**	**Type 3**	**Type 4**
apollo3d	1.787	1.819	1.820	1.543	2.827
UTAustinVilla2013	0.992	0.892	1.276	0.573	1.141
fcportugal3d	2.423	3.025	3.275	2.678	4.033

Repetition on Different Robot Types

- Type 0: Standard Nao model
- Type 1: Longer legs and arms
- Type 2: Quicker moving legs
- Type 3: Wider hips and longest legs and arms
- Type 4: Added toes to foot

	Avg. Goal Difference per Robot Type				
Opponent	**Type 0**	**Type 1**	**Type 2**	**Type 3**	**Type 4**
apollo3d	1.787	1.819	1.820	1.543	2.827
UTAustinVilla2013	0.992	0.892	1.276	0.573	1.141
fcportugal3d	2.423	3.025	3.275	2.678	4.033

Computation per type

$\approx$ 700k parameter sets evaluated

$\approx$ 1.5 years compute time ($\approx$ 50 hours on condor cluster)

RoboCup 2014

Won competition with undefeated record: outscored opps 52–0

Opponent	Avg. Goal Diff.	Record (W-L-T)	Goals (F/A)	KO Score %
BahiaRT	2.075 (0.030)	990-0-10	2092/17	96.2
FCPortugal	2.642 (0.034)	986-0-14	2748/106	83.4
magmaOffenburg	2.855 (0.035)	990-0-10	2864/9	88.3
RoboCanes	3.081 (0.046)	974-0-26	3155/74	69.4
FUT-K	3.236 (0.039)	998-0-2	3240/4	96.3
SEU_Jolly	4.031 (0.062)	995-0-5	4034/3	87.6
KarachiKoalas	5.681 (0.046)	1000-0-0	5682/1	87.5
ODENS	7.933 (0.041)	1000-0-0	7933/0	92.1
HfutEngine	8.510 (0.050)	1000-0-0	8510/0	94.7
Mithras3D	8.897 (0.041)	1000-0-0	8897/0	90.4
L3M-SIM	9.304 (0.043)	1000-0-0	9304/0	93.7

RoboCup 2014

Won competition with undefeated record: outscored opps 52–0

Opponent	Avg. Goal Diff.	Record (W-L-T)	Goals (F/A)	KO Score %
BahiaRT	2.075 (0.030)	990-0-10	2092/17	96.2
FCPortugal	2.642 (0.034)	986-0-14	2748/106	83.4
magmaOffenburg	2.855 (0.035)	990-0-10	2864/9	88.3
RoboCanes	3.081 (0.046)	974-0-26	3155/74	69.4
FUT-K	3.236 (0.039)	998-0-2	3240/4	96.3
SEU_Jolly	4.031 (0.062)	995-0-5	4034/3	87.6
KarachiKoalas	5.681 (0.046)	1000-0-0	5682/1	87.5
ODENS	7.933 (0.041)	1000-0-0	7933/0	92.1
HfutEngine	8.510 (0.050)	1000-0-0	8510/0	94.7
Mithras3D	8.897 (0.041)	1000-0-0	8897/0	90.4
L3M-SIM	9.304 (0.043)	1000-0-0	9304/0	93.7

- After: 11,000 games: won all by 67 (no losses)

RoboCup 2014

Won competition with undefeated record: outscored opps 52–0

Opponent	Avg. Goal Diff.	Record (W-L-T)	Goals (F/A)	KO Score %
BahiaRT	2.075 (0.030)	990-0-10	2092/17	96.2
FCPortugal	2.642 (0.034)	986-0-14	2748/106	83.4
magmaOffenburg	2.855 (0.035)	990-0-10	2864/9	88.3
RoboCanes	3.081 (0.046)	974-0-26	3155/74	69.4
FUT-K	3.236 (0.039)	998-0-2	3240/4	96.3
SEU_Jolly	4.031 (0.062)	995-0-5	4034/3	87.6
KarachiKoalas	5.681 (0.046)	1000-0-0	5682/1	87.5
ODENS	7.933 (0.041)	1000-0-0	7933/0	92.1
HfutEngine	8.510 (0.050)	1000-0-0	8510/0	94.7
Mithras3D	8.897 (0.041)	1000-0-0	8897/0	90.4
L3M-SIM	9.304 (0.043)	1000-0-0	9304/0	93.7

- After: 11,000 games: won all by 67 (no losses)
- **Highlights** from Final vs. RoboCanes (University of Miami)

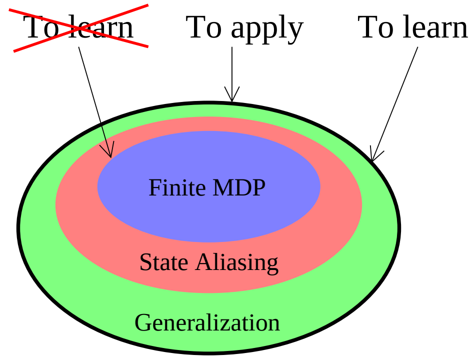
RoboCup 2014

Won competition with undefeated record: outscored opps 52–0

Opponent	Avg. Goal Diff.	Record (W-L-T)	Goals (F/A)	KO Score %
BahiaRT	2.075 (0.030)	990-0-10	2092/17	96.2
FCPortugal	2.642 (0.034)	986-0-14	2748/106	83.4
magmaOffenburg	2.855 (0.035)	990-0-10	2864/9	88.3
RoboCanes	3.081 (0.046)	974-0-26	3155/74	69.4
FUT-K	3.236 (0.039)	998-0-2	3240/4	96.3
SEU_Jolly	4.031 (0.062)	995-0-5	4034/3	87.6
KarachiKoalas	5.681 (0.046)	1000-0-0	5682/1	87.5
ODENS	7.933 (0.041)	1000-0-0	7933/0	92.1
HfutEngine	8.510 (0.050)	1000-0-0	8510/0	94.7
Mithras3D	8.897 (0.041)	1000-0-0	8897/0	90.4
L3M-SIM	9.304 (0.043)	1000-0-0	9304/0	93.7

- After: 11,000 games: won all by 67 (no losses)
- **Highlights** from Final vs. RoboCanes (University of Miami)
- More info: www.cs.utexas.edu/~AustinVilla/sim/3dsimulation/

Practical RL



- **R**epresentation
  - Selecting the Algorithm: parameterized domains [K.&S., MLJ 2011]
  - Adapting Representation: NEAT+Q [Whiteson & S., JMLR 2006]
- **I**nteraction
  - With adversaries: CMLeS [Chakraborty & S., ICML 2010]
  - With ad hoc teammates: PLASTIC [Barrett, thesis 2014]
  - With people: TAMER [Knox & S., AAMAS 2010]
- **S**ynthesis
  - Of Algorithms: Layered Learning [S., MIT Press 2000]
  - Of Concepts: Fitted R-MAXQ [Jong & S., ECML 2009]
- **M**ortality
  - Leverage the Past: Transfer Learning [Taylor, S., & Liu, JMLR 2007]
  - Acknowledge a Finite Future: TEXPLORE [Hester & S., MLJ 2013]

Making Friends on the Fly: Advances in Ad Hoc Teamwork

Samuel Barrett, Katie Genter, and Peter Stone

Ad Hoc Teamwork [Stone et al., AIJ 2013]

- Only in control of a single agent or subset of agents
- Unknown teammates
- Shared goals
- No pre-coordination

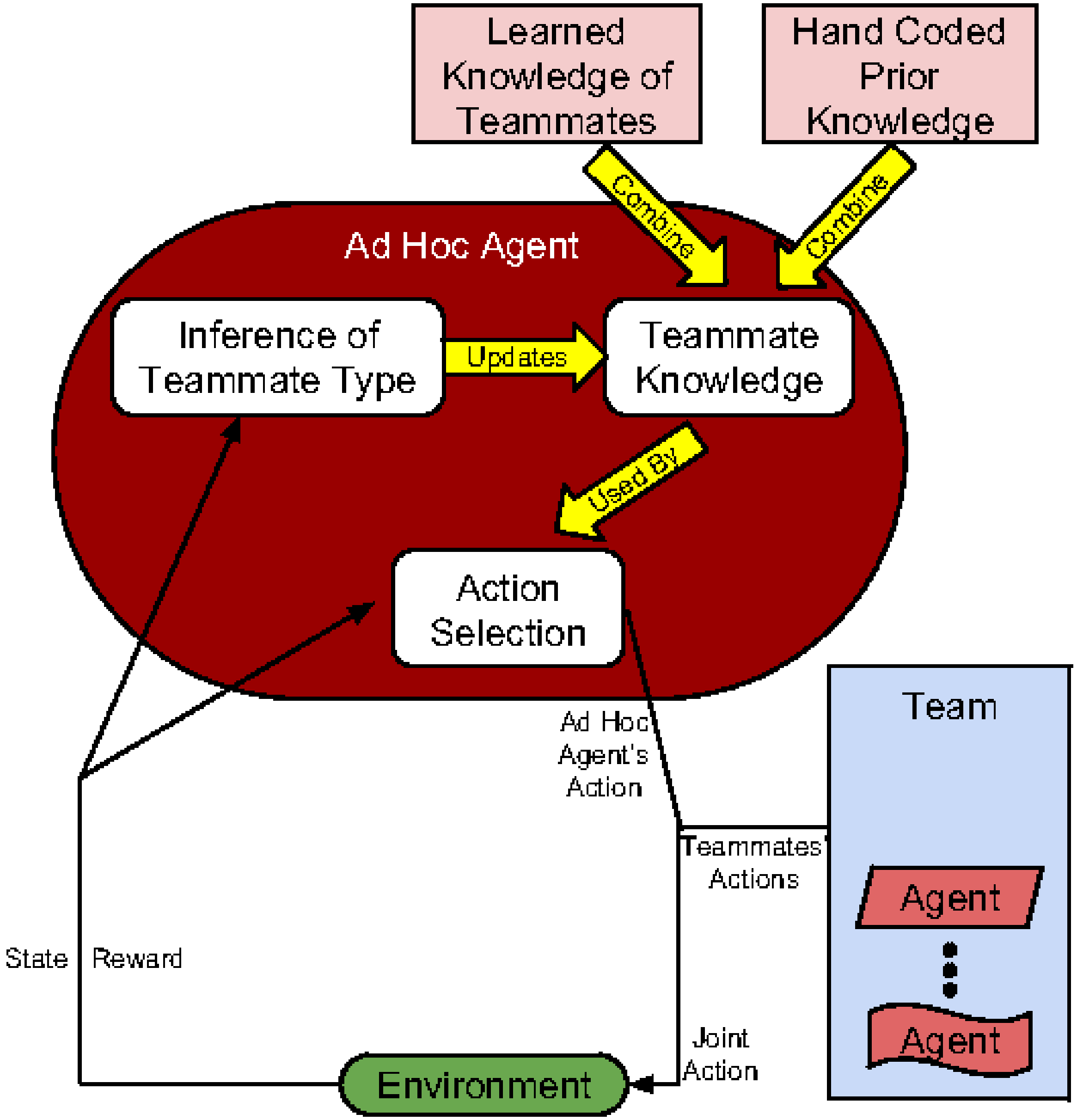
Ad Hoc Teamwork [Stone et al., AIJ 2013]

- Only in control of a single agent or subset of agents
- Unknown teammates
- Shared goals
- No pre-coordination

Examples in humans:

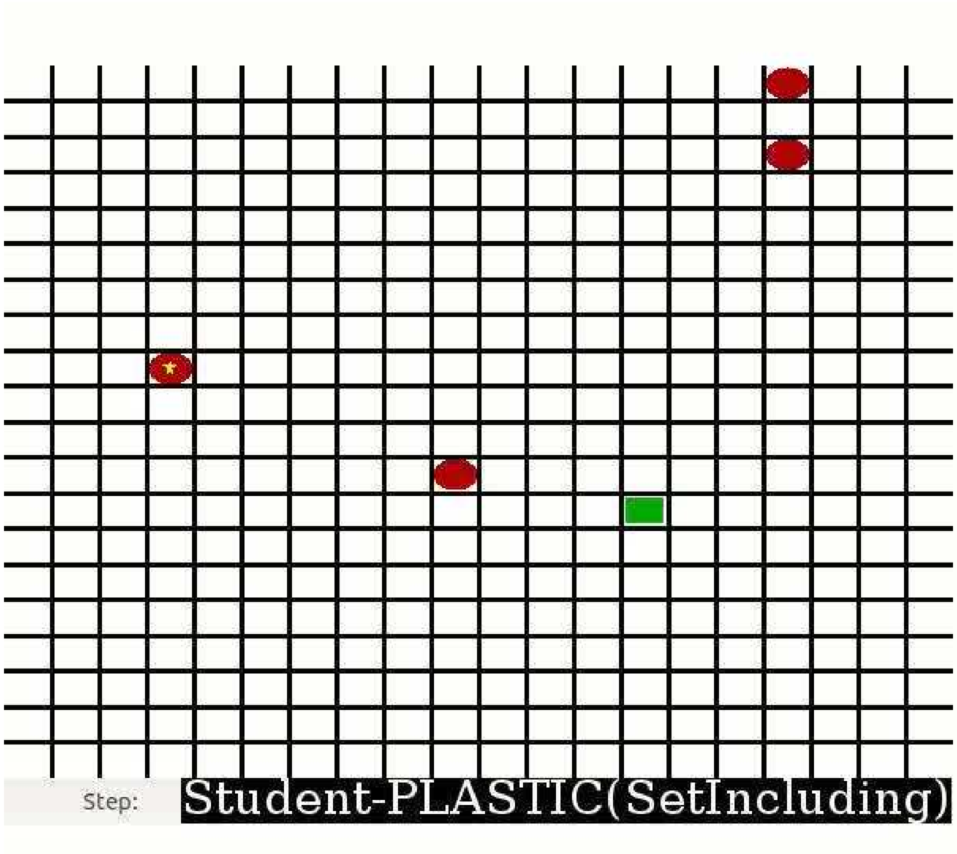
- Pick up soccer
- Accident response

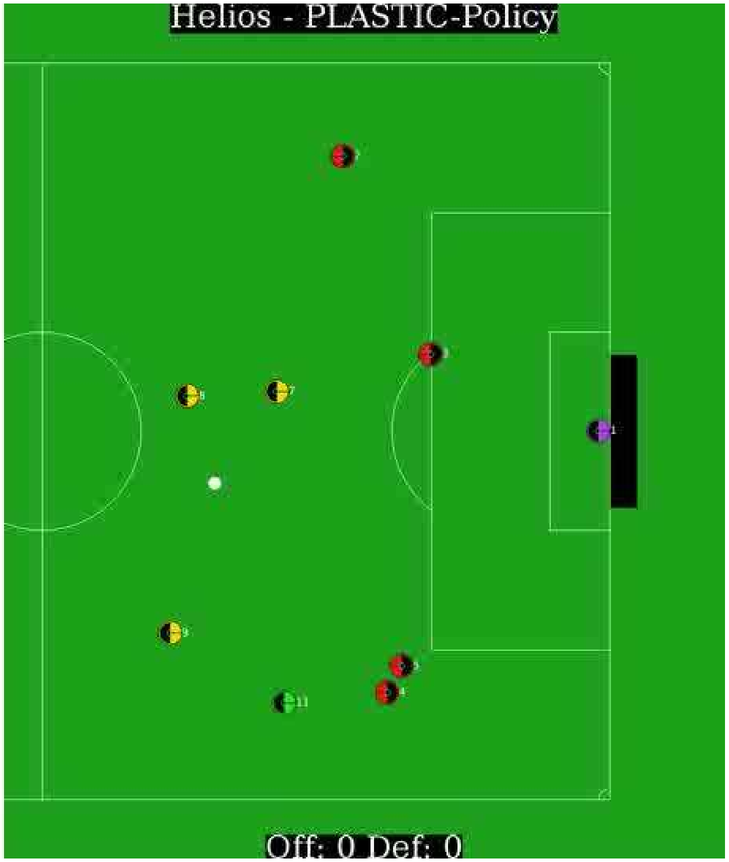
PLASTIC: Planning and Learning to Adapt Swiftly to Teammates to Improve Cooperation



Testbed Domains

- Agent replaces single teammate in otherwise coherent team
- Adapts based on knowledge learned from previous teammates





Other Ad Hoc Teamwork

- Introduced as AAAI Challenge Problem [Stone et al. '10]

Other Ad Hoc Teamwork

- Introduced as AAAI Challenge Problem [Stone et al. ’10]
- Theory: repeated games, bandits [Stone et al., ’11]

Other Ad Hoc Teamwork

- Introduced as AAAI Challenge Problem [Stone et al. '10]
- Theory: repeated games, bandits [Stone et al., '11]
- Experiments: **flocking** [Genter & Stone, '12]

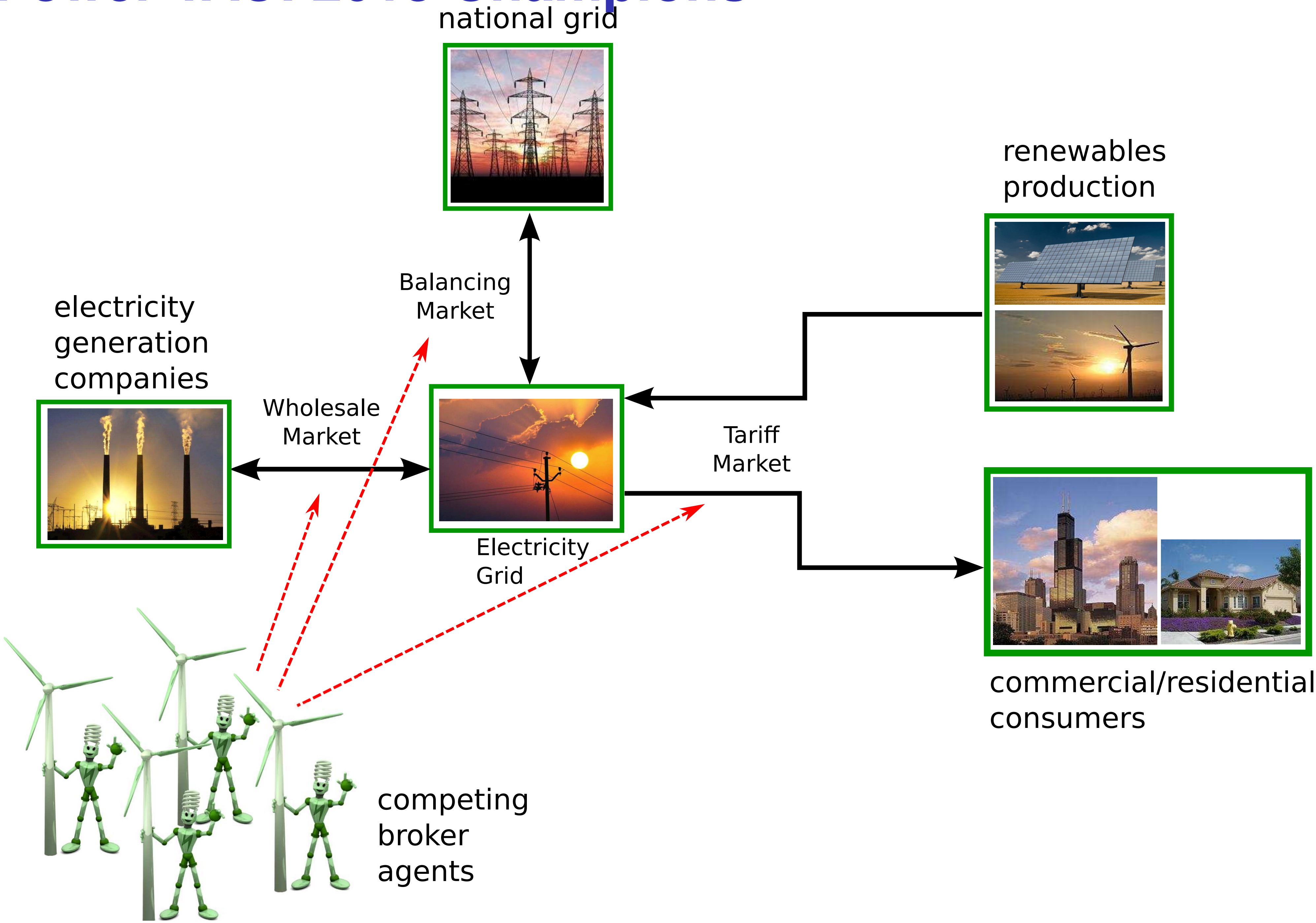
Other Ad Hoc Teamwork

- Introduced as AAAI Challenge Problem [Stone et al. '10]
- Theory: repeated games, bandits [Stone et al., '11]
- Experiments: **flocking** [Genter & Stone, '12]
- **RoboCup experiments** [Genter et al., '15]

Other Ad Hoc Teamwork

- Introduced as AAAI Challenge Problem [Stone et al. '10]
- Theory: repeated games, bandits [Stone et al., '11]
- Experiments: **flocking** [Genter & Stone, '12]
- **RoboCup experiments** [Genter et al., '15]
- AAAI Workshops, JAAMAS special issue

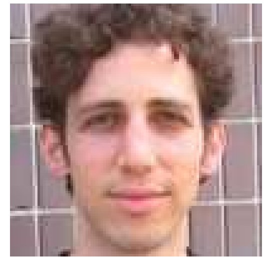
Power TAC: 2013 Champions



Acknowledgements

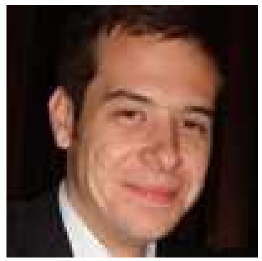
Nicholas Jong

Matthew Taylor



Shimon Whiteson

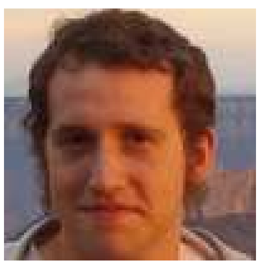
Doran Chakraborty



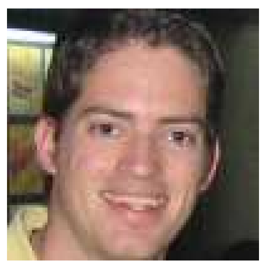
Todd Hester

Shivaram Kalyanakrishnan

Brad Knox



Samuel Barrett



Patrick MacAlpine

Daniel Urieli

Katie Genter

LARG (past and present), NSF, ONR, FHWA, DARPA, IBM, Google