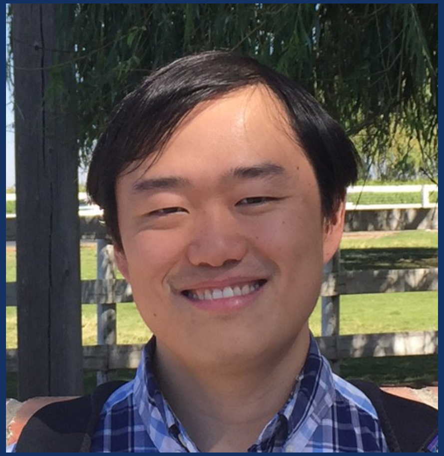
Towards Principled Approaches for Empirical Problems

Yuandong Tian

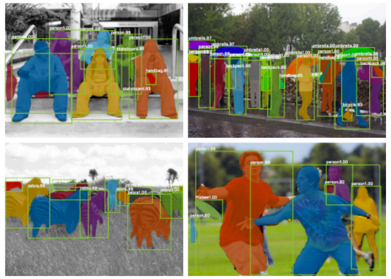
Research Scientist and Manager

Facebook AI Research

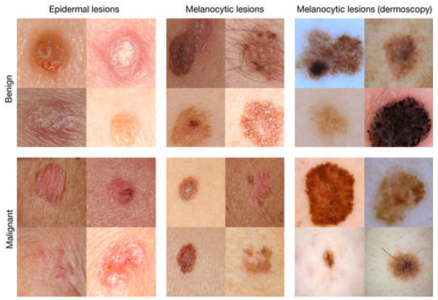


facebook Artificial Intelligence

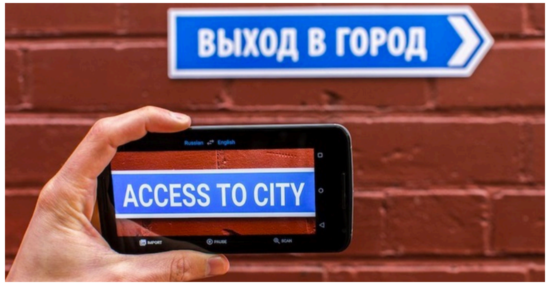
Great Empirical Success of AI



Object Recognition



Medical



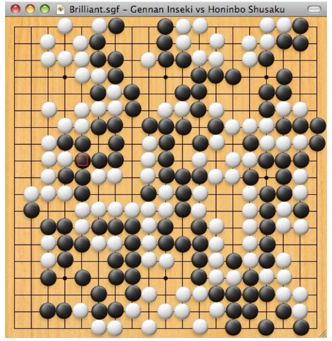
Translation

Speech Recognition

Personalization

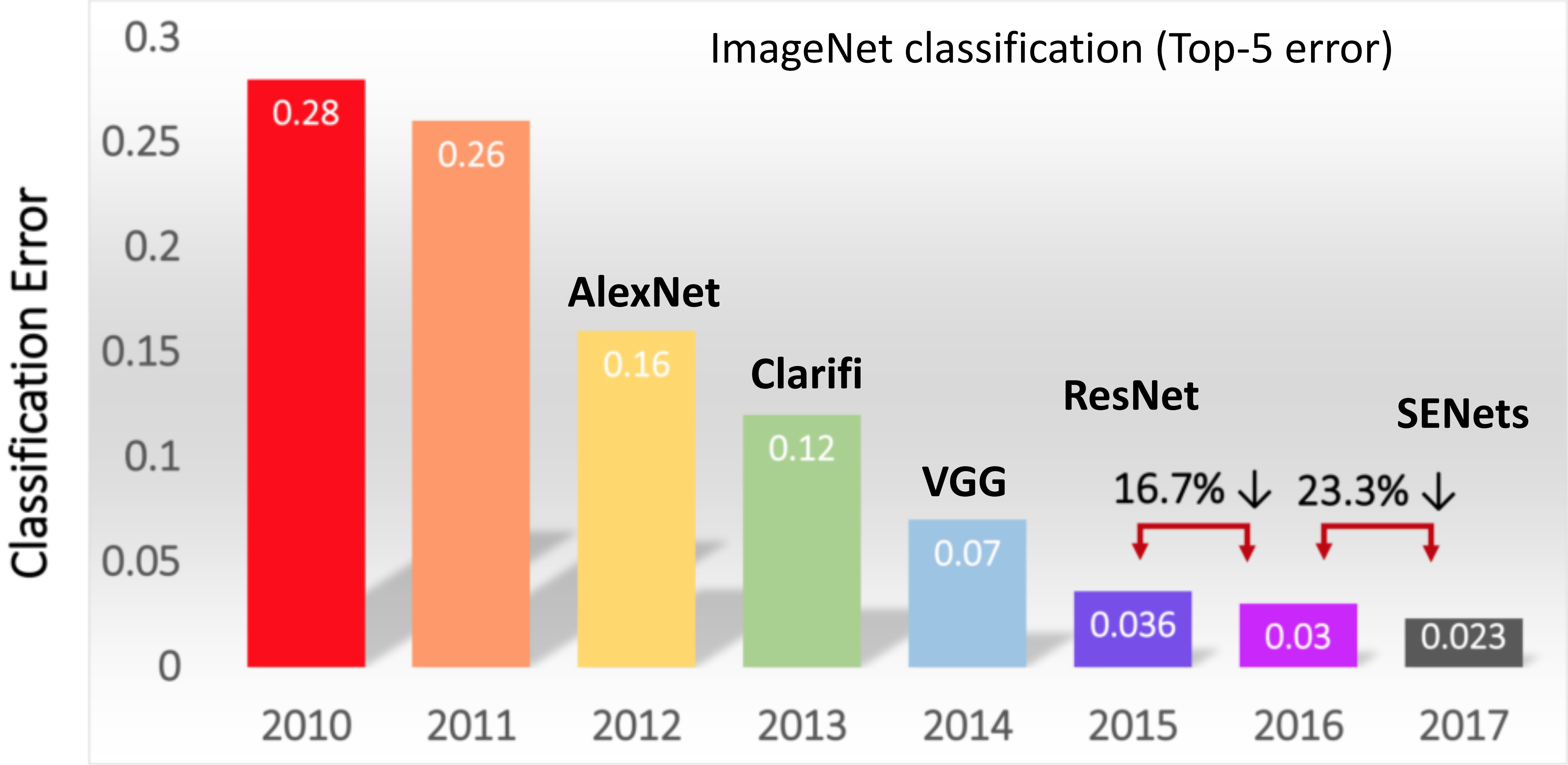
Surveillance

Smart Design



Board game

Great Empirical Success of AI



https://www.kaggle.com/getting-started/149448

Great Empirical Success of AI

AlphaGo (2016)

Chess

Shogi

Dota 2

StarCraft 2

Great Empirical Success of AI

Initialized by Human

The importance of being on twitter

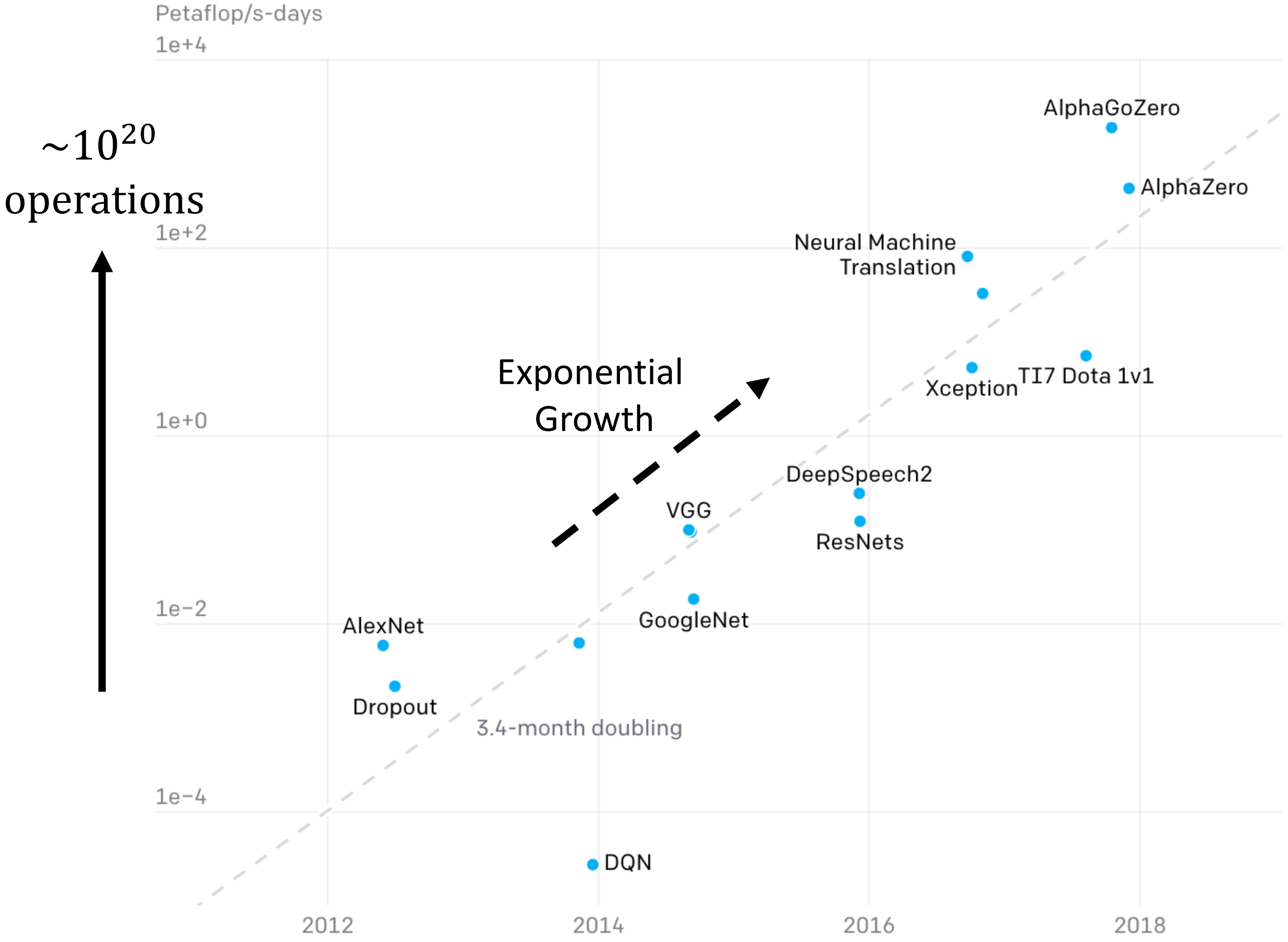
by Jerome K. Jerome
London, Summer 1897

It is a curious fact that the last remaining form of social life in which the people of London are still interested is Twitter. I was struck with this curious fact when I went on one of my periodical holidays to the sea-side, and found the whole place twittering like a starling-cage. I called it an anomaly, and it is.

I spoke to the sexton, whose cottage, like all sexton's cottages, is full of antiquities and interesting relics of former centuries. I said to him, "My dear sexton, what does all this twittering mean?" And he replied, "Why, sir, of course it means Twitter." "Ah!" I said, "I know about that. But what is Twitter?"

https://twitter.com/quasimondo/status/1284509525500989445

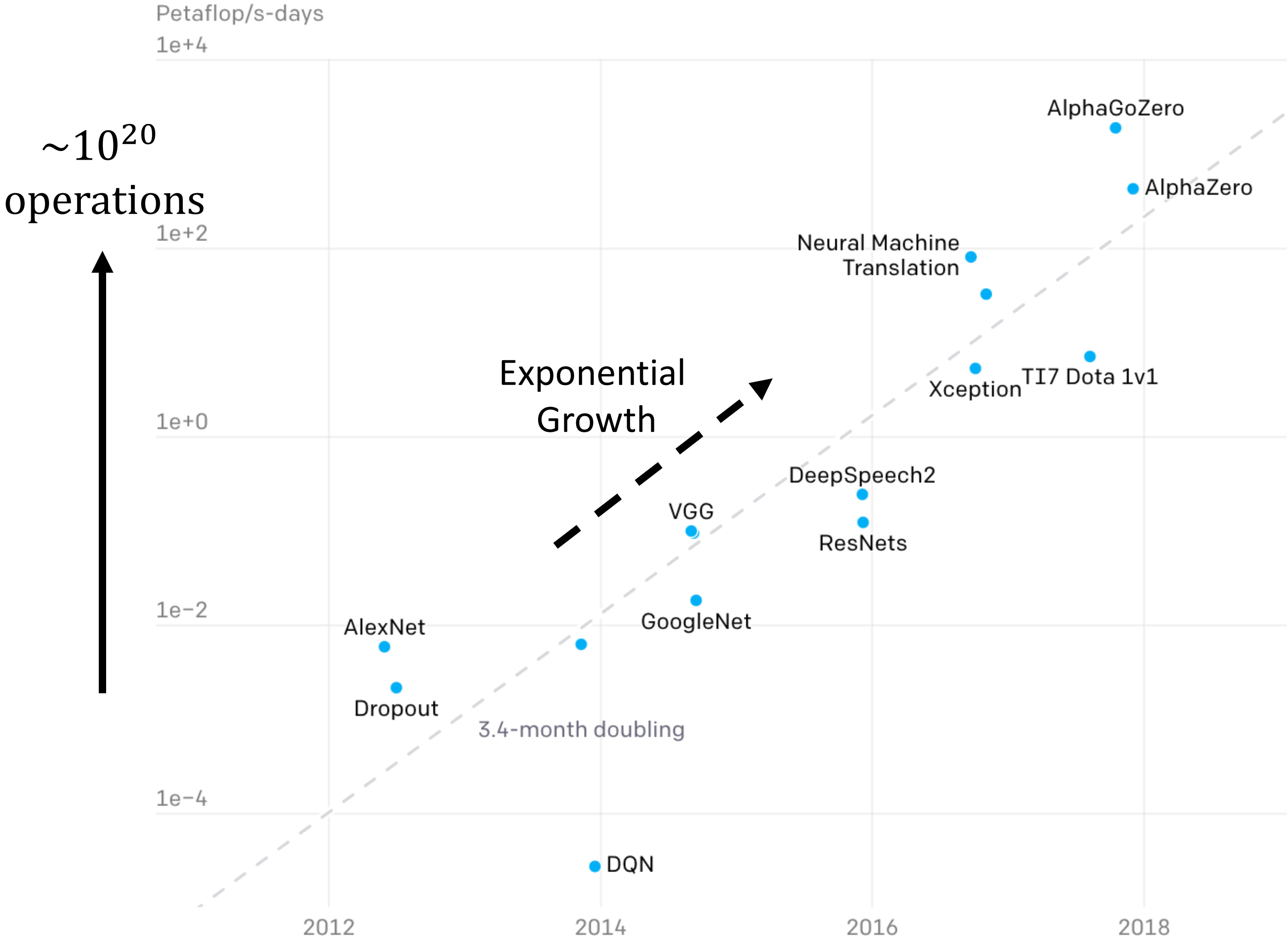
Will this trend continue?



facebook Artificial Intelligence

https://openai.com/blog/ai-and-compute/

Will this trend continue?



"

At first, I cannot do parameter sweeping

Then I cannot train the model

Then I cannot do fine-tuning

Then I cannot run one forward pass

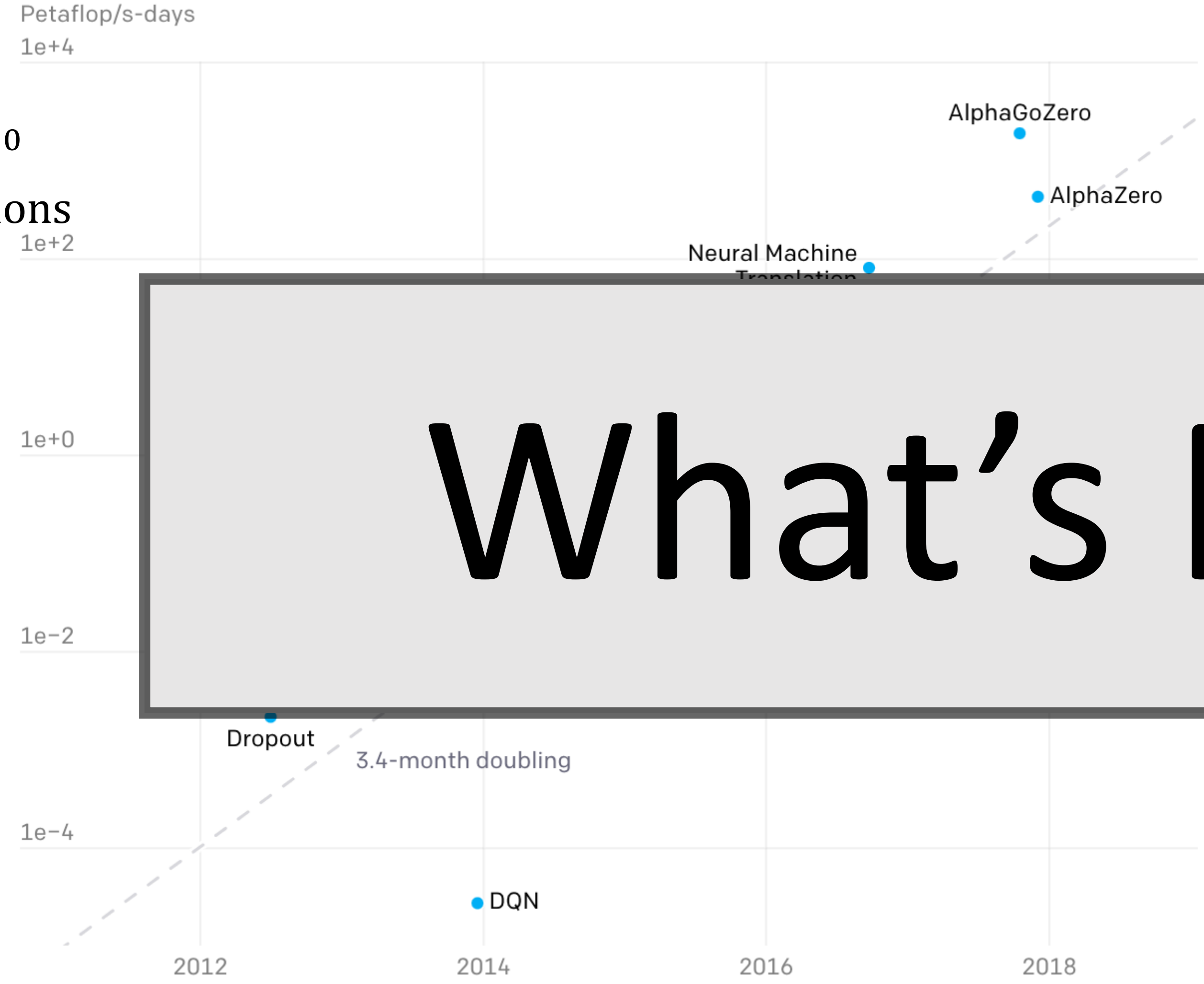
Then I cannot even download the model

…
"

https://openai.com/blog/ai-and-compute/

Will this trend continue?

$\sim 10^{20}$ operations



What’s Next?

“

At first, I cannot do parameter sweeping

ss

model

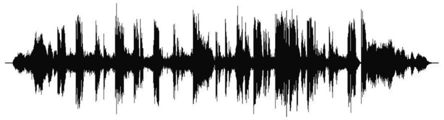
…

”

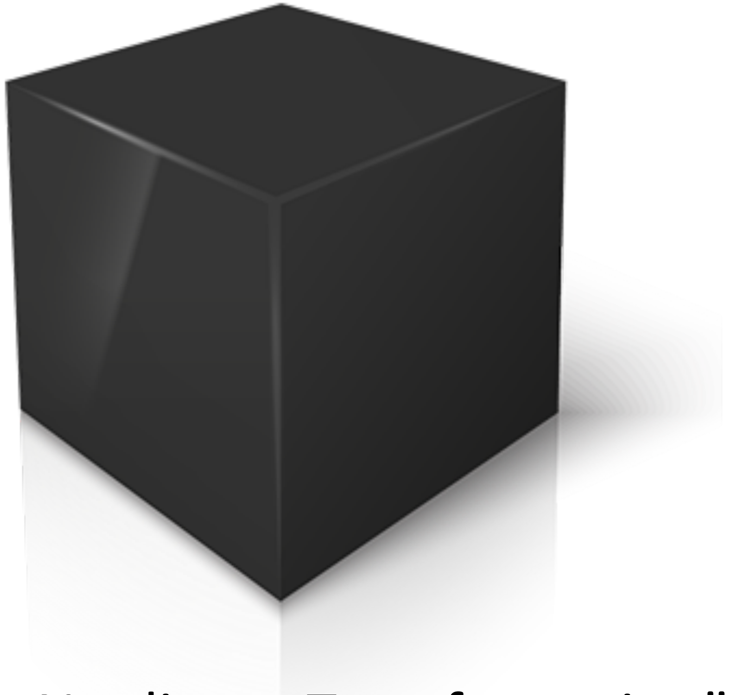
https://openai.com/blog/ai-and-compute/

Is Black-box Model Enough?

Input



This is an apple

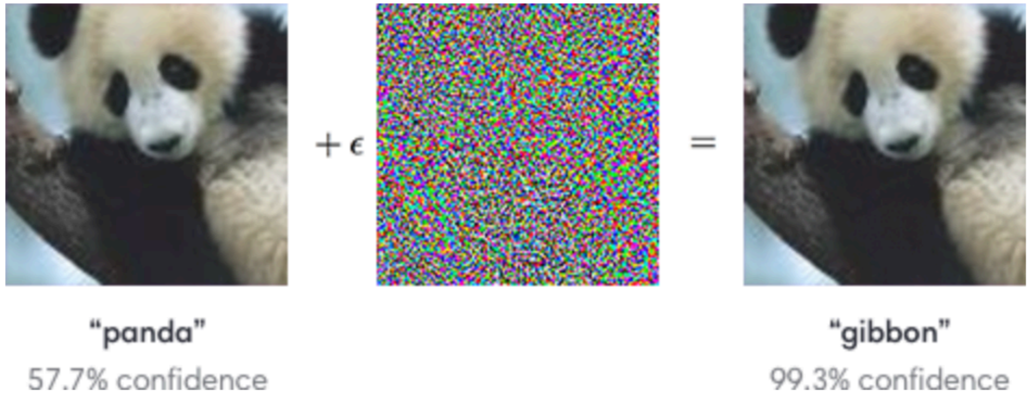


"Some Nonlinear Transformation"

Output

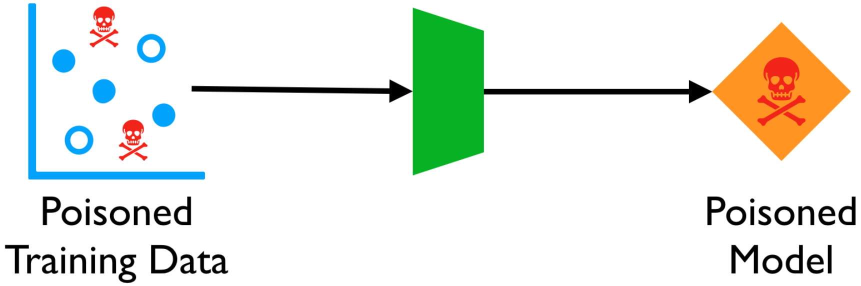
Using Black-box Model is tricky

Adversarial samples



Stop sign → a 45 mph sign

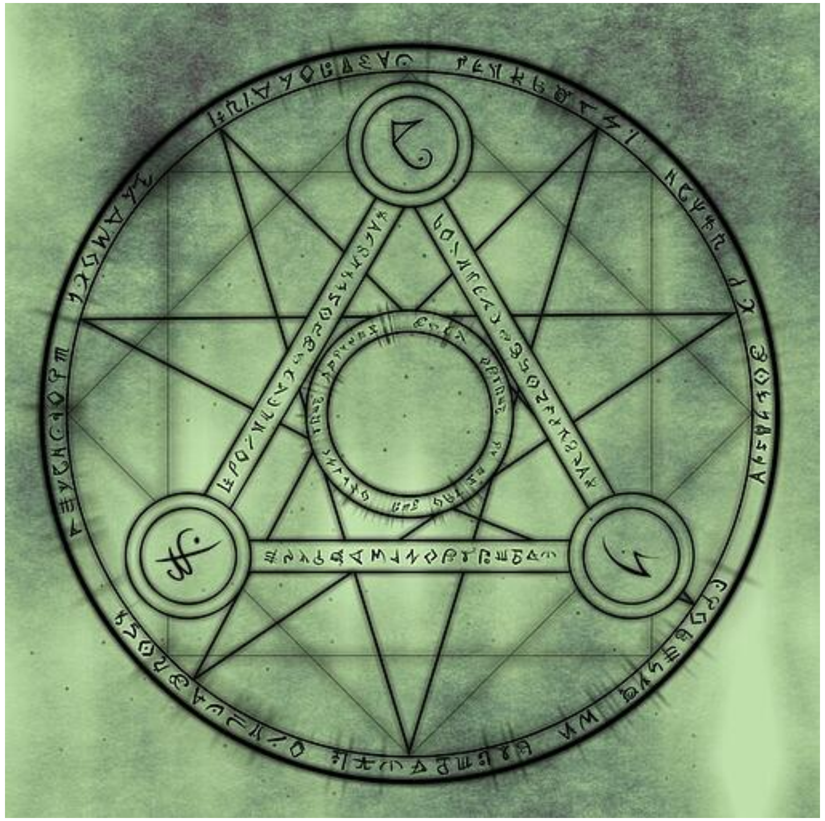
Data Poisoning



Interpretability

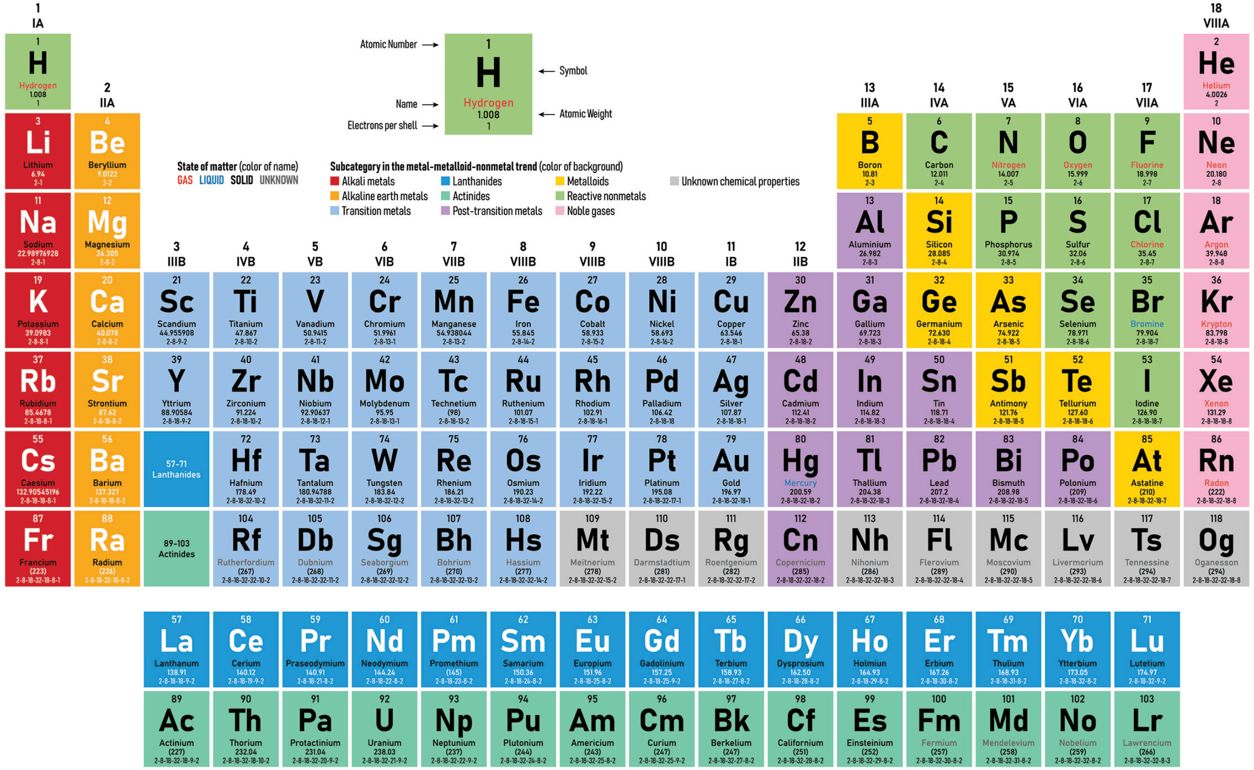
D. Blau, Network Dissection: Quantifying Interpretability of Deep Visual Representations, CVPR 2017

Let’s Check the History



Alchemy

Periodic Table of the Elements



Chemistry

The Black Powder

$$2KNO_3 + S + 3C \rightarrow K_2S + N_2 \uparrow + 3CO_2 \uparrow$$

(硝酸钾) KNO_3

(硫磺) S

(木炭) C

2 mol : 1 mol : 3 mol

Best mass ratio. 74.64% : 11.85% : 13.51%

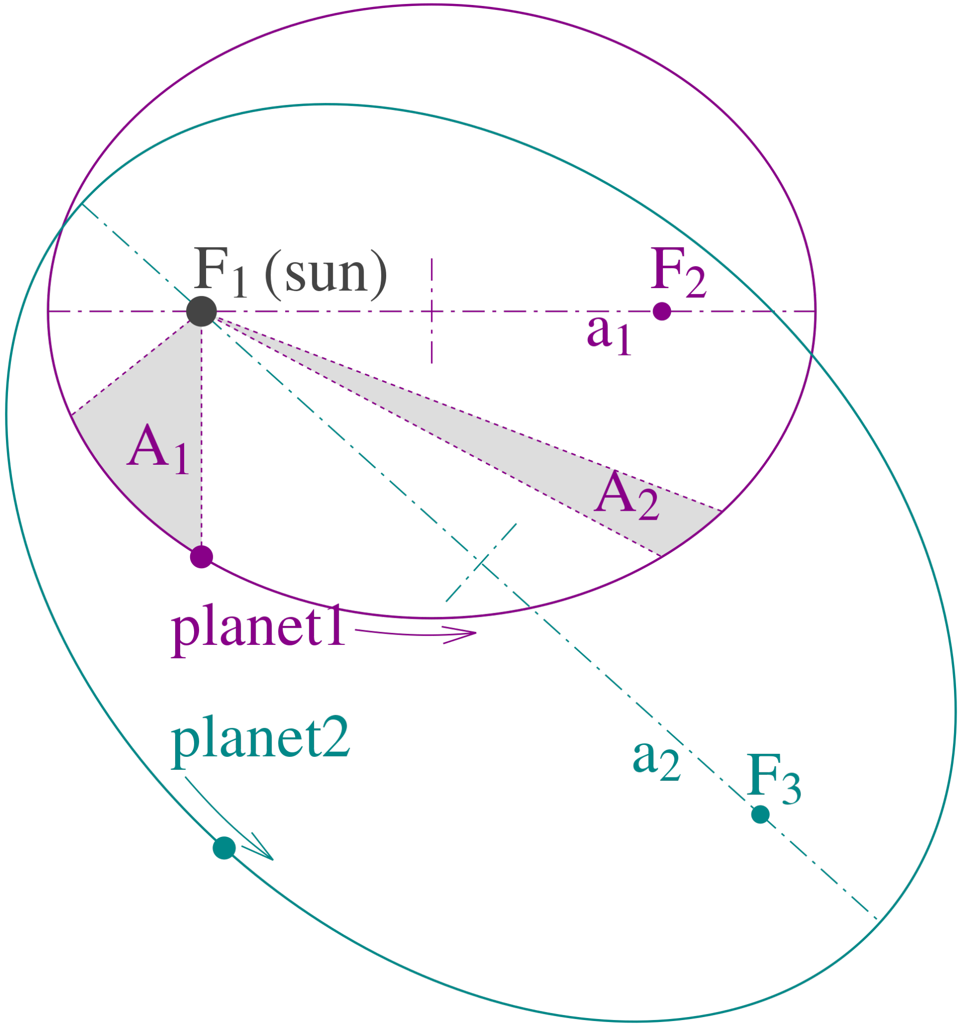
Black Powder Ratio in the History

	KNO_3	S	C
Song Dynasty (1044 AD)	50%	25%	25%
Early Ming Dynasty (~1400 AD)	71.4%	14.3%	14.3%
Mid Ming Dynasty (~1550 AD)	75.8%	10.6%	13.6%
Qing Dynasty (1753 AD)	80%	10.51%	9.88%
Qing Dynasty (1818 AD)	77.8%	9.7%	12.5%
Qing Dynasty (1839 AD)	74%	11%	15%
Current Standard	**75%**	**10%**	**15%**

Human
Parameter Tuning

胡建中 《清代火炮》

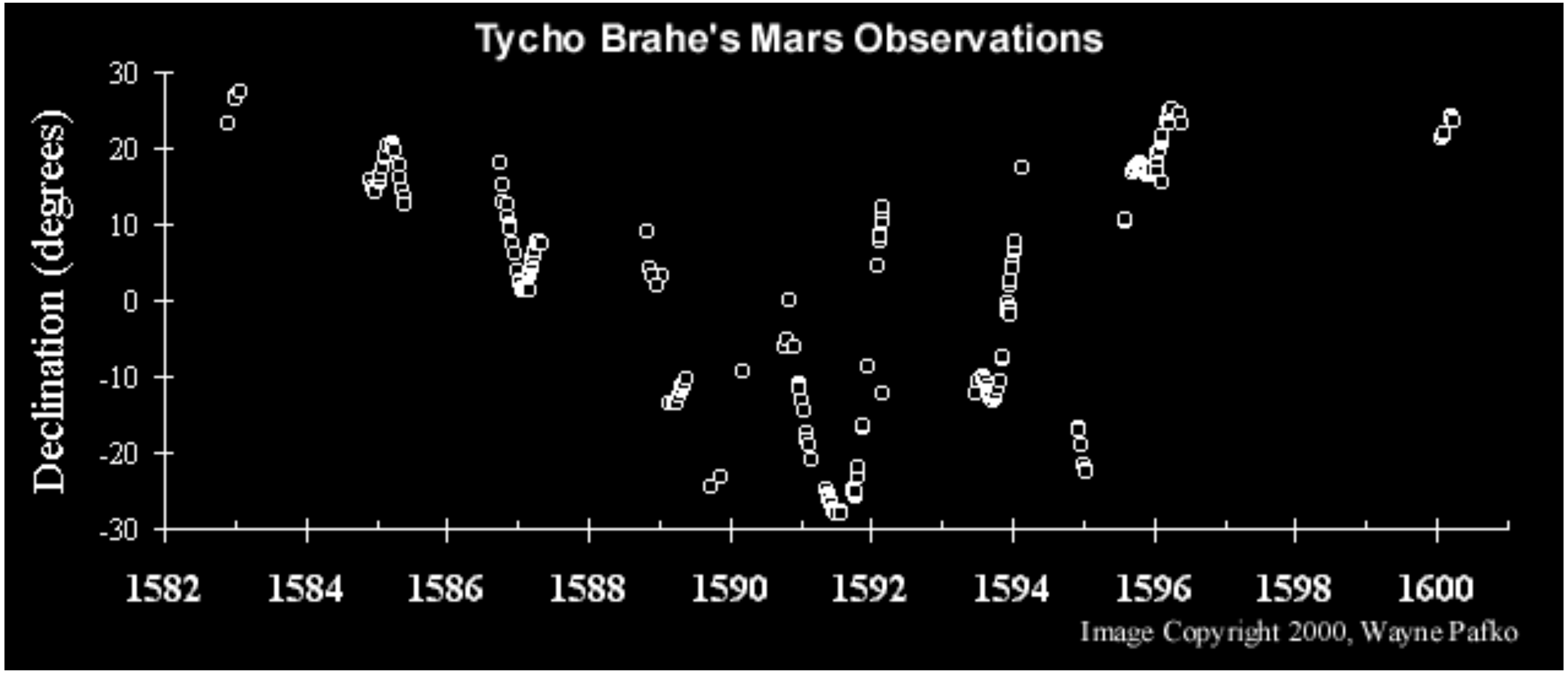
Kepler's laws of planetary motion



Johannes Kepler （开普勒）

https://en.wikipedia.org/wiki/Kepler%27s_laws_of_planetary_motion

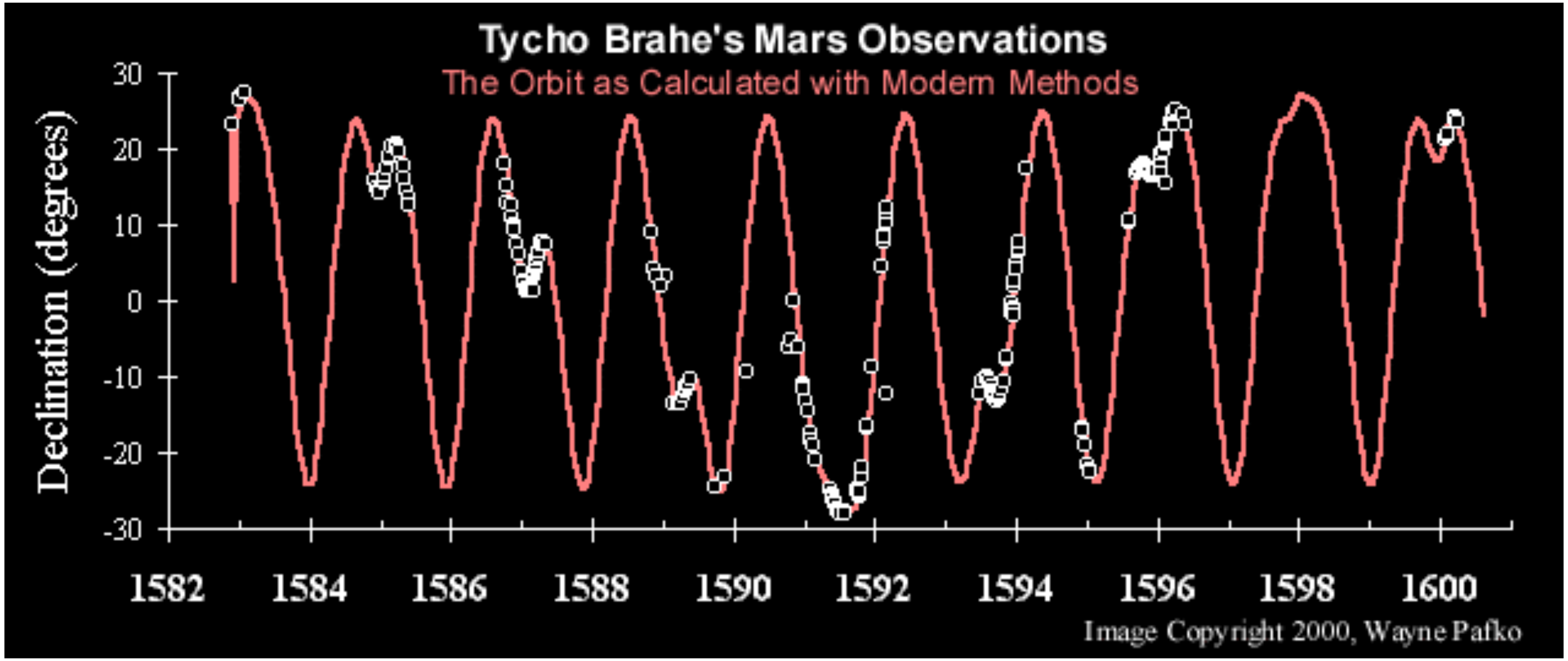
Tycho Brahe’s Mars Observations



Tycho Brahe （第谷）

How many curves can you fit with modern machine learning?

Tycho Brahe's Mars Observations

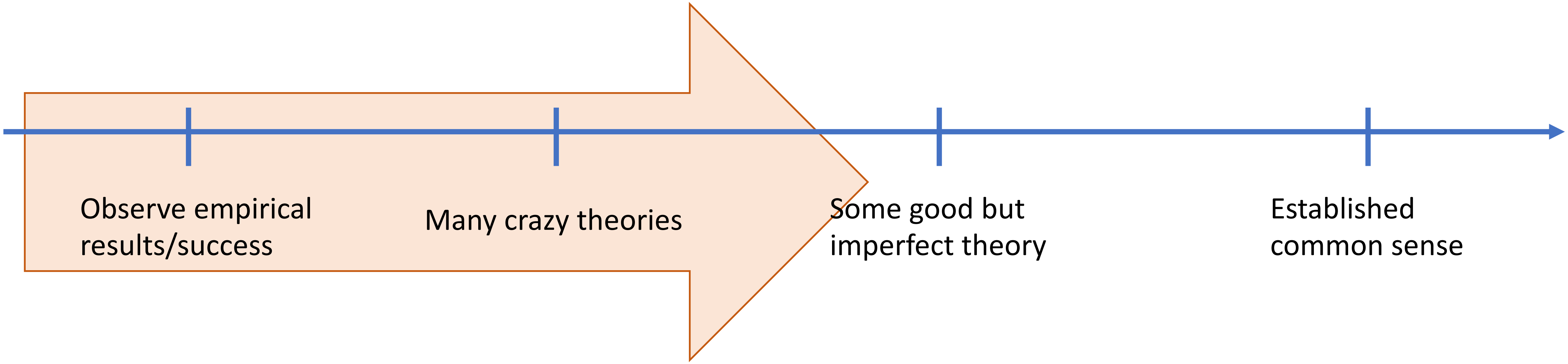


Tycho Brahe （第谷）

The true curve computed from the modern methods

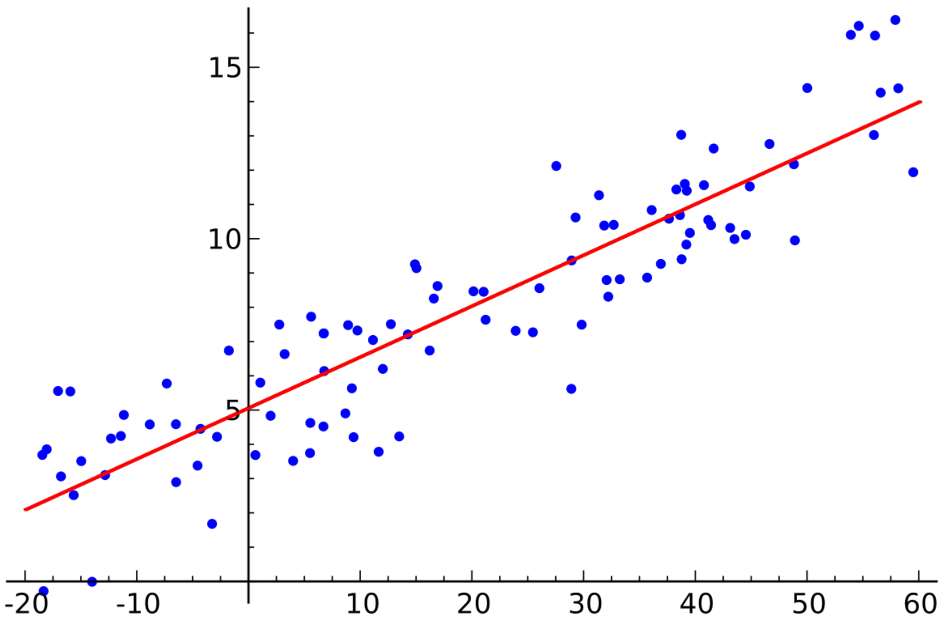
http://www.pafko.com/tycho/observe.html

Will History Repeat Itself?

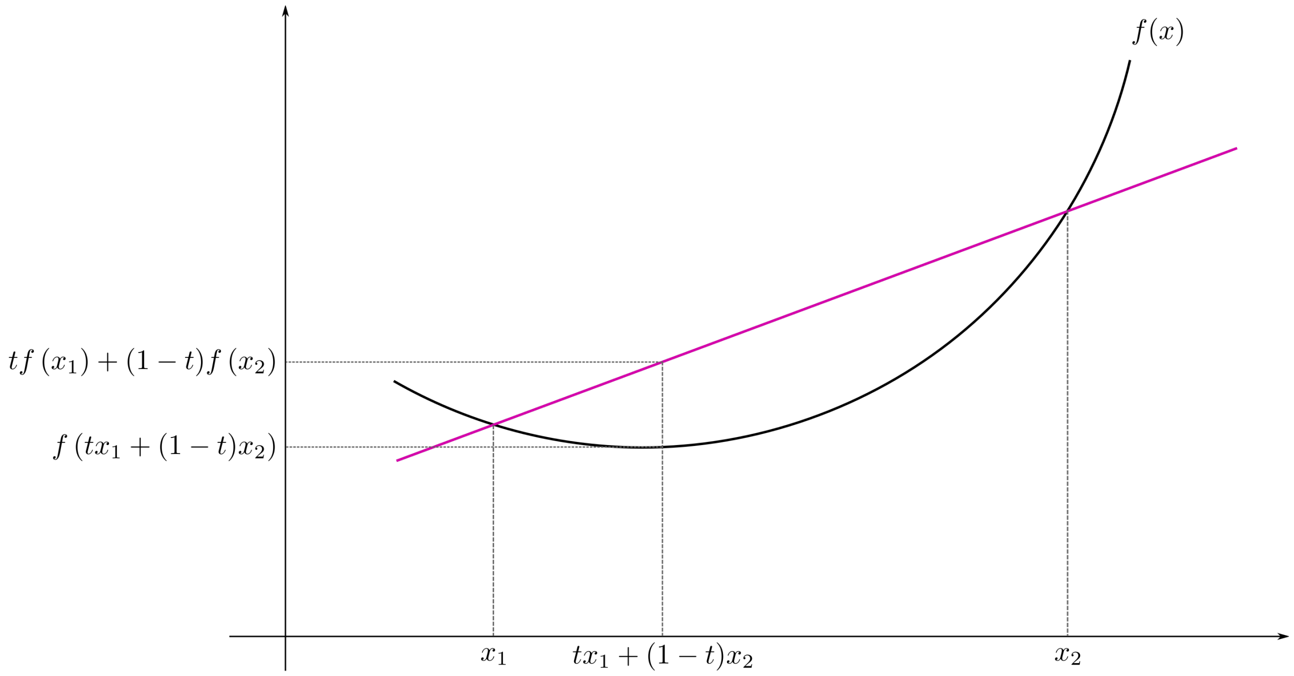


Where are we now?

Theory that matches with Practice

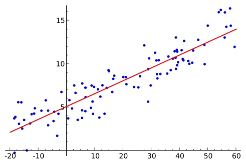


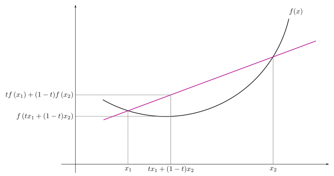
Linear Regression

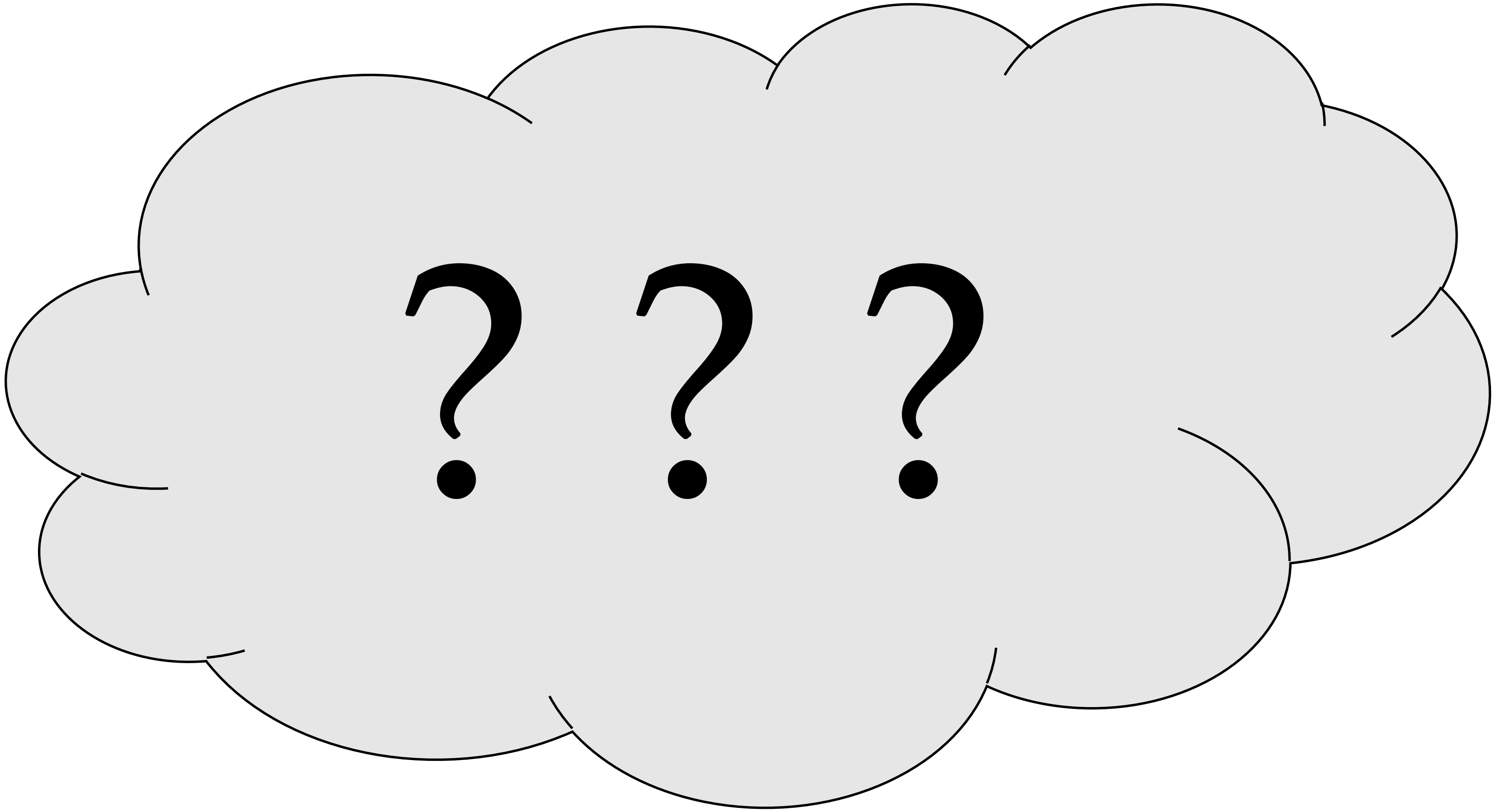


Convex Function

Theory that **doesn't match** with Practice



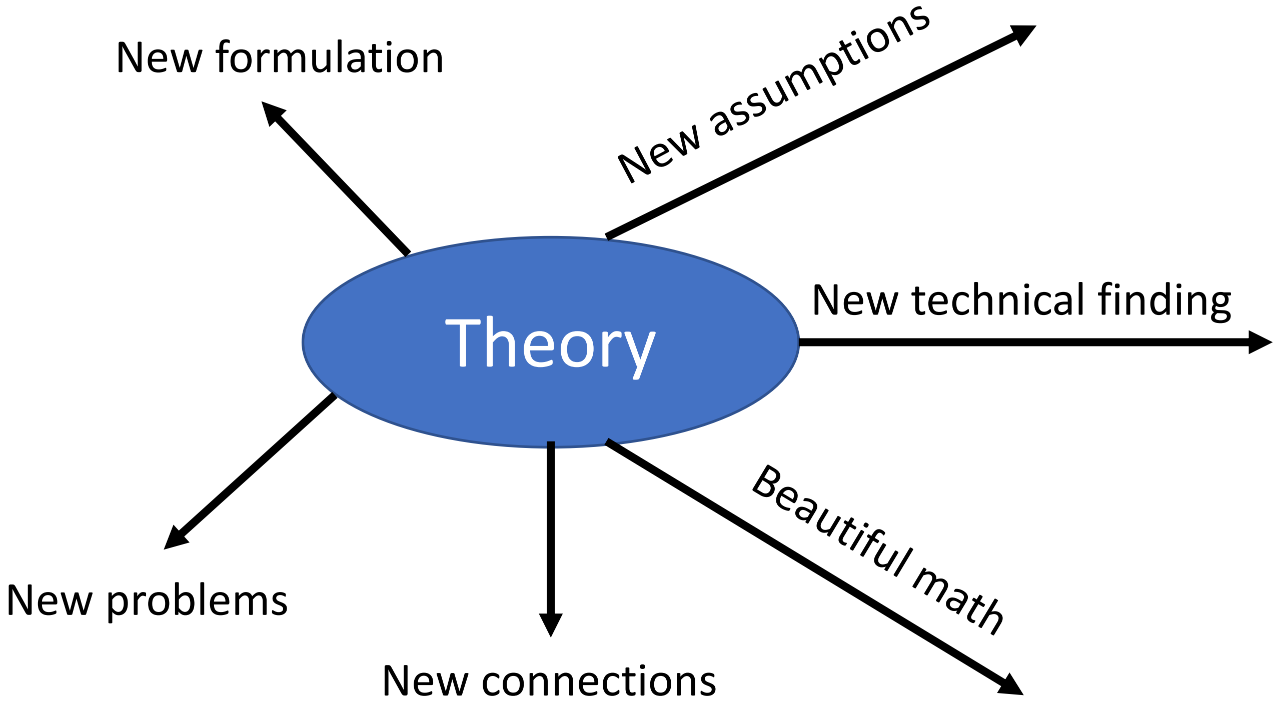




How can we move forward?

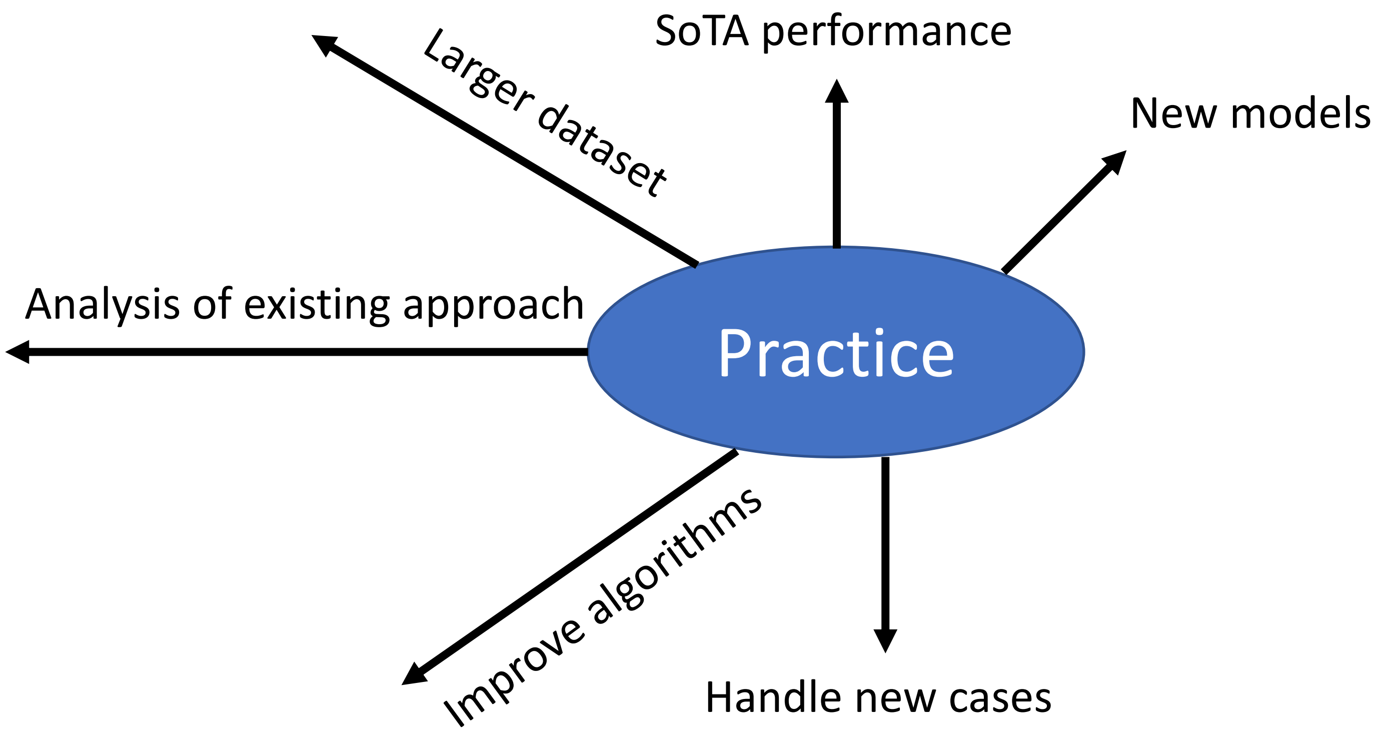
Theory and Practice

How to develop theory?



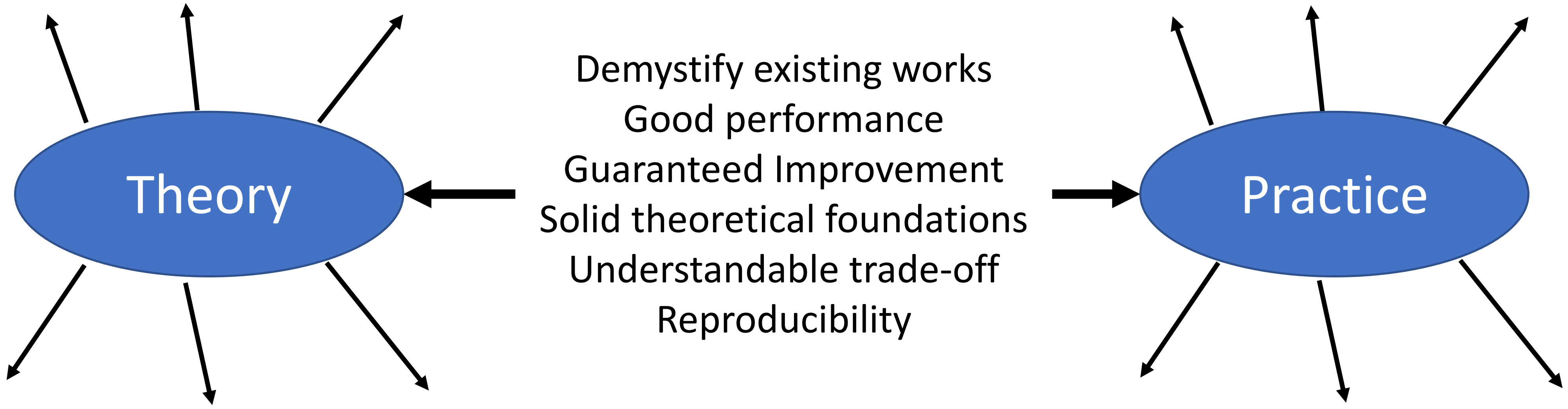
Theory and Practice

How to develop empirical work?



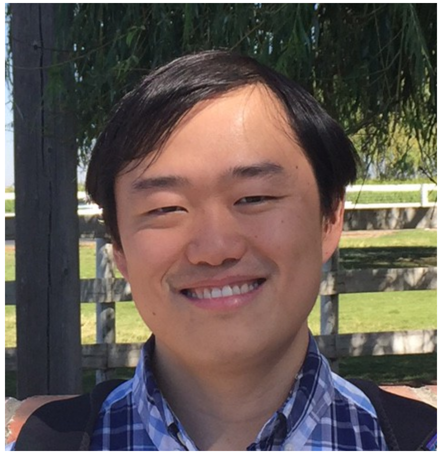
Theory and Practice

The best research work we could imagine:



Super Hard … But that’s the way to go!

Career Path



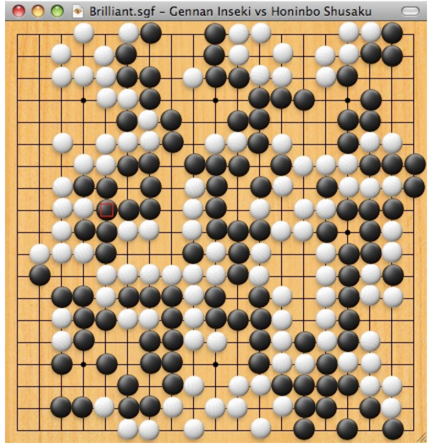
Theoretical Understanding of Models and Algorithms

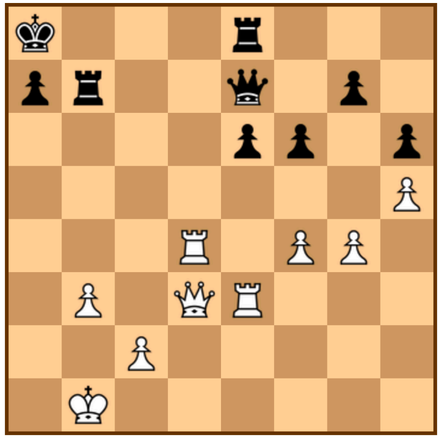
Computer Vision | **Reinforcement Learning**

2006 — 2008 — 2013 — 2015 — 2020

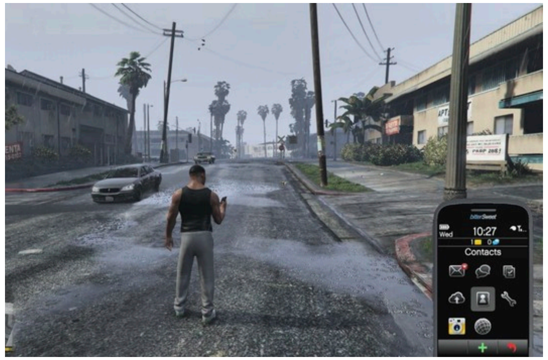
PhD | **Waymo** | **Facebook AI Research**

The Charm of Games





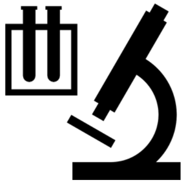
Complicated long-term strategies.



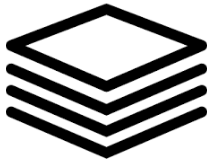
Realistic Worlds

Game as a Vehicle of AI

Infinite supply of *fully* labeled data

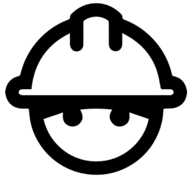


Controllable and replicable

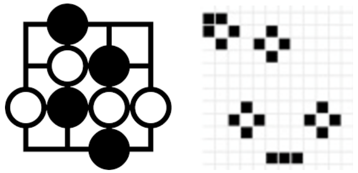


Low cost per sample

Faster than real-time



Less safety and ethical concerns



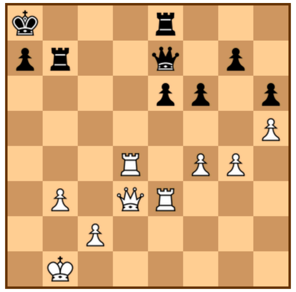
Complicated dynamics with simple rules.

How Game AI works

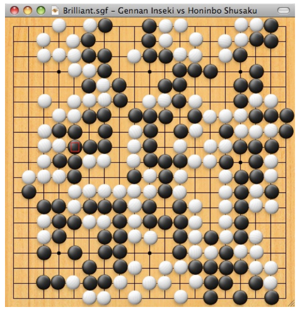
Even with a super-super computer,
it is not possible to search the entire space.

How Game AI works

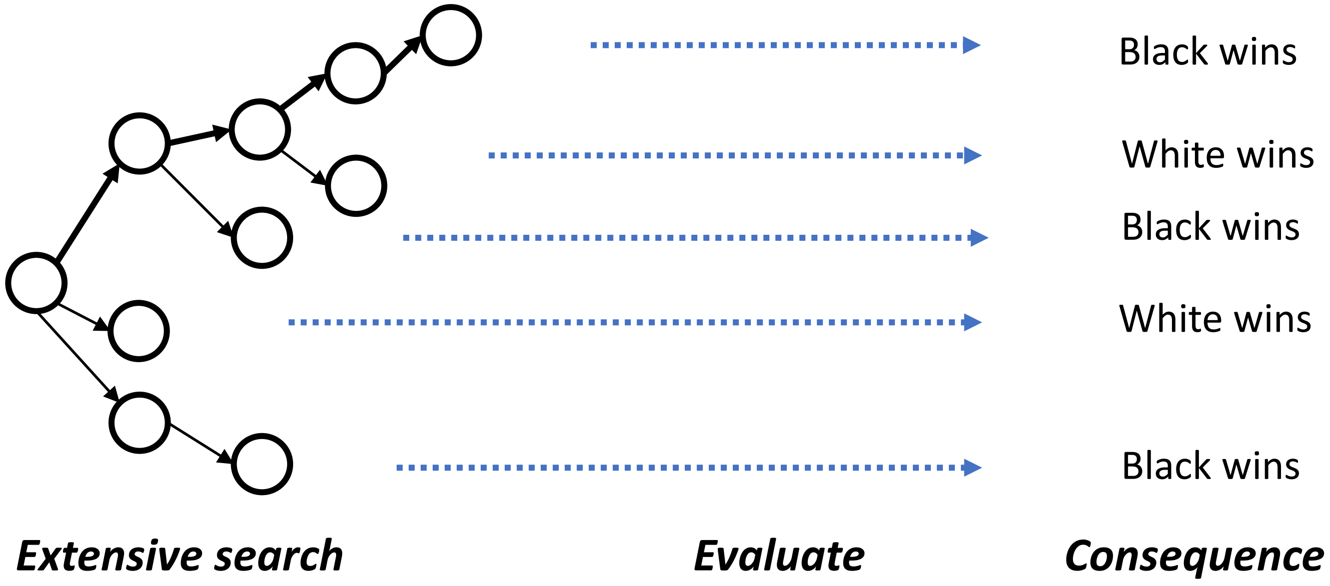
Even with a super-super computer,
it is not possible to search the entire space.



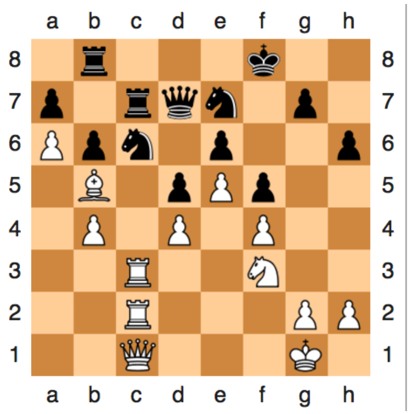
Lufei Ruan vs. Yifan Hou (2010)



Current game situation

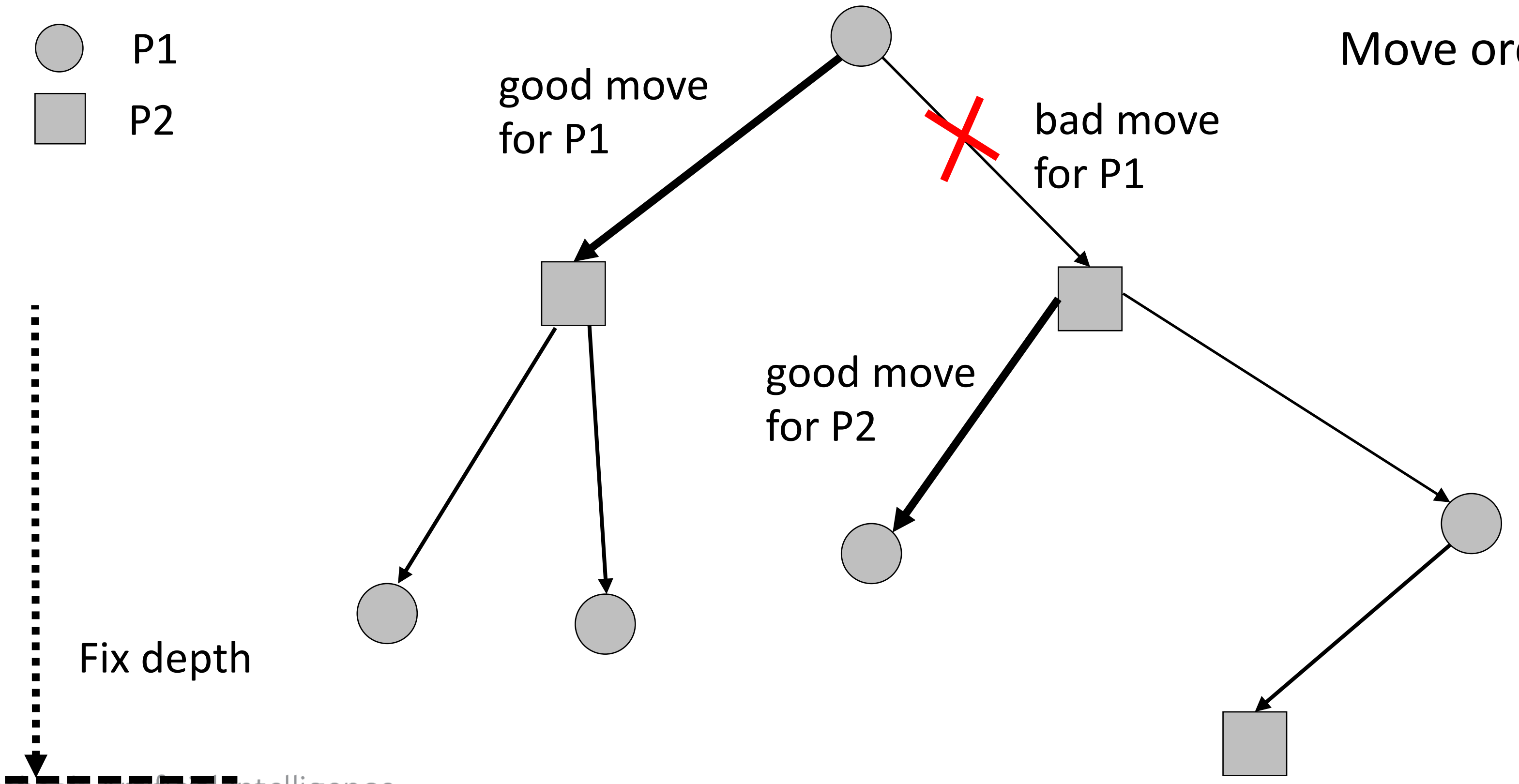


Alpha-beta Pruning

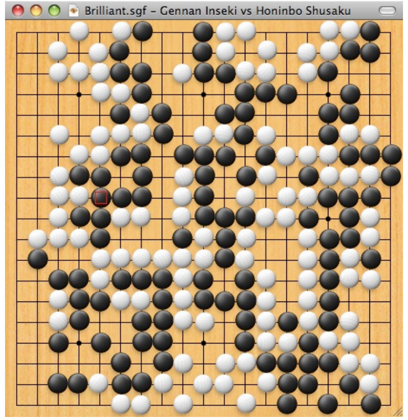


A good counter move eliminates other choices.

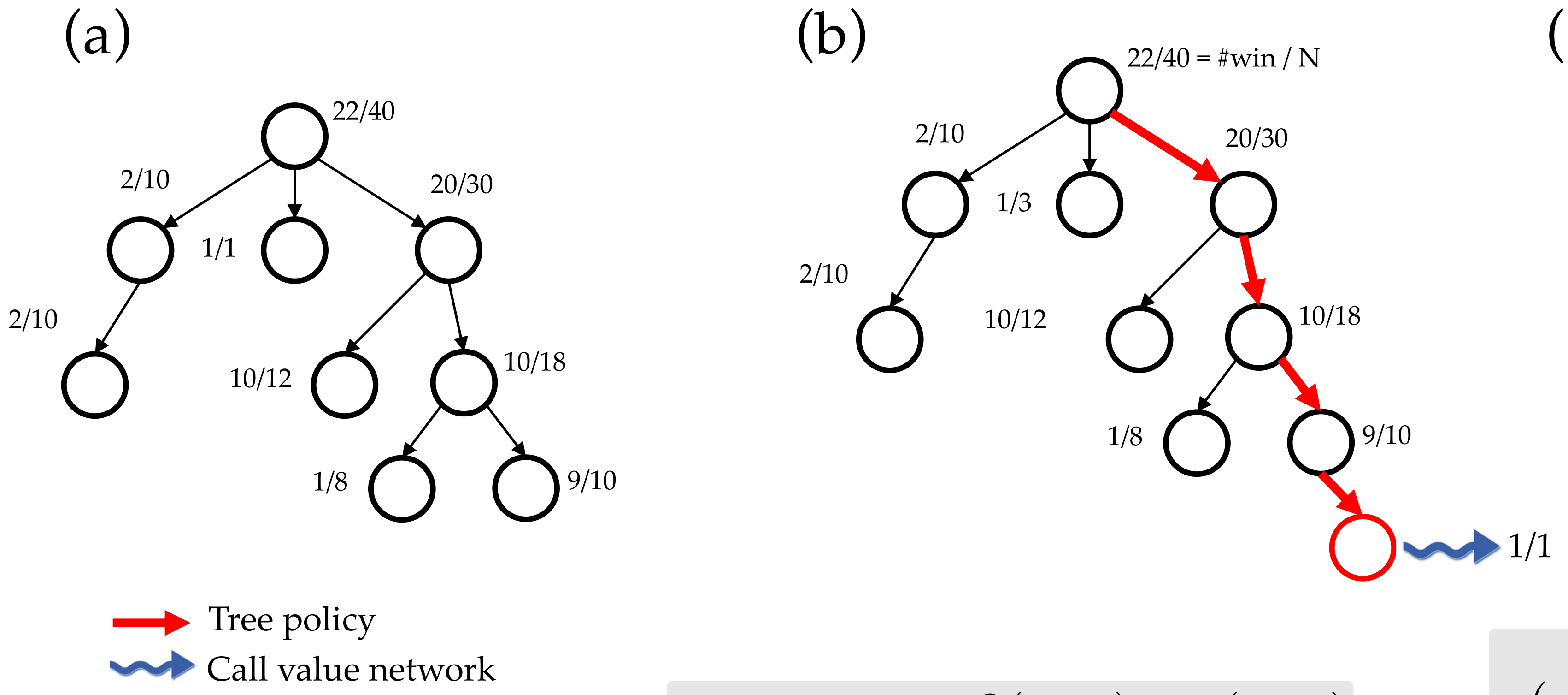
Move order is important!



Monte Carlo Tree Search



Aggregate win rates, and search towards the good nodes.



$$a_t = \arg\max_a Q(s_t, a) + u(s_t, a)$$

$$u(s,a) = c_{\text{puct}} P(s,a) \frac{\sqrt{\sum_b N(s,b)}}{1 + N(s,a)}$$

How to model Policy/Value function?

Non-smooth + high-dimensional
Sensitive to situations. One stone changes in Go leads to different game.

Traditional approach

- Many manual steps
- Conflicting parameters, not scalable.
- Need strong domain knowledge.

Deep Learning

- End-to-End training
  - Lots of data, less tuning.
- Minimal domain knowledge.
- Amazing performance

AlphaGo Series

AlphaGo Lee
(Mar. 2016)

AlphaGo Master
(May. 2017)

AlphaGo Zero
(Oct. 2017)

Without Human Knowledge

The Mystery

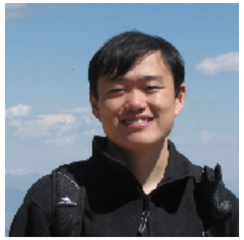
- Mystery
  - Is the proposed algorithm really universal?
  - Is the bot almighty? Is there any weakness in the trained bot?

- Lack of Ablation Studies
  - What factor is critical for the performance?
  - Is the algorithm robust to random initialization and changes of hyper parameters?
  - Any adversarial samples?

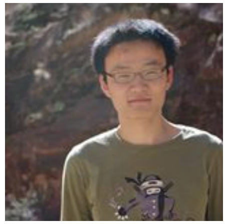
Impressive Results, No code, No model

Demystify existing empirical results
Good performance
Reproducibility

OpenGo project

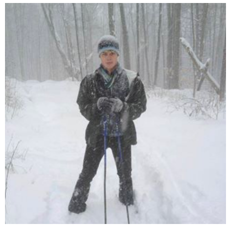

Yuandong Tian


Jerry Ma*


Qucheng Gong*


Shubho Sengupta*

Zhuoyuan Chen


James Pinkerton

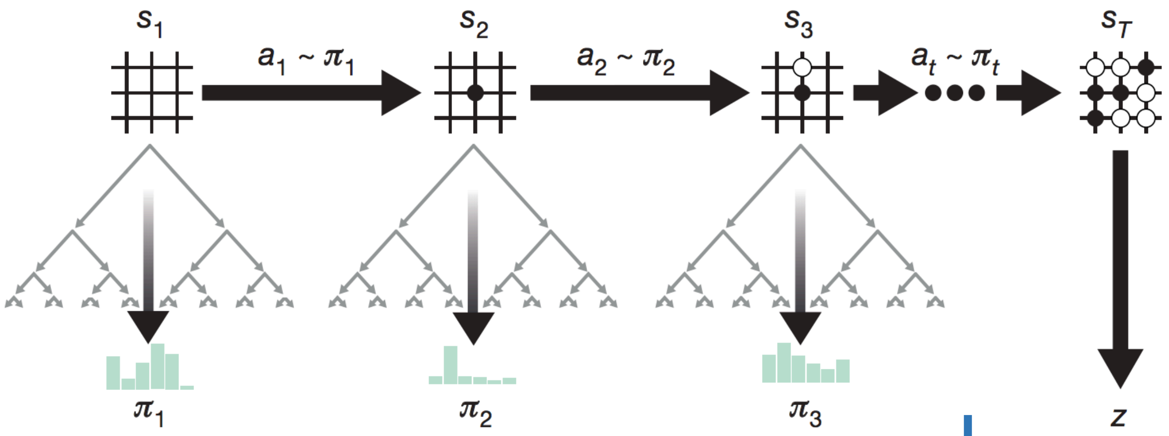

Larry Zitnick

*Equal Contributions

[Y. Tian et al., ELF OpenGo: An Analysis and Open Reimplementation of AlphaZero, ICML 2019]

AlphaGoZero / AlphaZero

Generate Training data



Without human knowledge

θ_i

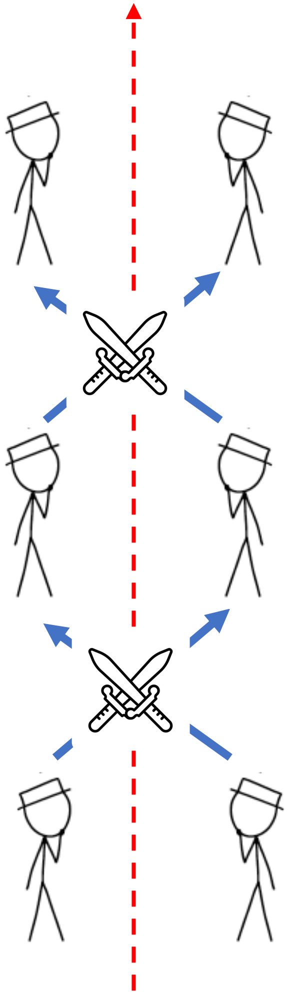
Update Models

Self-Replays

[Silver et al, Mastering the game of Go without human knowledge, Nature 2017]

The idea of Self-Play

Stroke left and right （左右互搏）



Generate Self-play Games

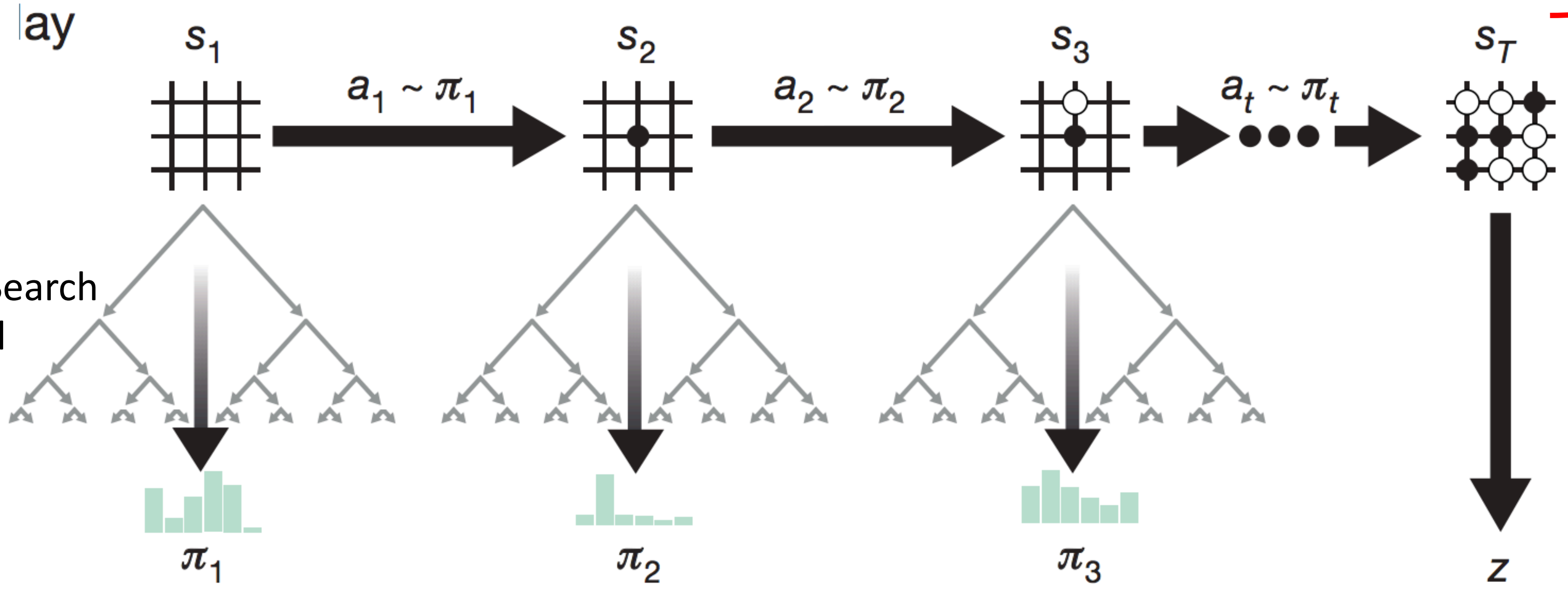
(s_1, π_1, z)

(s_2, π_2, z)

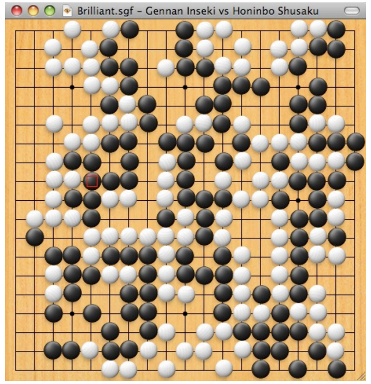
(s_3, π_3, z)

Training samples for θ_{i+1}

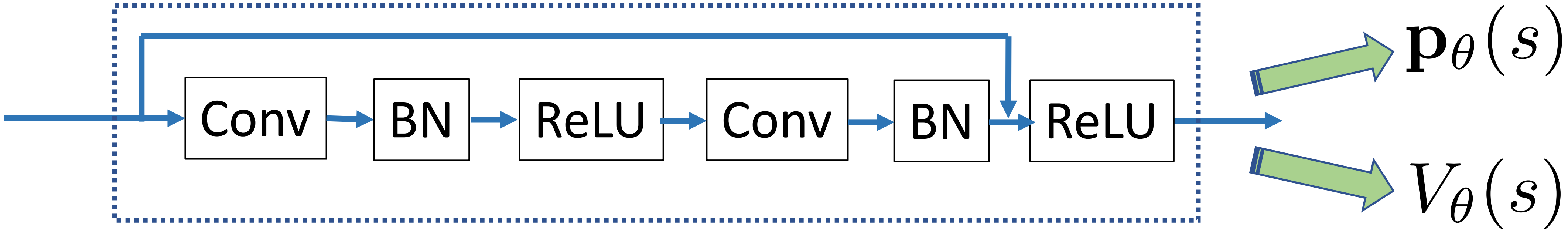
Monte Carlo Tree Search with current model θ_i



Update Models



$\boldsymbol{s}$



Player situation at time 0

Opponent situation at time 0

Player situation at t=-7

Color to play

Input features (19x19x17): $(X, Y, X_{-1}, Y_{-1}, \ldots, X_{-7}, Y_{-7}, C)$

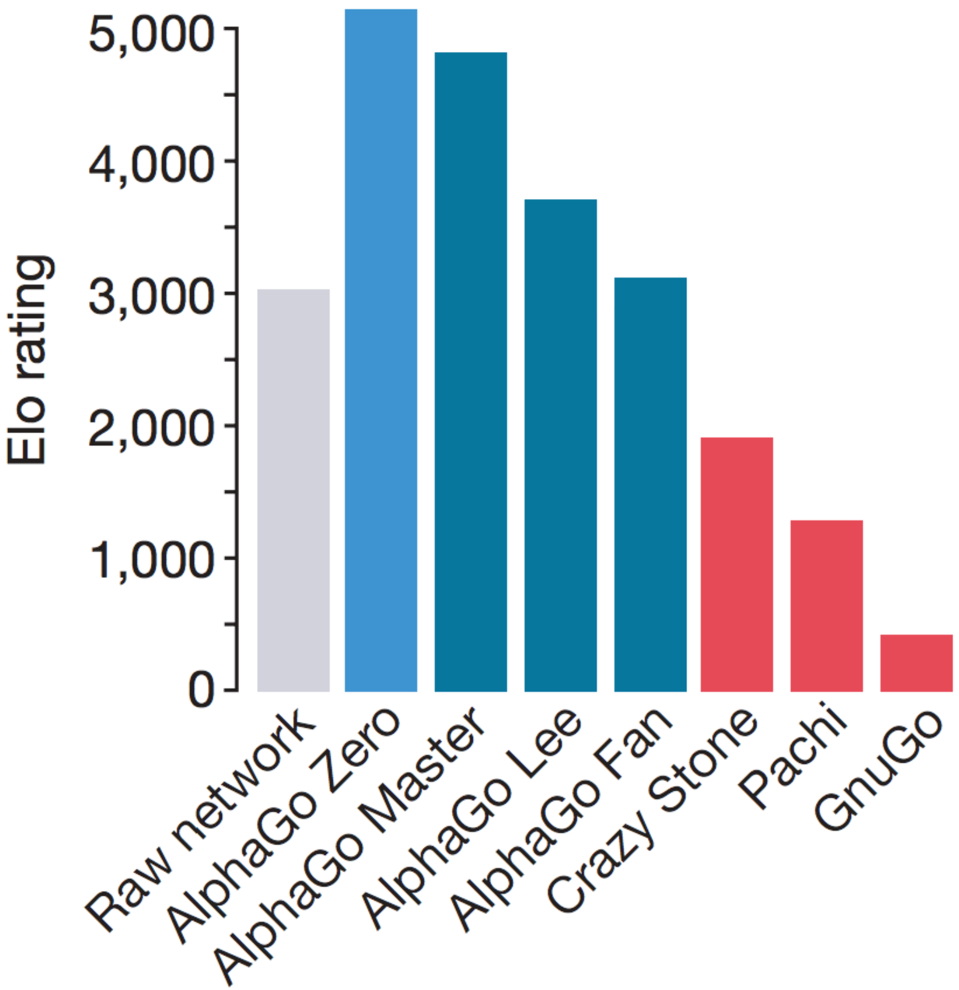
Objective:

$$(s, \boldsymbol{\pi}, z)$$

$$J(\theta) = (z - V_\theta)^2 - \boldsymbol{\pi}^T \log \mathbf{p}_\theta + c\|\theta\|^2$$

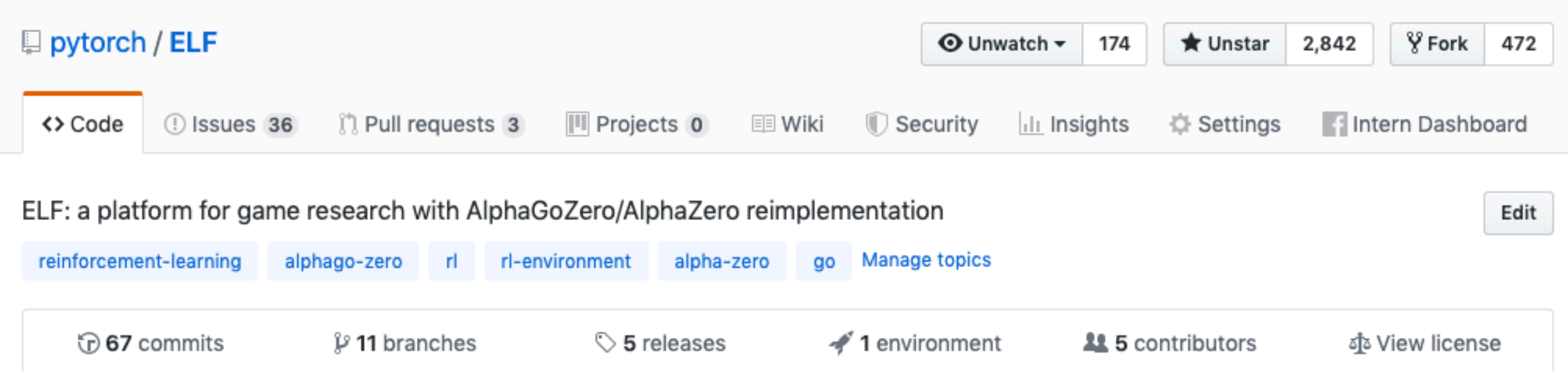
AlphaGo Zero Strength

- 3 days version
  - 4.9M Games, 1600 rollouts/move
  - 20 block ResNet
  - Defeat AlphaGo Lee.
- 40 days version
  - 29M Games, 1600 rollouts/move
  - 40 blocks ResNet.
  - Defeat AlphaGo Master by 89:11



ELF OpenGo

- System can be trained with 2000 GPUs in 2 weeks (20 block version)
- Superhuman performance against professional players and strong bots.
- Abundant ablation analysis
- Decoupled design, code reusable for other games.



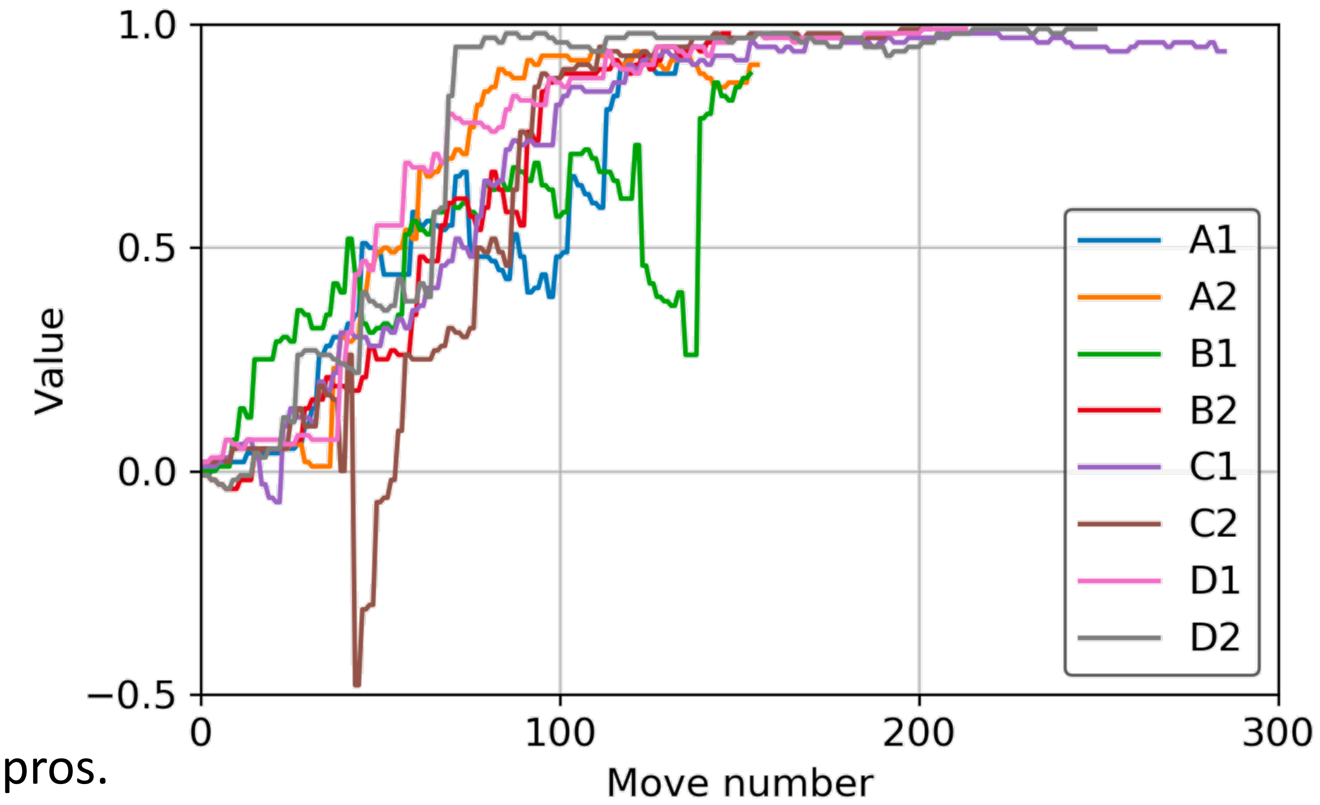
We open source the code and the pre-trained model for the Go and ML community

ELF OpenGo Performance

Vs top professional players

Name (rank)	ELO (world rank)	Result
Kim Ji-seok	3590 (#3)	5-0
Shin Jin-seo	3570 (#5)	5-0
Park Yeonghun	3481 (#23)	5-0
Choi Cheolhan	3466 (#30)	5-0

Single GPU, 80k rollouts, 50 seconds
Offer unlimited thinking time for the players



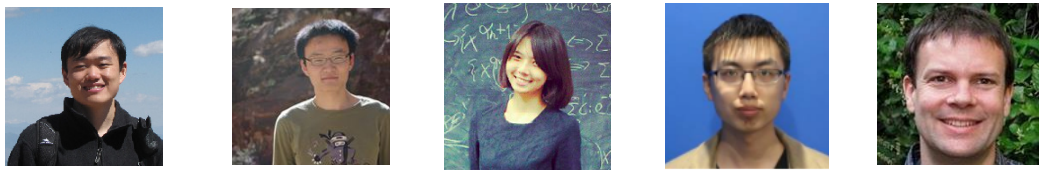
Vs professional players

Single GPU, 2k rollouts, 27-0 against Taiwanese pros.

Vs strong bot (LeelaZero)

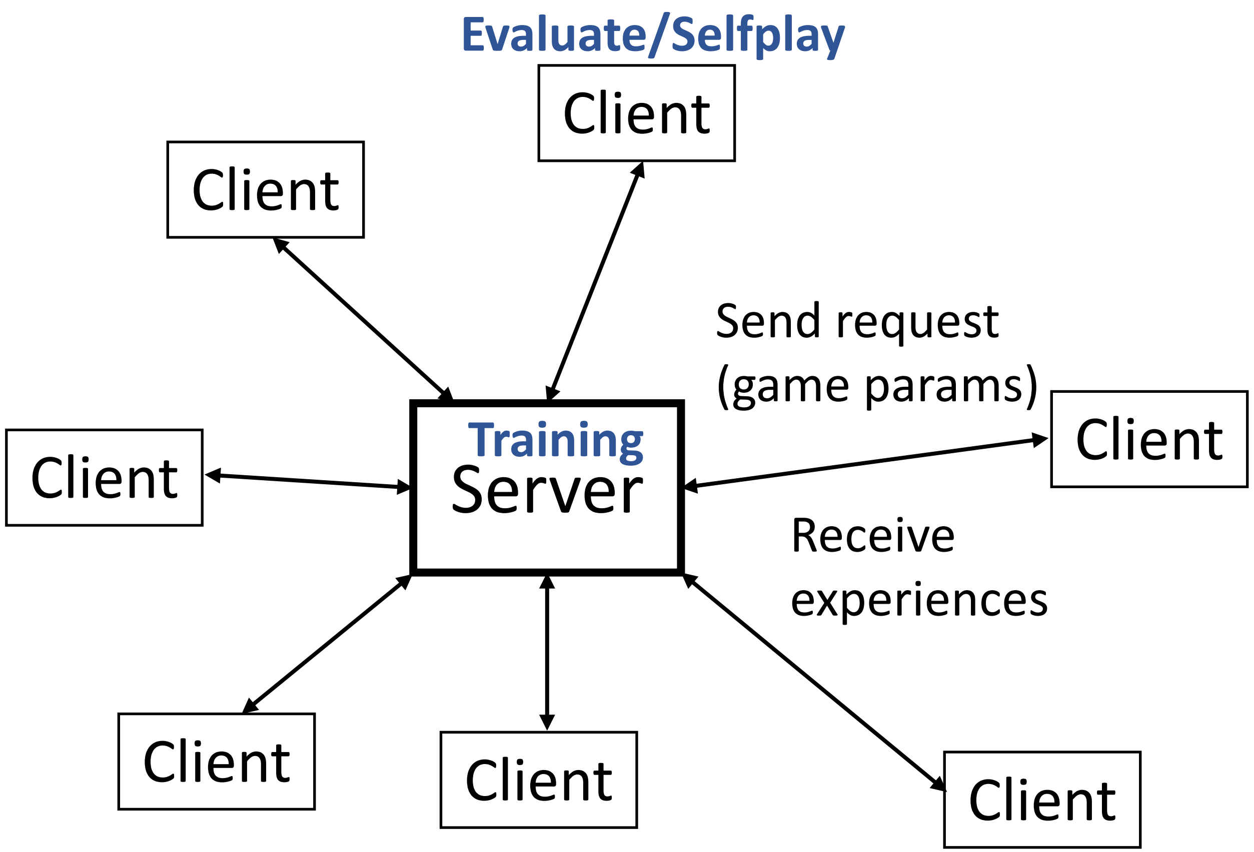
[158603eb, 192x15, Apr. 25, 2018]: 980 wins, 18 losses (98.2%)

Distributed System



Yuandong Tian   Qucheng Gong   Wenling Shang   Yuxin Wu   Larry Zitnick

[Y. Tian et al, ELF: An Extensive, Lightweight and Flexible Research Platform for Real-time Strategy Games, NIPS 2017]

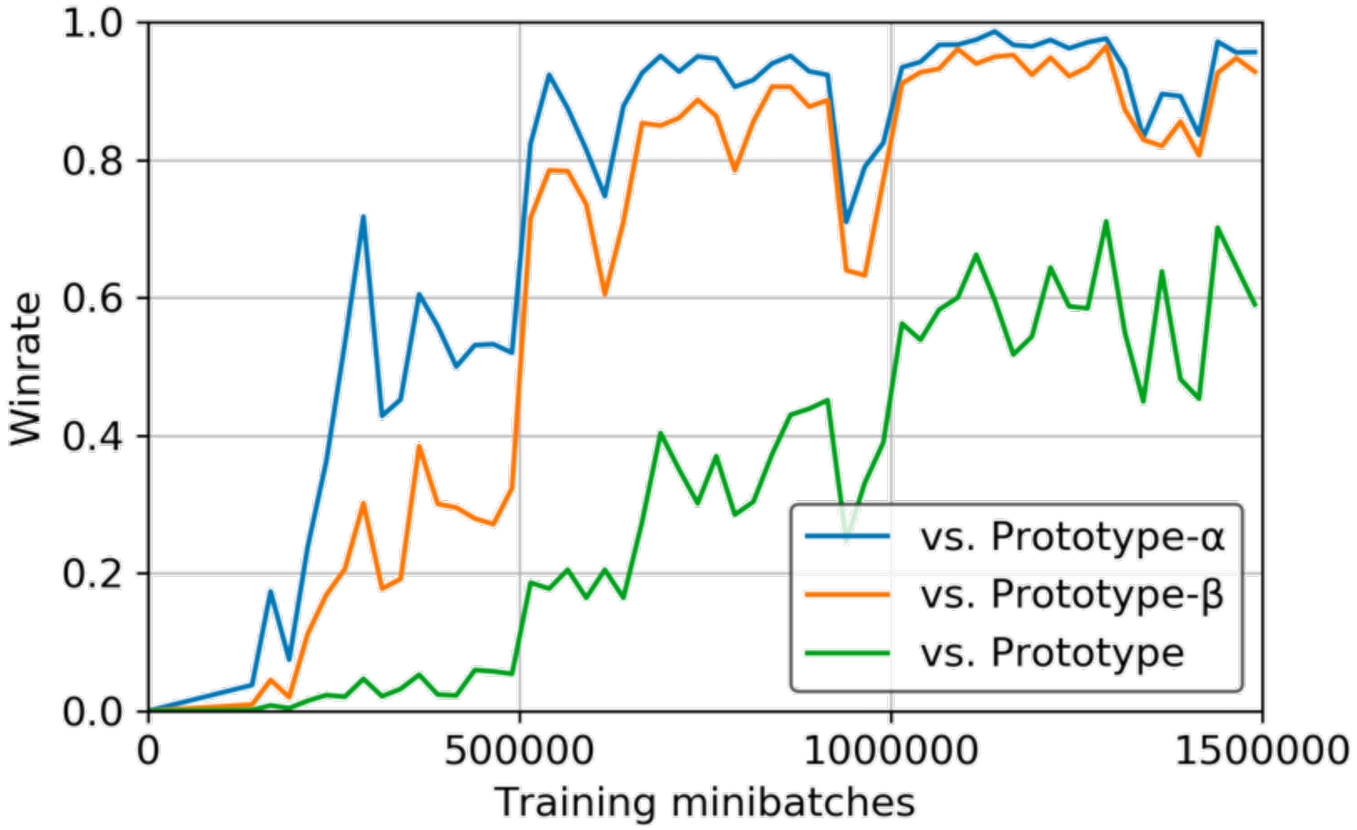


Putting AlphaGoZero and AlphaZero into the same framework

AlphaGoZero (more synchronization)
AlphaZero (less synchronization)

Server controls synchronization
Server also does training.

Training Stage of Final Model



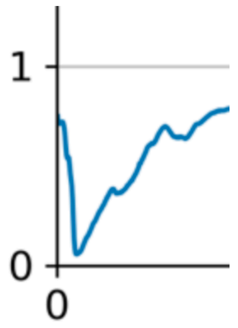
Prototype-α = strong amateur level

Prototype-β = professional level

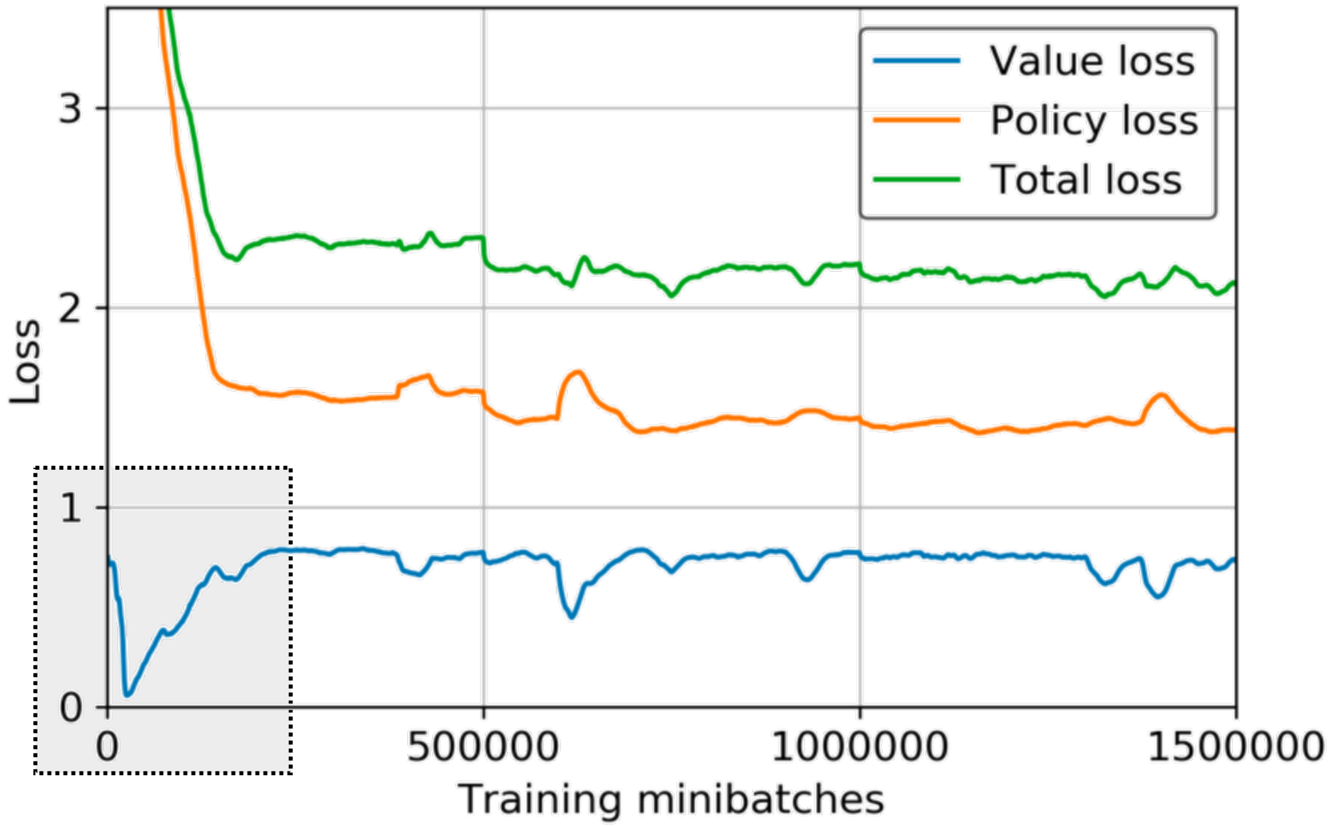
Prototype = superhuman level
(model against professional players)

A lot of zig-zag in the training process

Overfitting issues



Dip of the value function



Overestimate white winrate

Black resigns prematurally

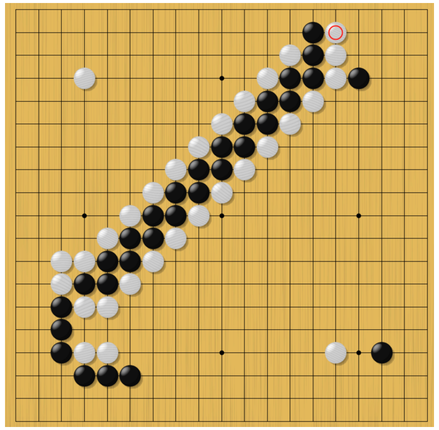
Black loses many games

Imbalanced replay buffer

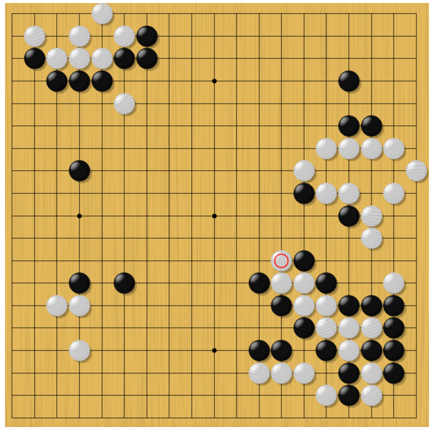
Large replay buffer is the key

Adaptive resign threshold has delays

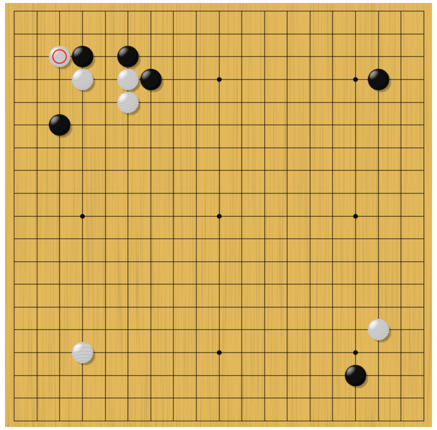
Ladder Issues



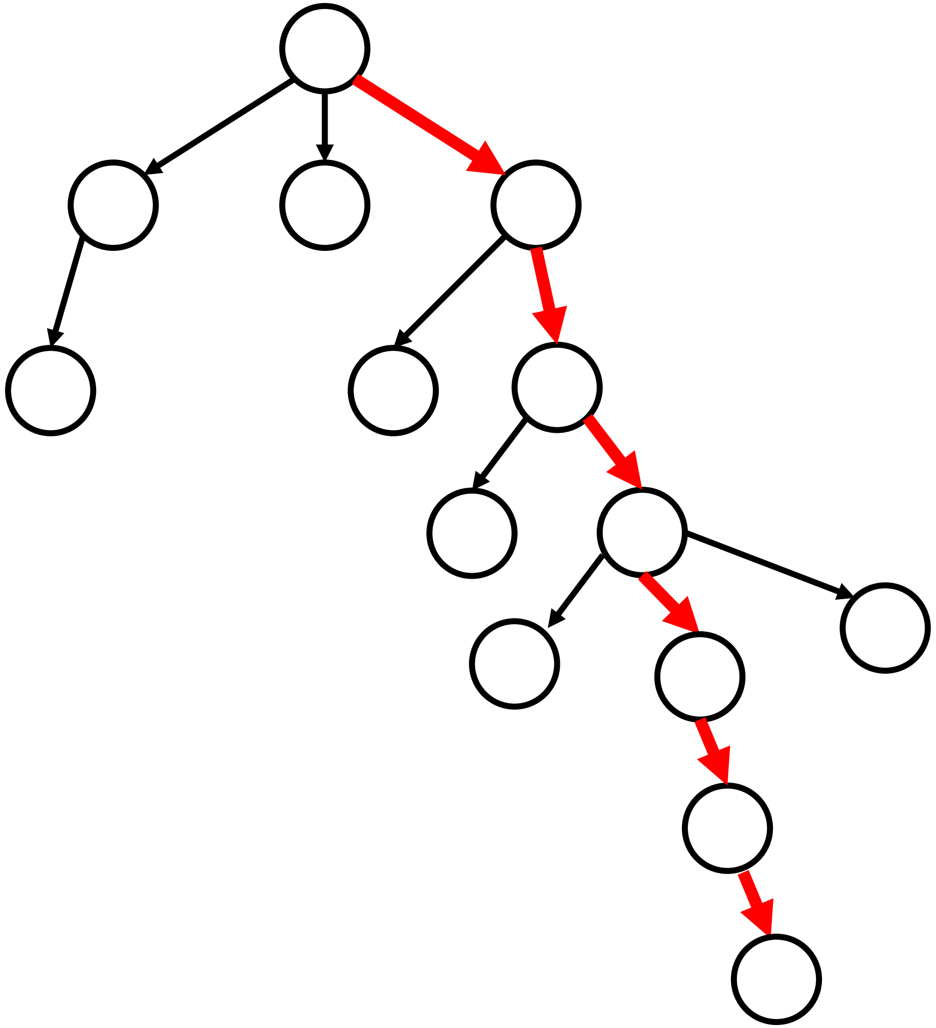
Run a ladder and lost



Run shorter ladder and lost



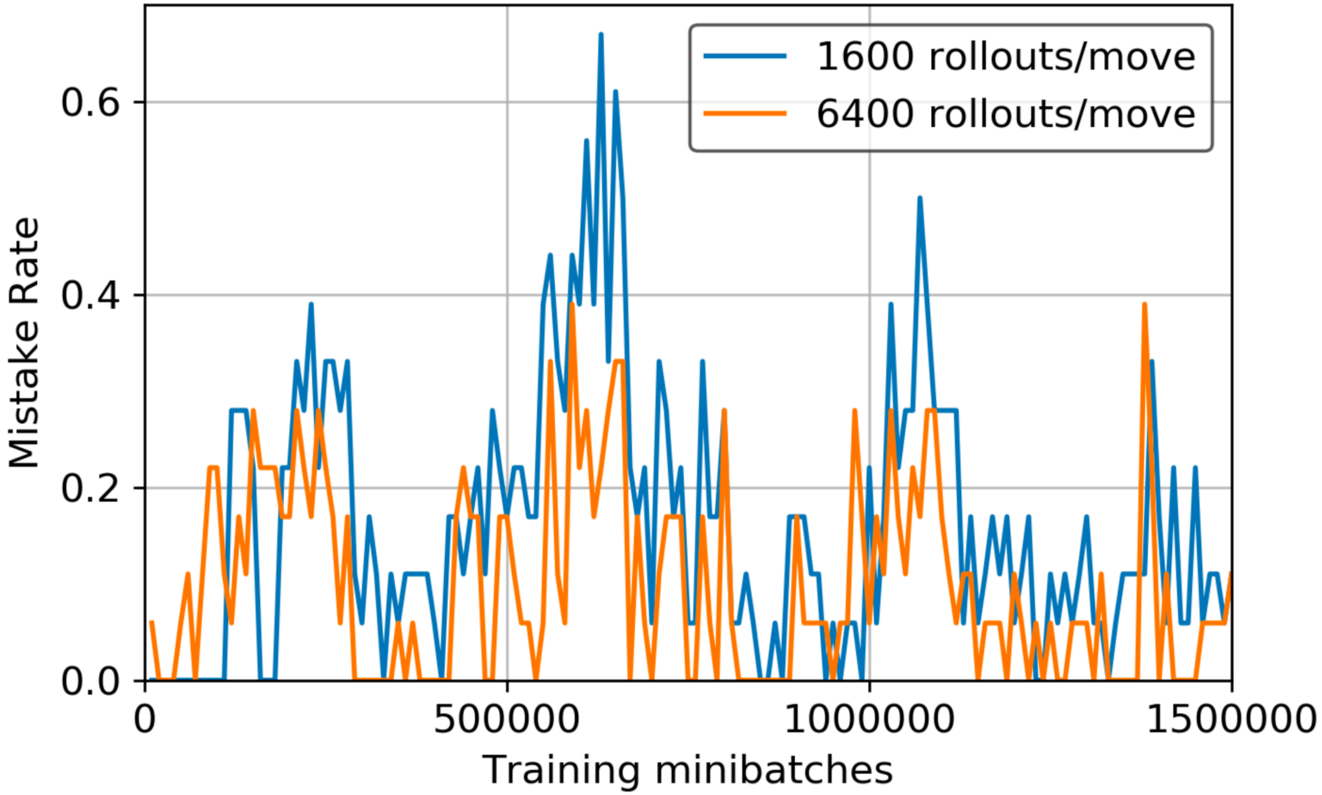
Doesn't run ladder



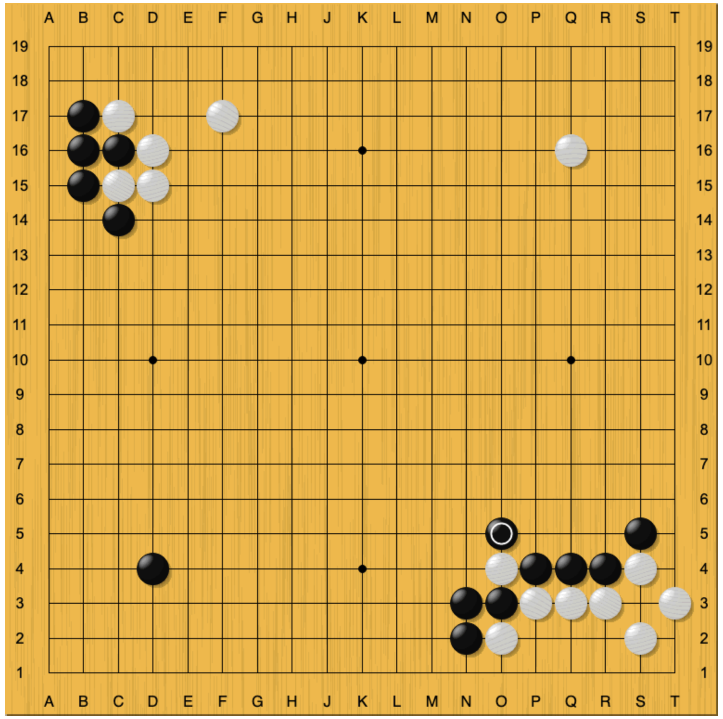
There is only one long path that is correct

Value propagation is really slow.

Did we solve ladder?



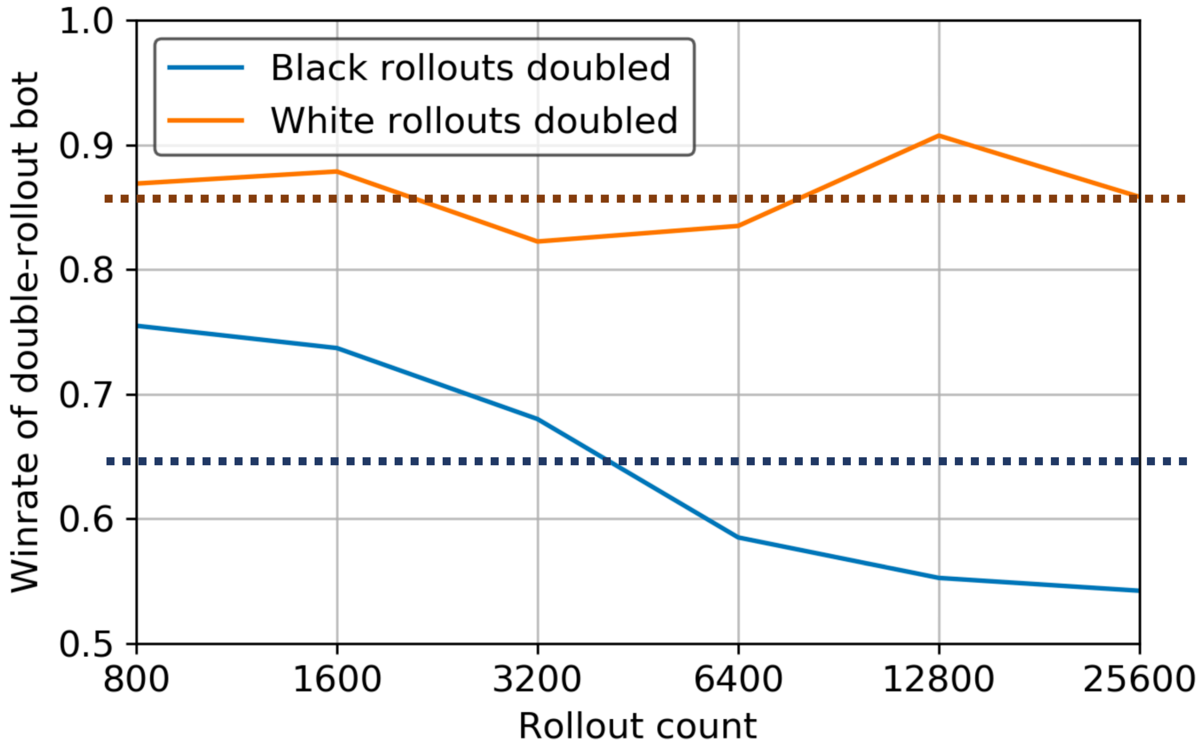
No



Why is the model still strong? → It plays alternative moves to avoid these situations.

Why MCTS is so important?

Look-ahead is how new knowledge is created.



On Final Model

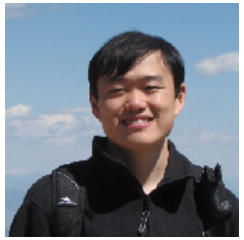
White rollouts 2x → ~85% winrate

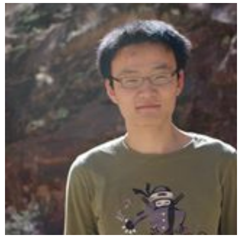
Black rollouts 2x → ~65% winrate

Training is almost always constrained by model capacity (why 40b > 20b)

Principled Algorithm
Guaranteed Performance
Good Empirical Results

Joint Policy Search and Contract Bridge Bidding

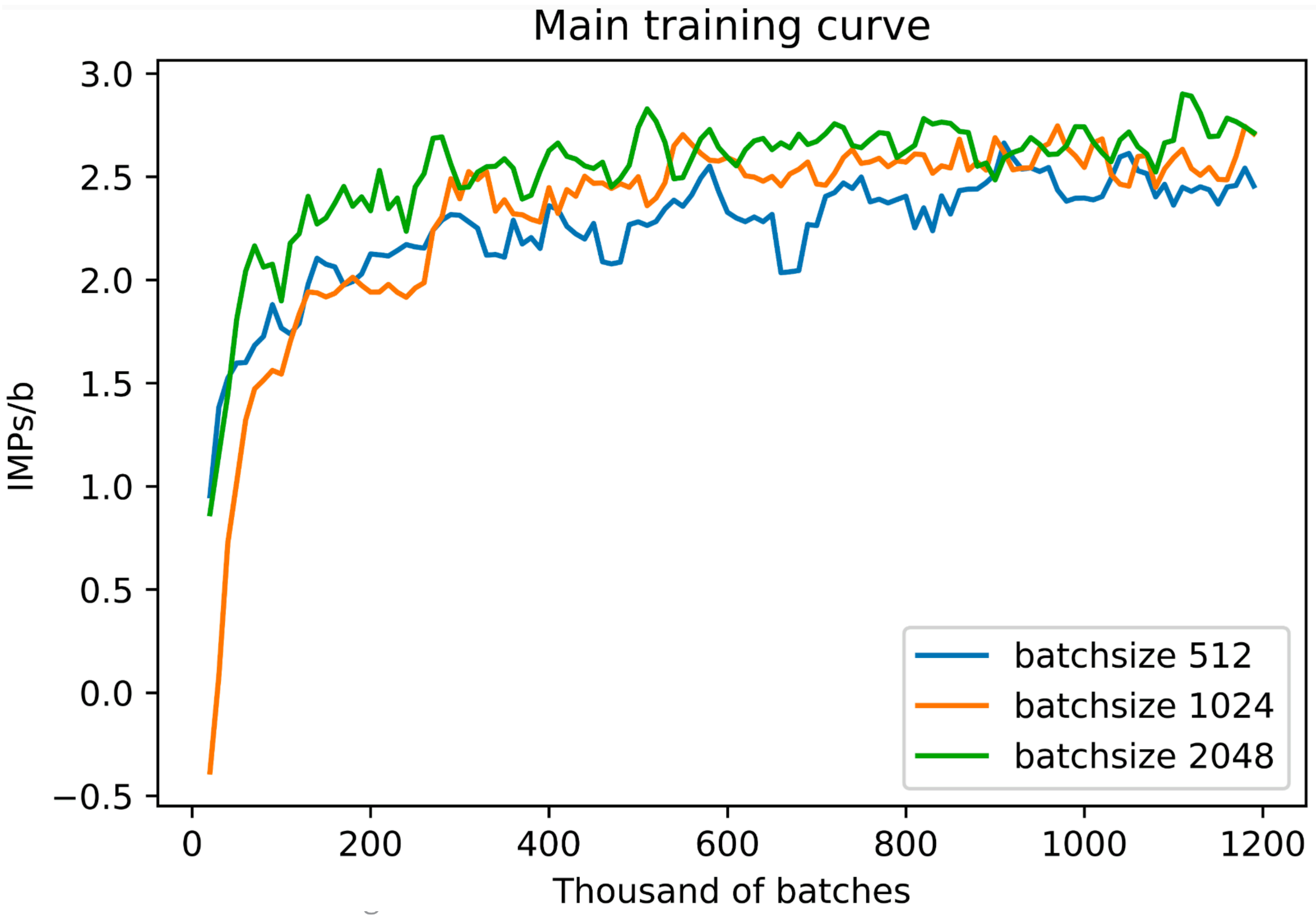

Yuandong Tian


Qucheng Gong


Tina Jiang

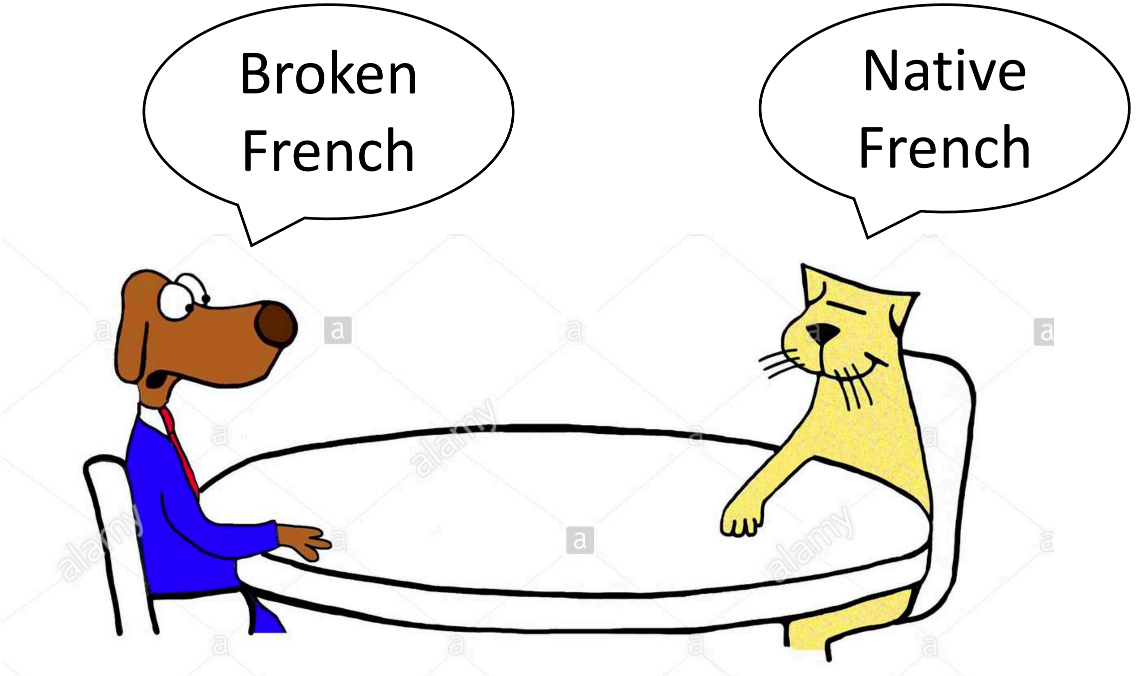
[Y. Tian et al., Joint Policy Search for Collaborative Multi-agent Imperfect Information Game, NeurIPS 2020]

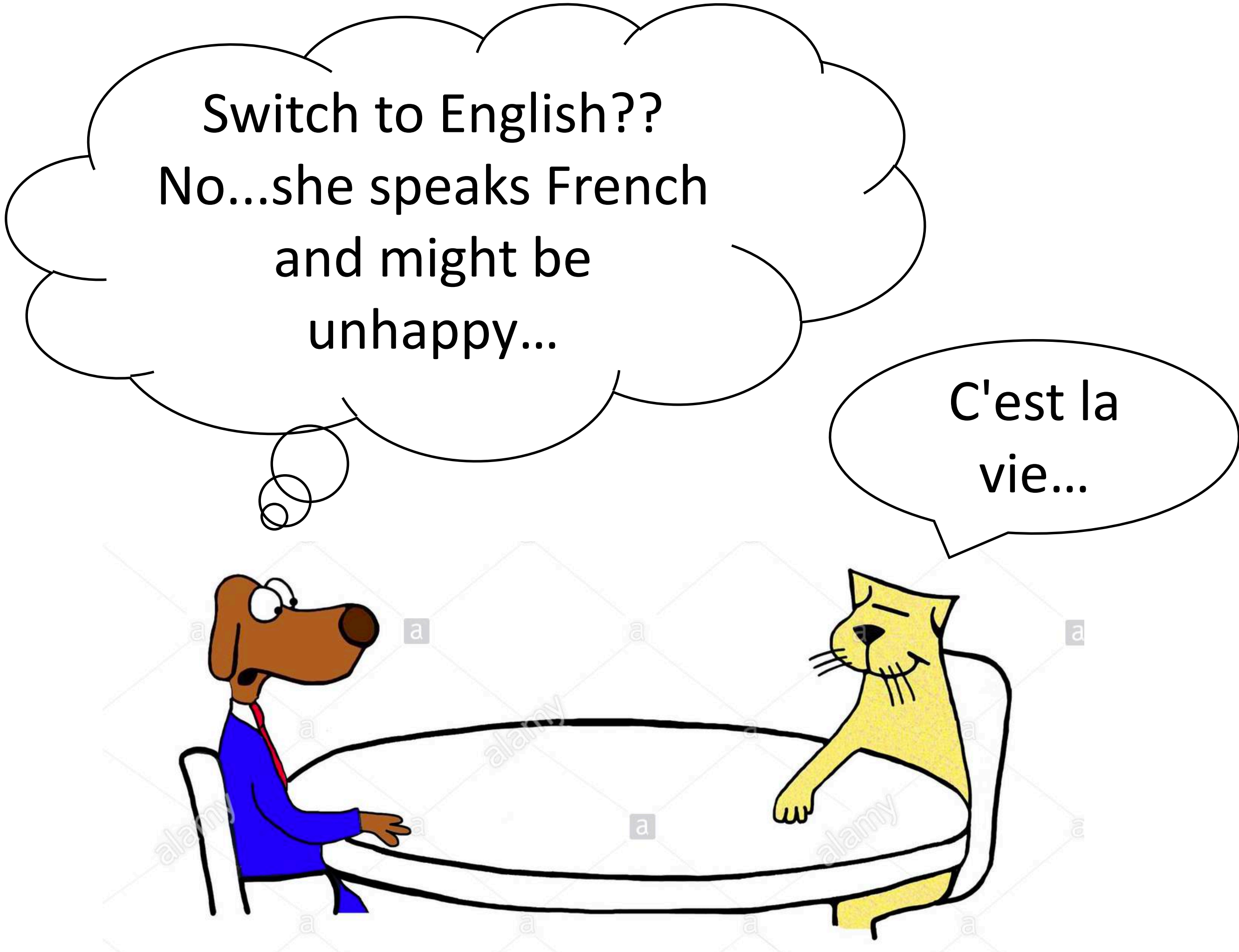
When Self-Play Fails?



Training with self-play + A2C get stuck in local minima

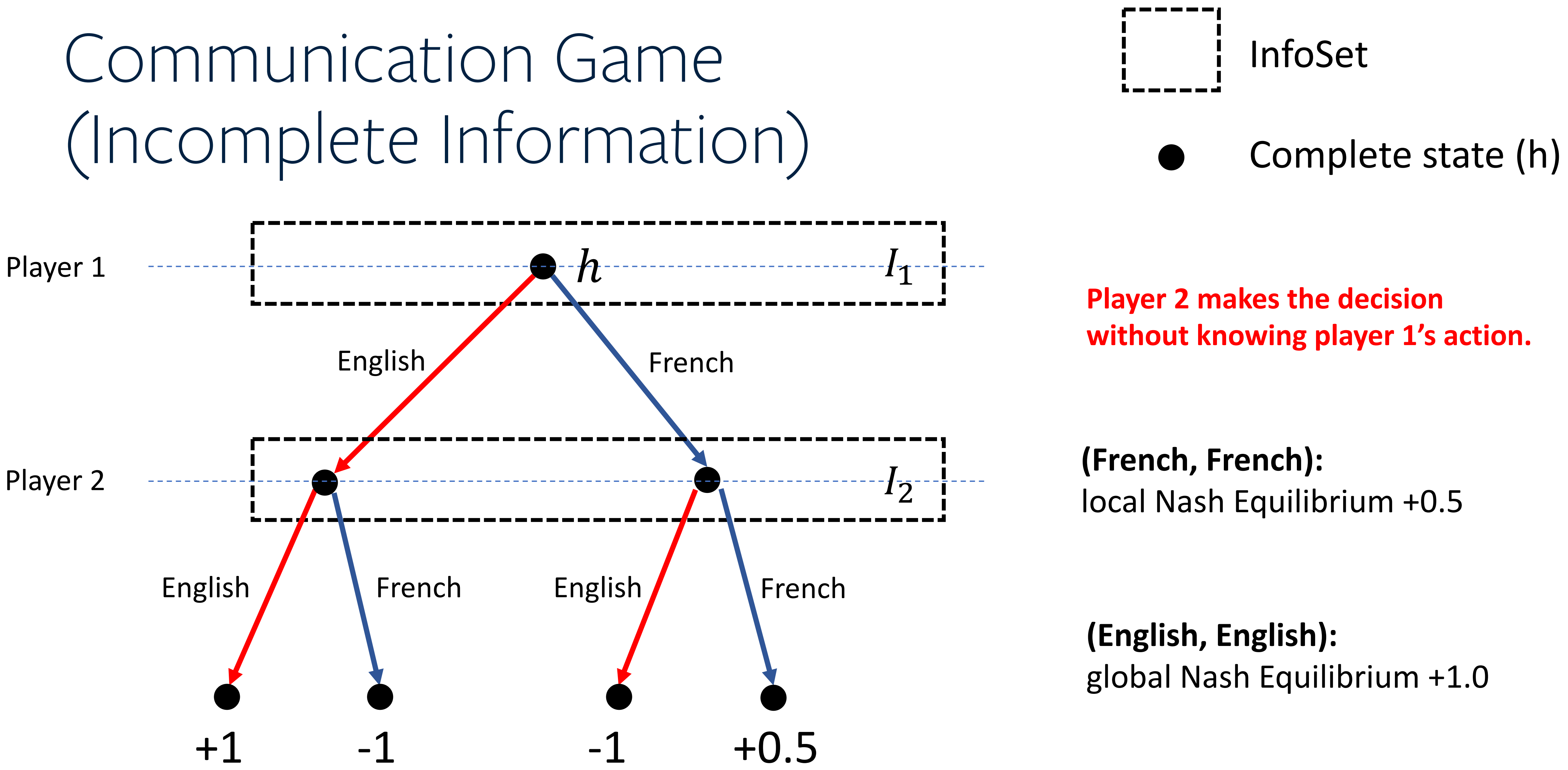
An example





A **unilateral** change of policy doesn't improve co-operative communication
(many single-agent DRL approach improves by unilateral changes of agent policy)

Communication Game (Incomplete Information)



Player 2 makes the decision without knowing player 1's action.

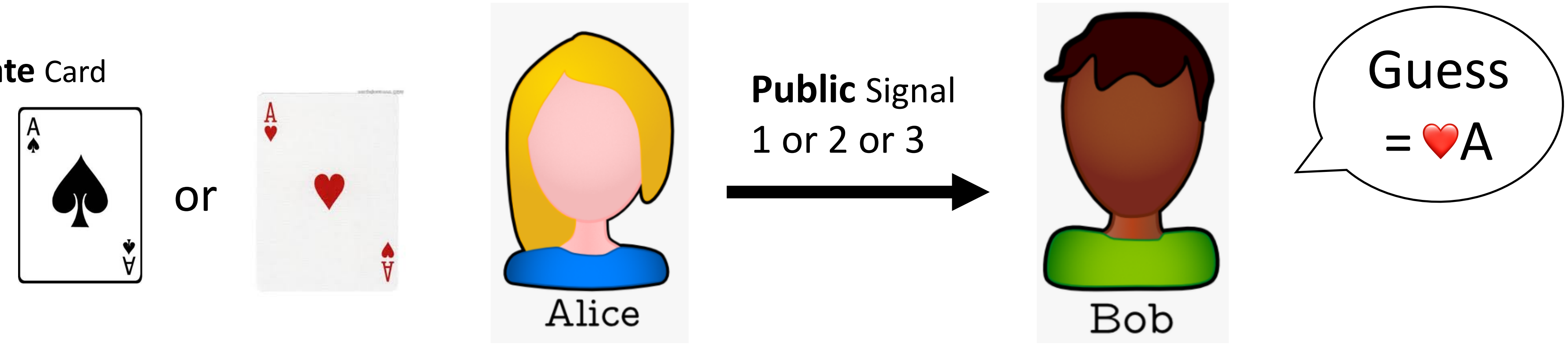
(French, French):
local Nash Equilibrium +0.5

(English, English):
global Nash Equilibrium +1.0

A joint optimization of policy $\sigma(I_1)$ and $\sigma(I_2)$ *yields optimal solution*

Another Illustrative Example (Imperfect Information)

Private Card



One possible solution (6 symmetric solutions):

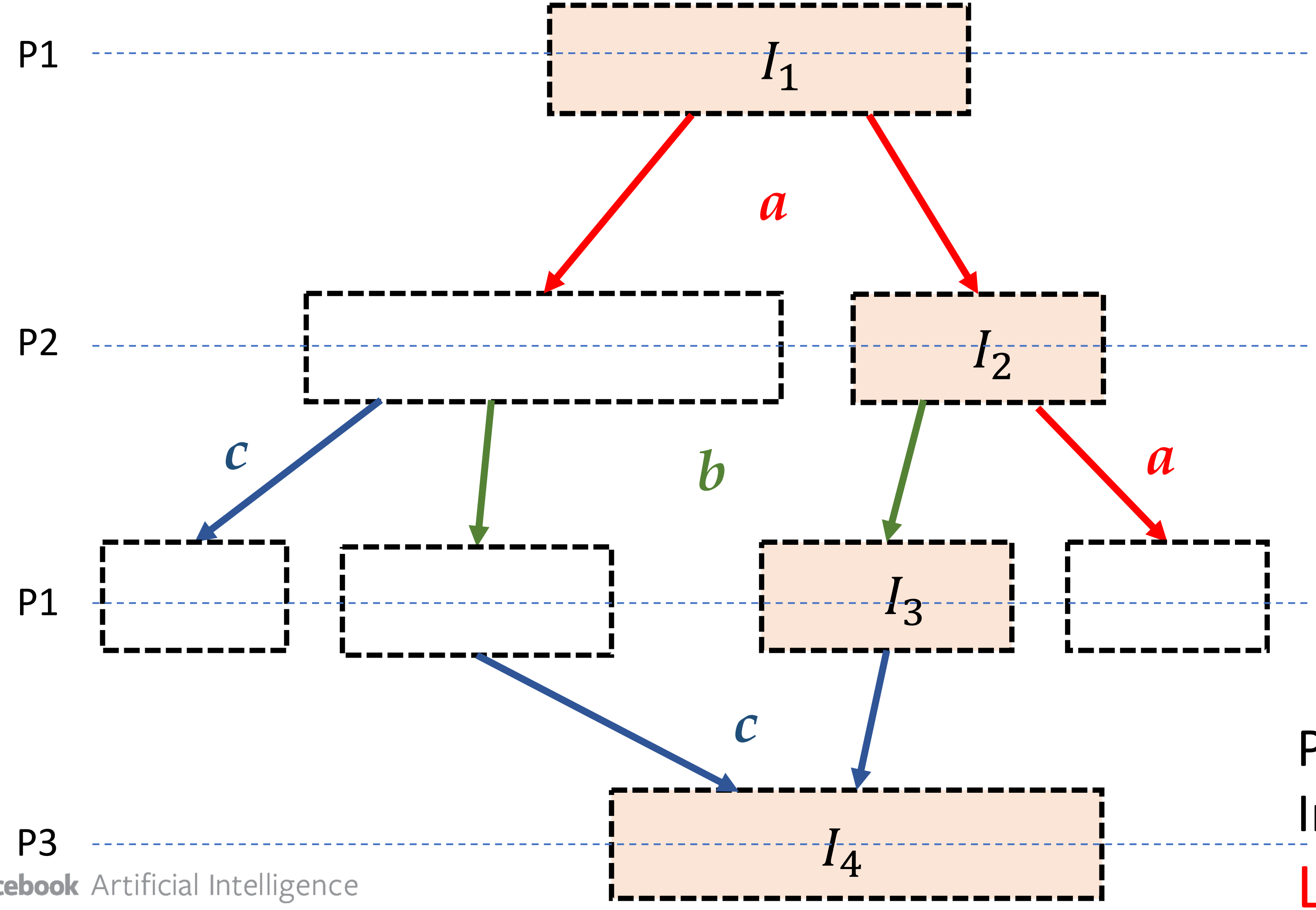
Private card	Alice's Action	Bob's Action
❤️ A	1	Guess ❤️ A
A	3	Guess A
--	2	--

Not used

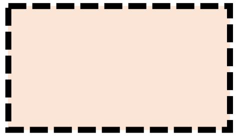
What if Allice and Bob never use signal 2,

but sending signal 2 has additional rewards?

Optimize Policies in Multiple Infosets



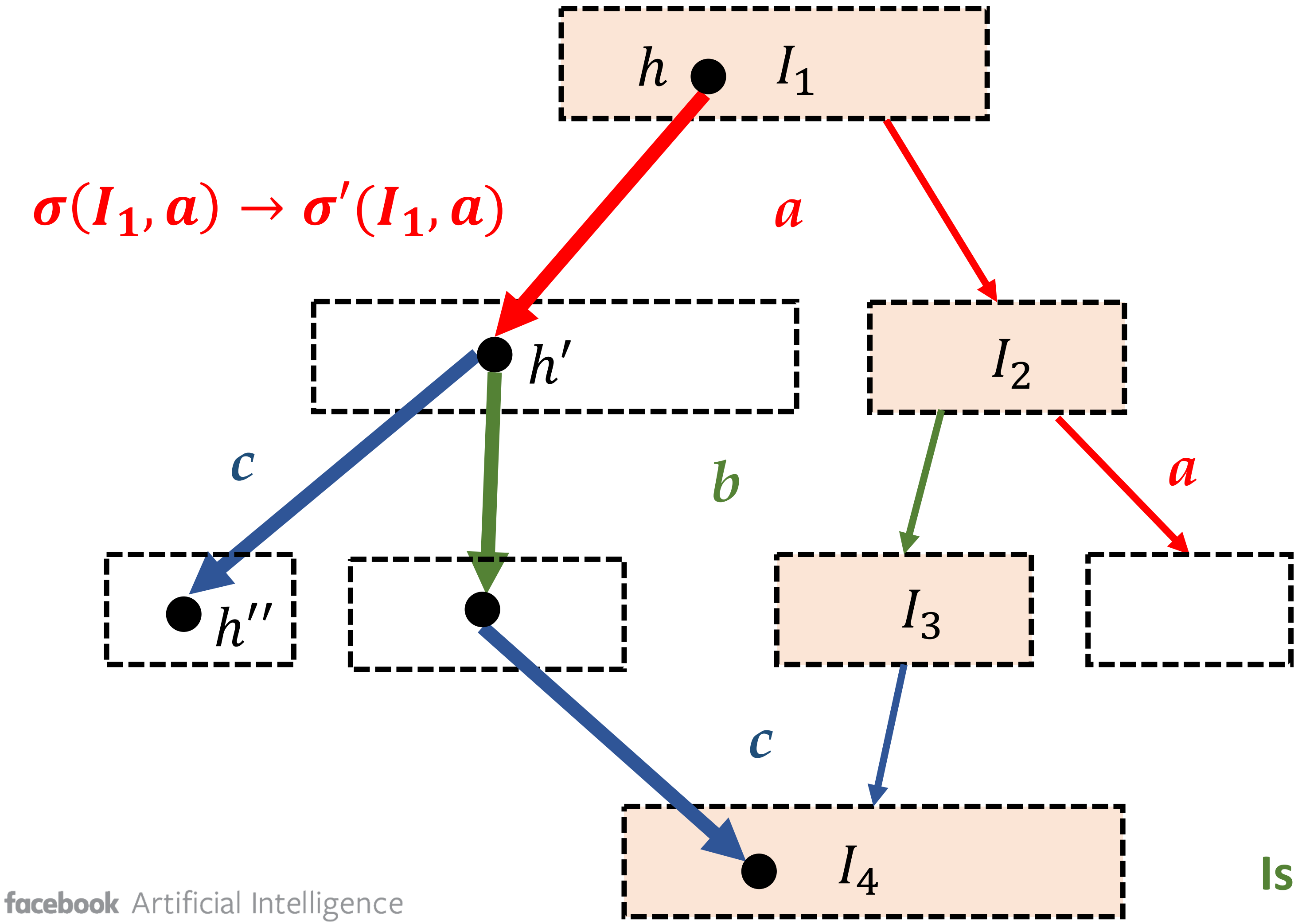
A sparse set of *active* infosets to be optimized



Policy: $\sigma(I)$

Perfect information → A Subtree
Imperfect information → A Graph
Lots of dependencies!

Dependency between policies



A change of $\sigma(I_1, a)$ affects **all** the reachability of down-stream states and/or infosets, no matter they are *active* or not.

A trajectory could re-enter into another active set and leave and re-enter again.

The value of an inactive infoset I_3 will change since the reachability to I_3 changes.

An infoset might contain both affected states and unaffected states.

Is there a good way to track value changes?

Optimize Policies in Multiple Infosets

$$\sigma \quad \rightarrow \quad \sigma'$$

Current Policy → New Policy

$$\bar{v}^{\sigma} \quad \rightarrow \quad \bar{v}^{\sigma'}$$

Current
Game Value

New
Game Value

Policy-change Density

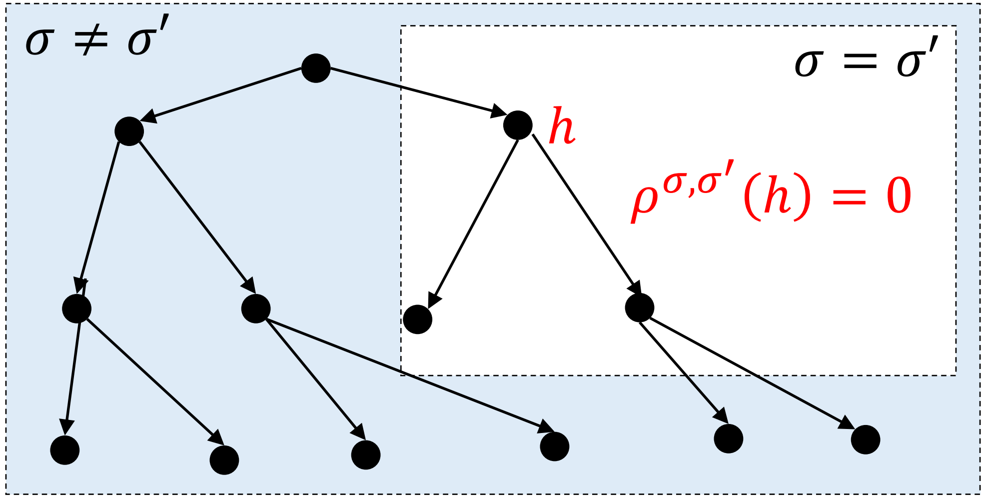
Density $\rho^{\sigma,\sigma'}(h) = \pi^{\sigma'}(h)\left[\sum_{a\in A(I)} \sigma'(I,a)v^{\sigma}(ha) - v^{\sigma}(h)\right]$

Two key properties:

(a) Its summation yields overall value changes

$$\bar{v}^{\sigma'} - \bar{v}^{\sigma} = \sum_{h\notin Z} \rho^{\sigma,\sigma'}(h)$$

(b) For regions with the same policy, it vanishes even if the overall reachability changes.



Value Changes w.r.t Localized Policy Change

Theorem

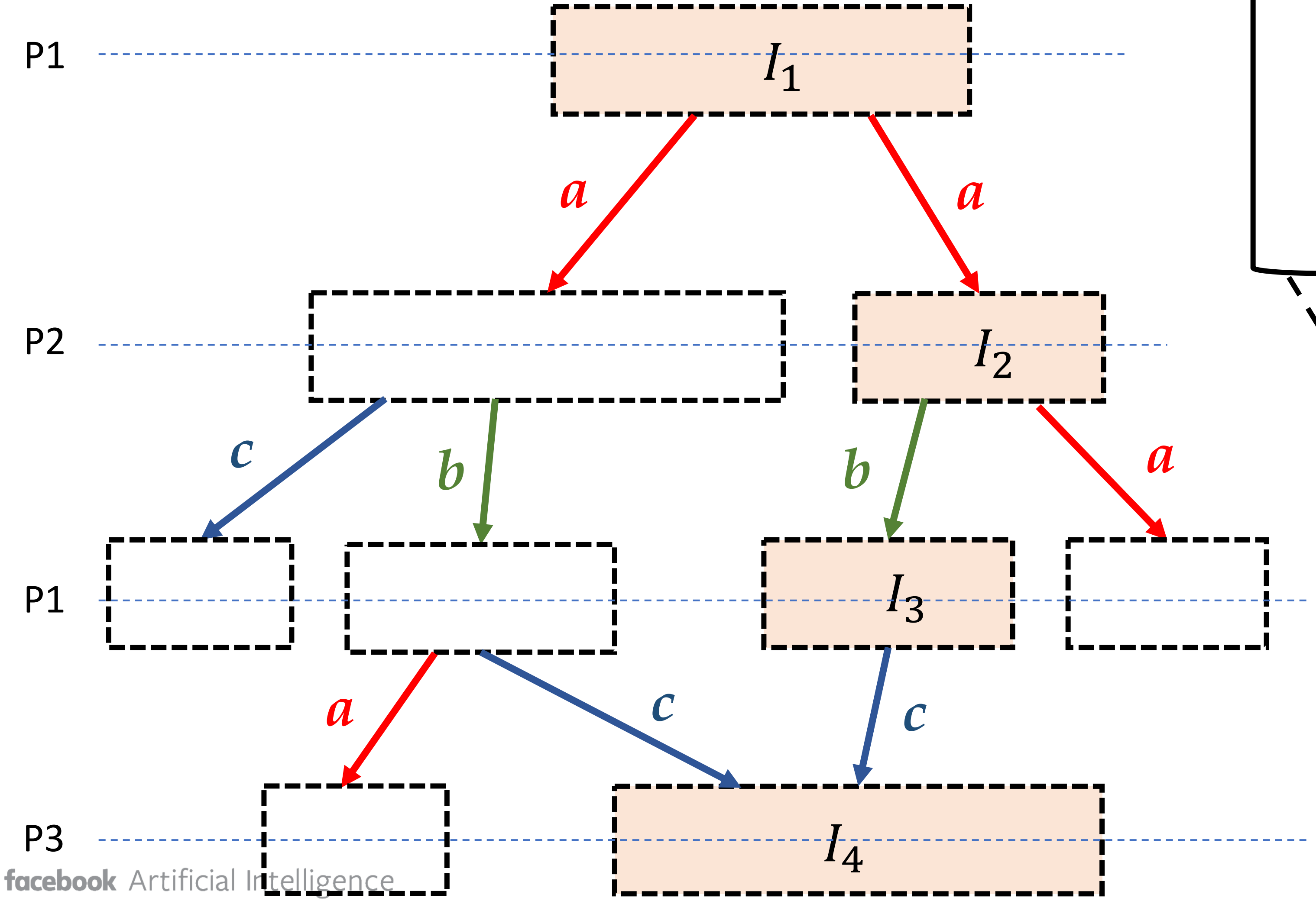
$$\bar{v}^{\sigma'} - \bar{v}^{\sigma} = \sum_{I \in \mathcal{I}} \sum_{h \in I} \rho^{\sigma,\sigma'}(h)$$

Overall value changes

All active Infosets
where $\sigma' \neq \sigma$

All complete states, doesn't matter
whether their reachability is affected or not

JPS (Joint Policy Search)



1. Initial infosets $I_{\text{cand}} = \{I_1\}$
2. Pick $I \in I_{\text{cand}}$
3. Pick an action a
4. Set $\sigma'(I, b) = \delta(a = b)$
5. Compute $\rho^{\sigma, \sigma'}$
6. Set $I_{\text{cand}} = \text{Succ}(I, a)$

Backtrace (depth-first search)

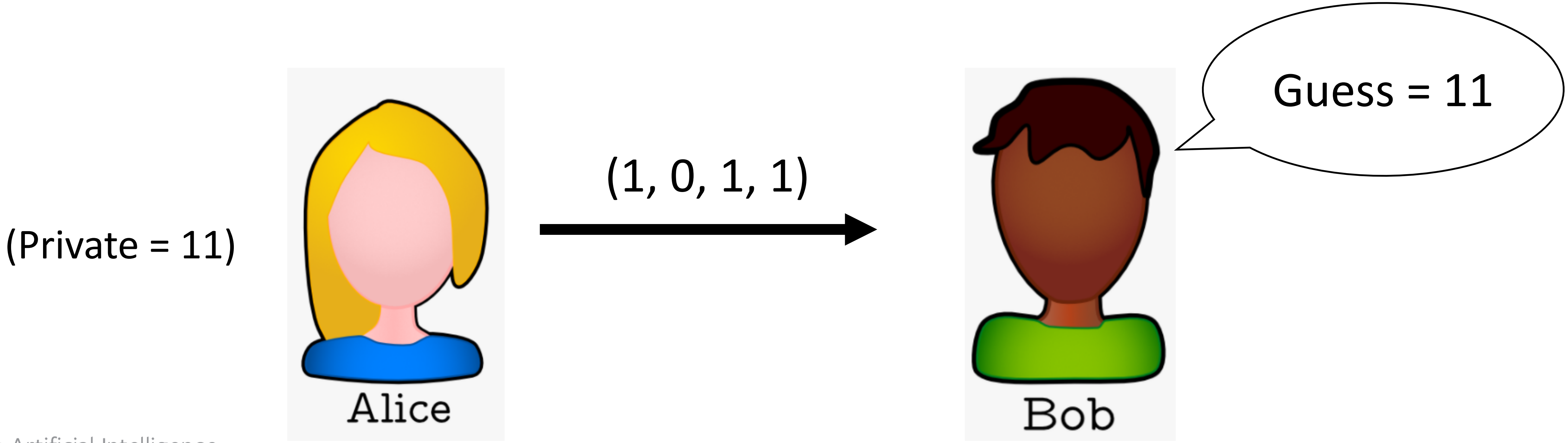
Repeat until maximal depth is reached.

Algorithm 1 Joint Policy Search (Tabular form)

function JSP-MAIN(σ)
    for $i = 1 \ldots T$ **do**
        Compute reachability π^σ and value v^σ under σ. Pick initial infoset I_1.
        $\sigma \leftarrow \mathrm{JPS}(\sigma, \{I_1\}, 1)$.
    end for
end function
function JPS($\sigma, \mathcal{I}_{\mathrm{cand}}, d$) ▷ *$\mathcal{I}_{\mathrm{cand}}$: candidate infosets*
    if $d \geq D$ **then**
        return 0. ▷ *Search reaches maximal depth D*
    end if
    for $I \in \mathcal{I}_{\mathrm{cand}}$ and $h \in I$ **do**
        Compute $\pi^{\sigma'}(h)$ by back-tracing $h' \sqsubset h$ until $I(h')$ is active. Otherwise $\pi^{\sigma'}(h) = \pi^\sigma(h)$.
    end for
    Compute $J^{\sigma,\sigma'}(I) = \sum_{h \in I} \rho^{\sigma,\sigma'}(h)$ for each $I \in \mathcal{I}_{\mathrm{cand}}$ using Eqn. 5.
    for $I \in \mathcal{I}_{\mathrm{cand}}$ and $a \in A(I)$ **do**
        Set I active. Set $\sigma'(I)$ and reachability accordingly Eqn. 6.
        Set $r(I, a) = \mathrm{JPS}(\sigma, \mathrm{succ}(I, a), d + 1) + J^{\sigma,\sigma'}(I)$
    end for
    return $\max(0, \max_{I,a} r(I, a))$ ▷ Also consider if no infoset in $\mathcal{I}_{\mathrm{cand}}$ is active.
end function

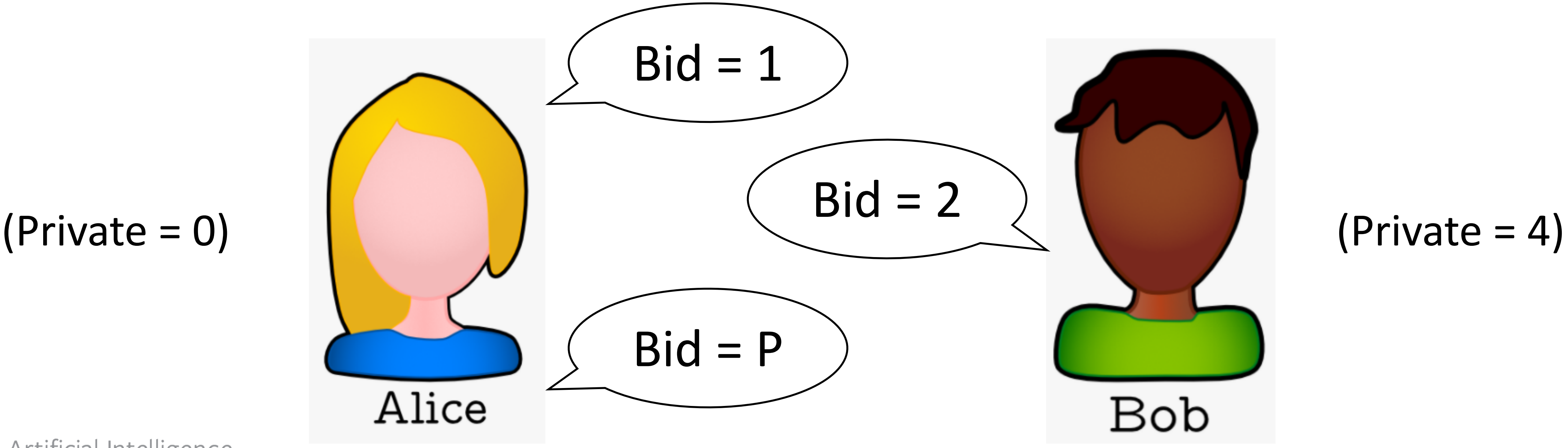
Results on Simple Games

Definition 1 (Simple Communication Game of length L). *Consider a game where* $s_1 \in \{0, \ldots, 2^L - 1\}$, $a_1 \in \mathcal{A}_1 = \{0, 1\}$, $a_2 \in \mathcal{A}_2 \in \{0, \ldots, 2^L - 1\}$. *P1 sends one binary public signal for L times, then P2 guess P1's private* s_1. *The reward* $r = \mathbf{1}[s_1 = a_2]$ *(i.e. 1 if guess right).*



Results on Simple Games

Definition 2 (Simple Bidding Game of size N). *P1 and P2 each dealt a private number* $s_1, s_2 \sim \text{Uniform}[0, \dots, N-1]$. $\mathcal{A} = \{\text{Pass}, 2^0, \dots, 2^k\}$ *is an ordered set. The game alternates between P1 and P2, and P1 bids first. The bidding sequence is strictly increasing. The game ends if either player passes, and* $r = 2^k$ *if* $s_1 + s_2 \geq 2^k$ *where* k *is the latest bid. Otherwise the contract fails and* $r = 0$.



Performance

	Comm (Def. 1)				Mini-Hanabi	Simple Bidding (Def. 2)			2SuitBridge (Def. 3)		
	$L=3$	$L=5$	$L=6$	$L=7$	[15]	$N=4$	$N=8$	$N=16$	$N=3$	$N=4$	$N=5$
CFR1k [43]	0.89*	0.85	0.85	0.85	9.11*	2.18*	4.96*	10.47	1.01*	1.62*	2.60
CFR1k+JPS	**1.00***	**1.00***	**1.00***	**1.00***	**9.50***	2.20*	**5.00***	**10.56***	**1.07***	**1.71***	**2.74***
A2C [26]	0.60*	0.57	0.51	0.02	8.20*	2.19	4.79	9.97	0.66	1.03	1.71
BAD [15]	**1.00***	0.88	0.50	0.29	9.47*	**2.23***	4.99*	9.81	0.53	0.98	1.31
Best Known	1.00	1.00	1.00	1.00	10	2.25	5.06	10.75	1.13	1.84	2.89
#States	633	34785	270273	2129793	53	241	1985	16129	4081	25576	147421
#Infosets	129	2049	8193	32769	45	61	249	1009	1021	5116	24571

JPS can improve existing policies, and help it jump out of local optima

Contract Bridge

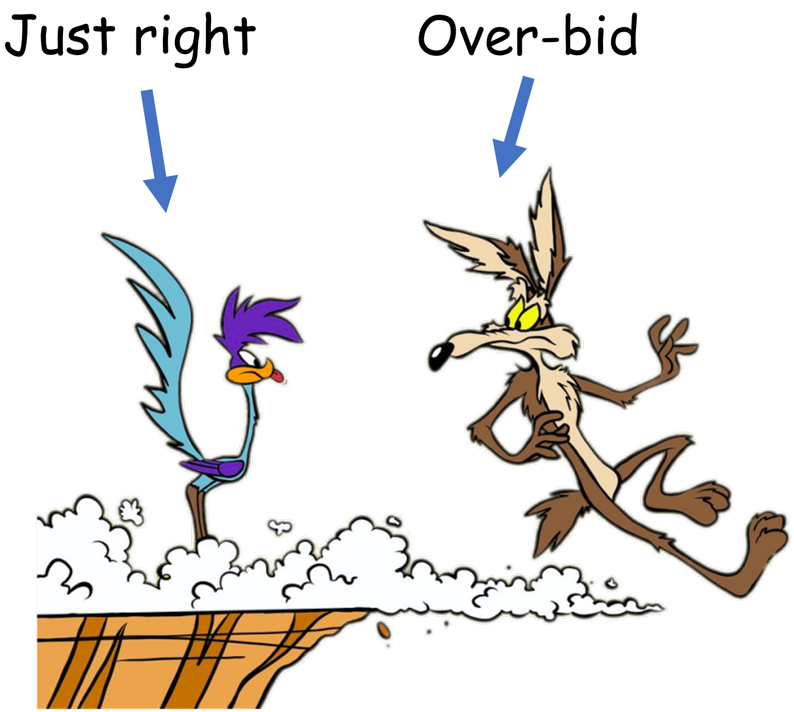
	N ♠A9743 ♥K8763 ♦A6 ♣7	
W ♠None ♥QJ952 ♦109 ♣KQ10982		**E** ♠Q82 ♥104 ♦QJ85432 ♣J
	S ♠KJ1065 ♥A ♦K7 ♣A6543	

- 25 million US players
- **100** years of history
- Incomplete Information
- Collaborative + Competitive
- Large State Space ($5.4*10^{28}$)

Bridge Bidding

West	North	East	South
			1♠
2♠[1]	2NT[2]	Pass	3♣
Pass	4♣[3]	Pass	4NT[4]
Pass	5♠[5]	Pass	7♠
Pass	Pass	Pass	

(1) Hearts and a minor. (2) Spade support, forcing to game. (3) Short clubs. (4) Keycard Blackwood. (5) Two key cards and the queen of spades, treating his fifth card as the equivalent of the queen.



Player only knows the private cards

Sequences of non-decreasing bids

The last bid is the contract

Fundamental Trade-off:
bid high via efficient communication, but not too much!

Evaluation against SoTA software (1000 games)

Methods	Vs. WBridge5 (IMPs/board)
Previous SoTA (Rong et al, 2019)	+ 0.25 (on 64 games)
Our A2C baseline	+ 0.29 ± 0.22
1% JPS (2 days)	+ 0.44 ± 0.20
5% JPS (2 days)	+ 0.37 ± 0.19
1% JPS (14 days)	**+ 0.63 ± 0.20**

WBridge5: Champions of computer bridge tournament in 2005, 2007, 2008, 2016-2018

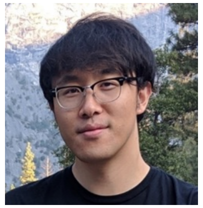
Bidding Visualization

Opening bids	Ours	SAYC
1♣	10+ HCP	12+ HCP, 3+♣
1♢	8-18 HCP, <4 ♡, <4 ♠	12+ HCP, 3+♢
1♡	4-16 HCP, 4-6♡	12+ HCP, 5+♡
1♠	4-16 HCP, 4-6♠	12+ HCP, 5+♠
1NT	12-17 HCP, bal	15-17 HCP, bal
2♣	6-13 HCP, 5+♣	22+ HCP
2♢	6-13 HCP, 5+♢	5-11 HCP, 6+♢
2♡	8-15 HCP, 5+♡	5-11 HCP, 6+♡
2♠	8-15 HCP, 5+♠	5-11 HCP, 6+♠

Good Empirical Performance
No theory yet

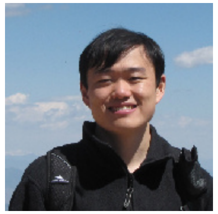
Learning Action Space in Monte Carlo Tree Search


Linnan Wang[1]


Saining Xie[2]

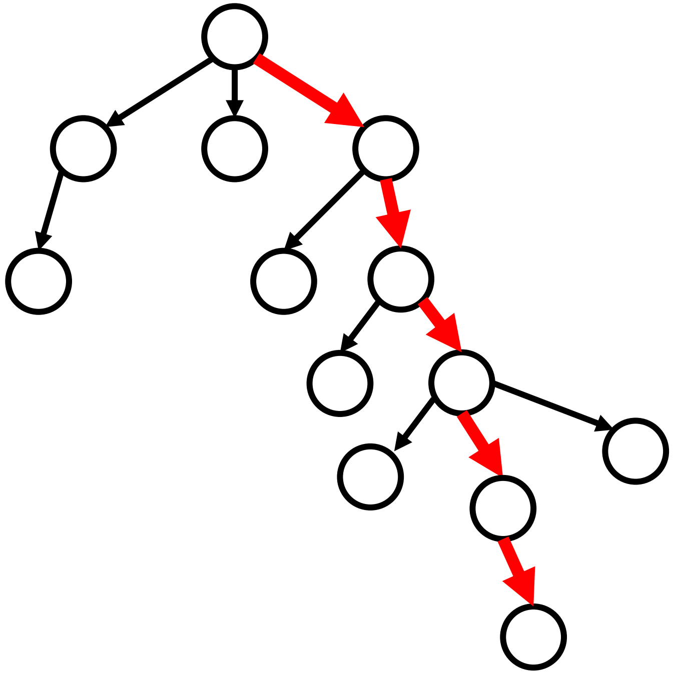

Teng Li[2]


Rodrigo Fonseca[1]

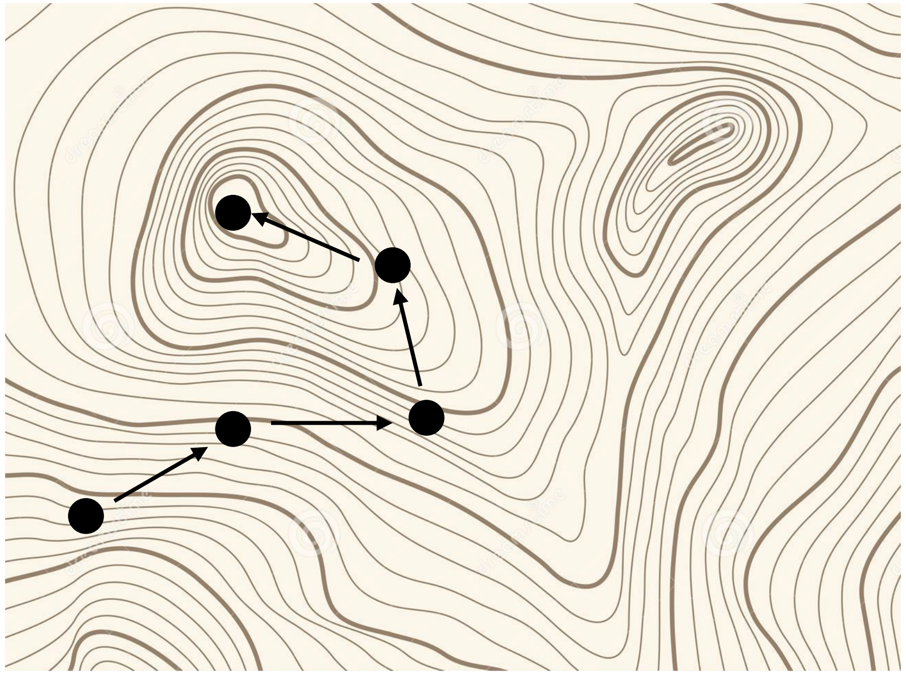

Yuandong Tian[2]

[1]Brown University, [2]Facebook AI Research

[L. Wang et al, Sample-Efficient Neural Architecture Search by Learning Action Space, arXiv]
[L. Wang et al, Learning Search Space Partition for Black-box Optimization using Monte Carlo Tree Search, NeurIPS 2020]



What else can Monte Carlo Tree Search (MCTS) be used?

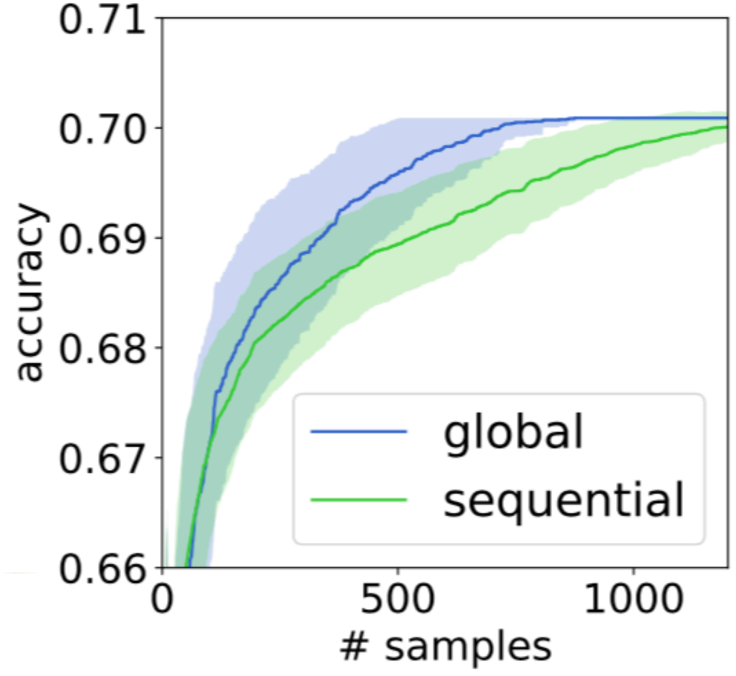


(Non-Convex) Optimization

Motivating Examples in Architecture Search

Depth = {1, 2, 3, 4, 5}
Channels = {32, 64}
KernelSize = {3x3, 5x5}

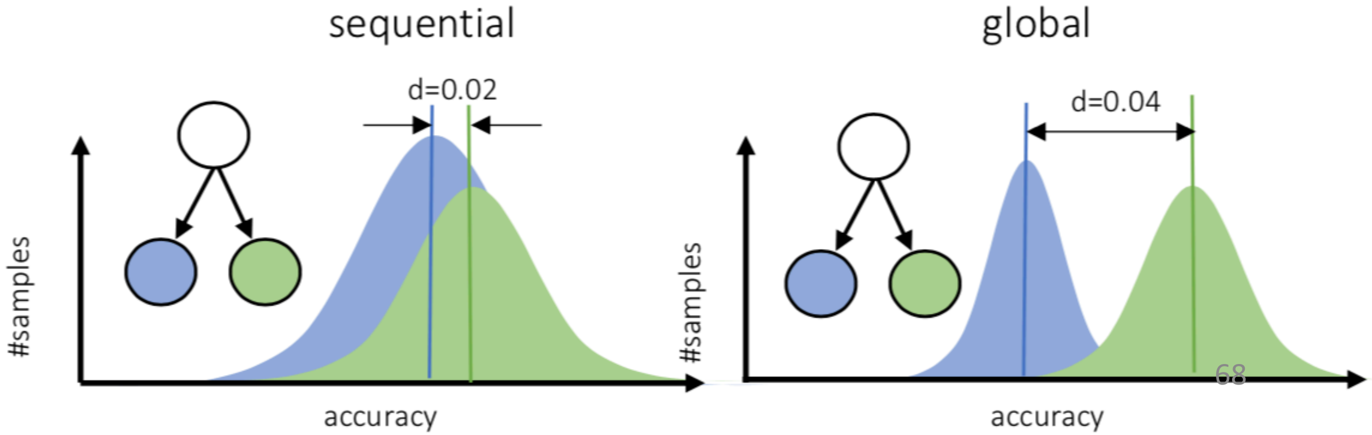
1364 networks.



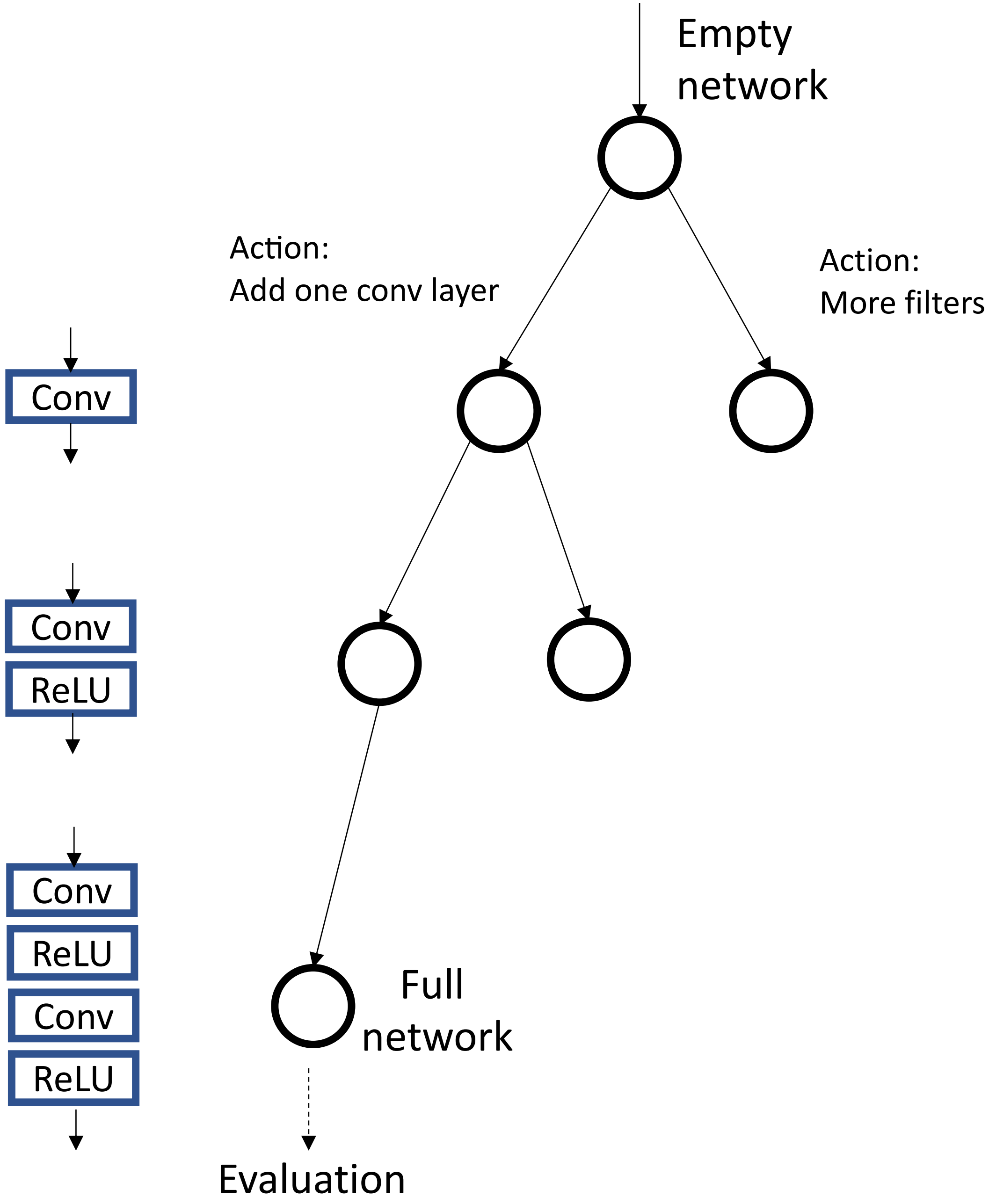
Action space

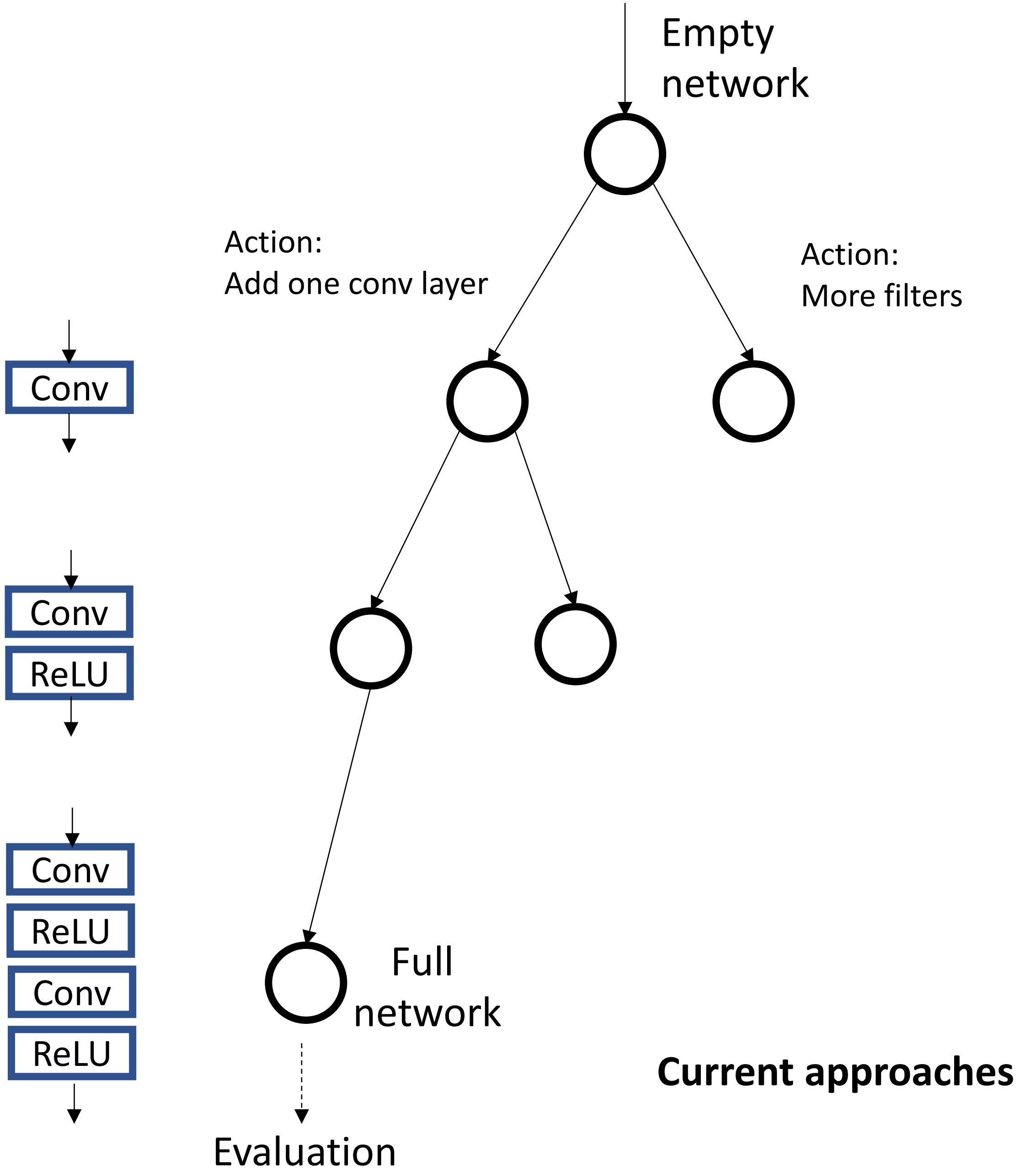
Sequential = { add a layer, set K, set C }

Global = { Set depth, set all K, set all C }

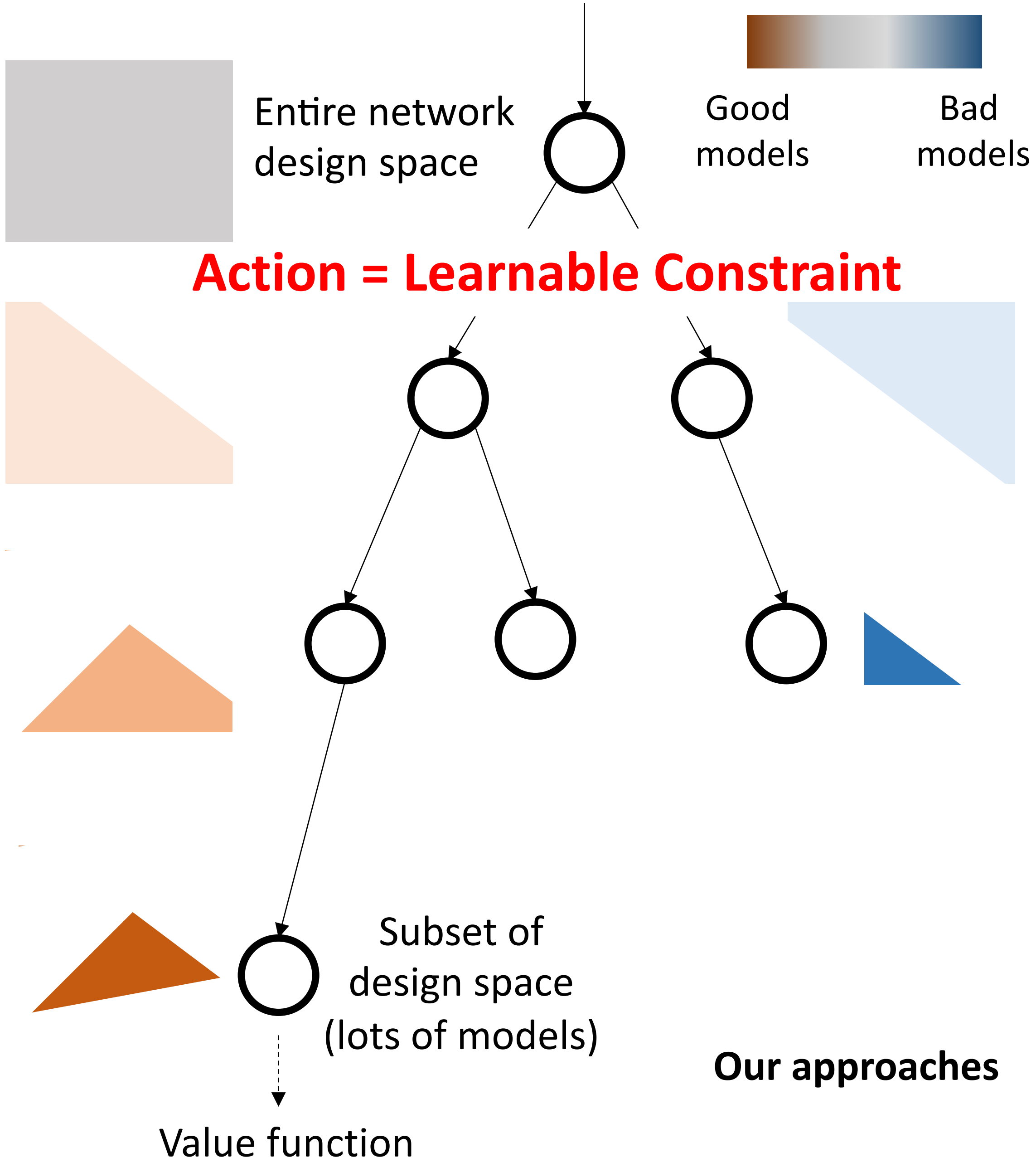


Global is better!


Empty
network
Action:
Add one conv layer
Action:
More filters
Conv
Conv
ReLU
Conv
ReLU
Conv
ReLU
Full
network
Evaluation

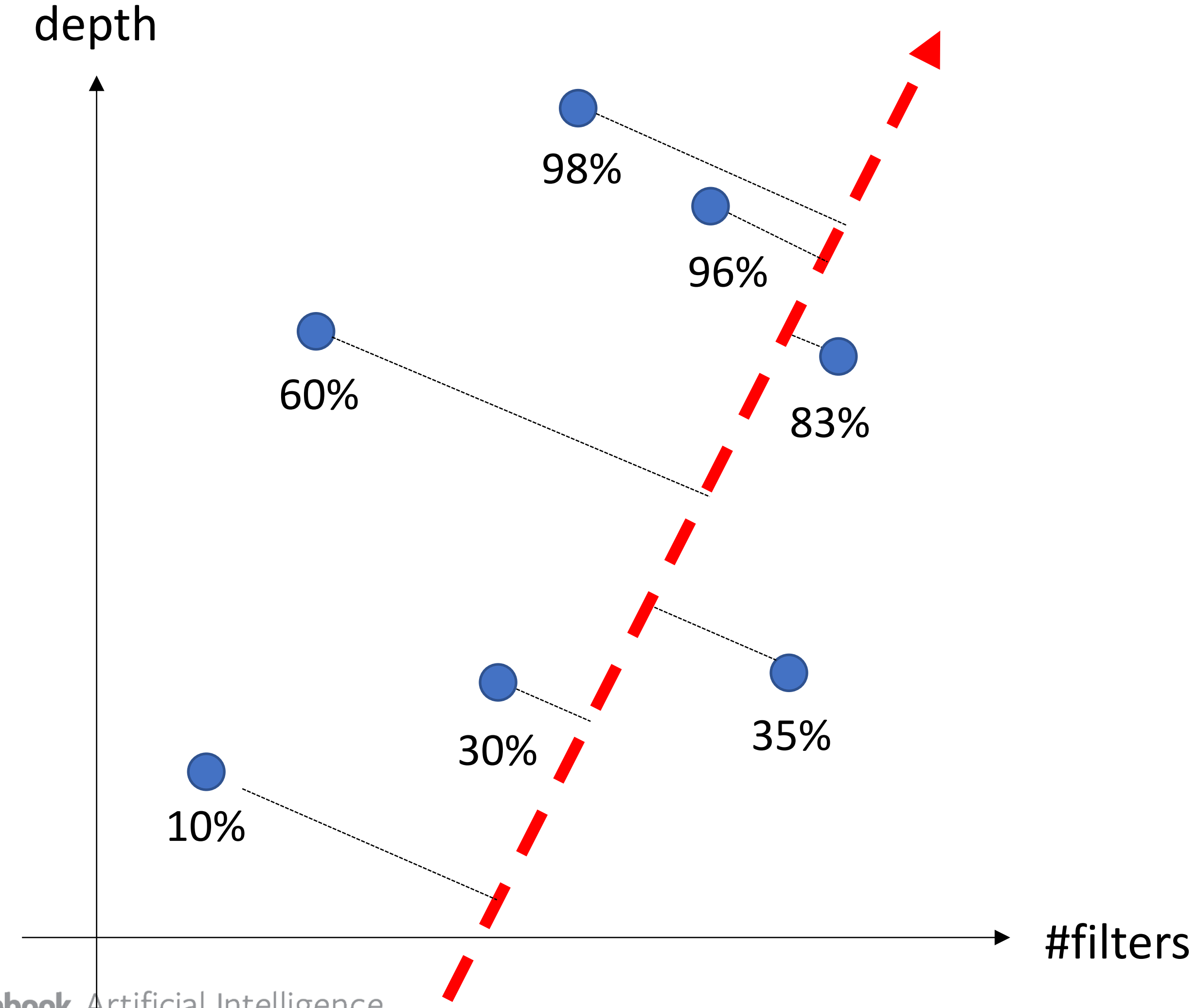


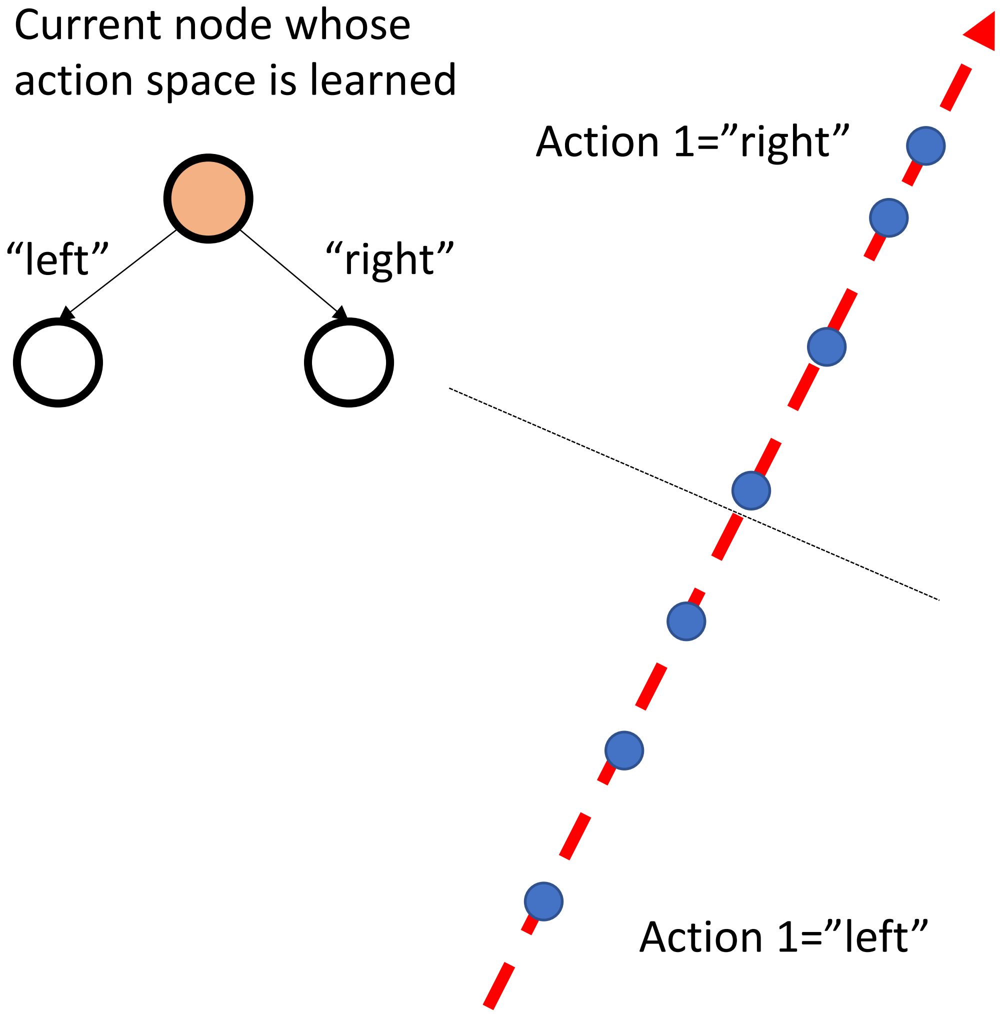
Empty network
Action: Add one conv layer
Action: More filters
Conv
Conv
ReLU
Conv
ReLU
Conv
ReLU
Full network
Evaluation
Current approaches



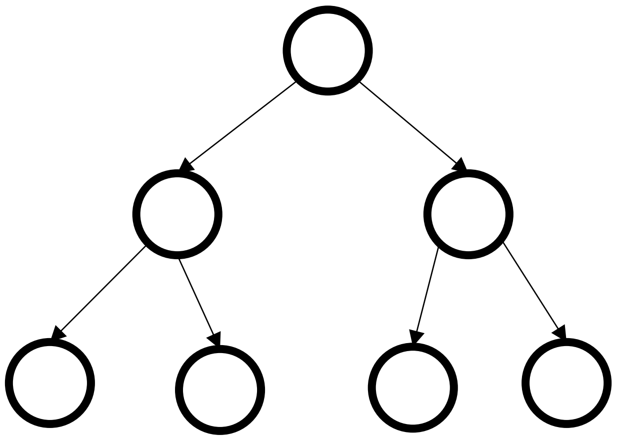
Entire network design space
Good models
Bad models
Action = Learnable Constraint
Subset of design space (lots of models)
Value function
Our approaches

Learn action space





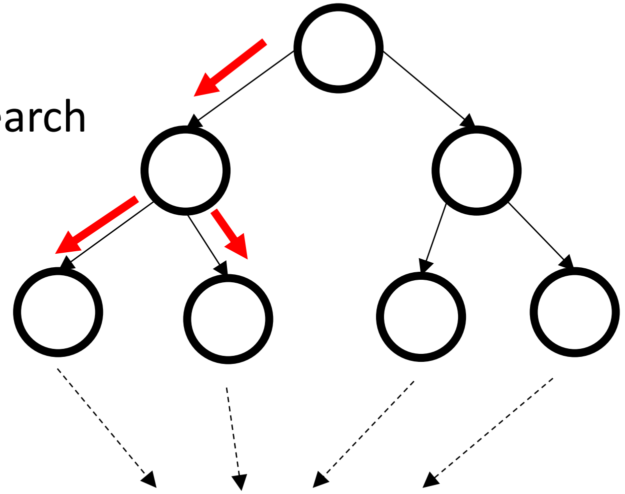
Approach



Fixed action branches
(but not action space)

(a) Search using current action space until a fixed #rollouts are used.

Monte Carlo Tree Search
(MCTS)



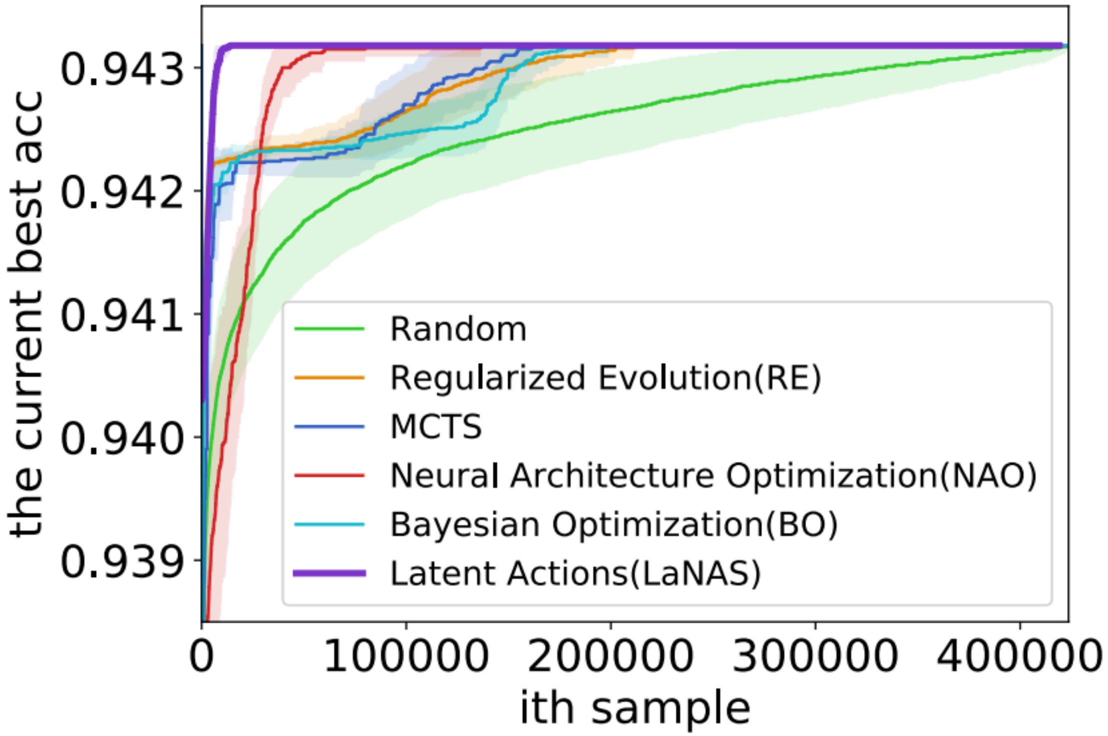
Value sampled from the
current subset of networks
(E.g., from truth table)

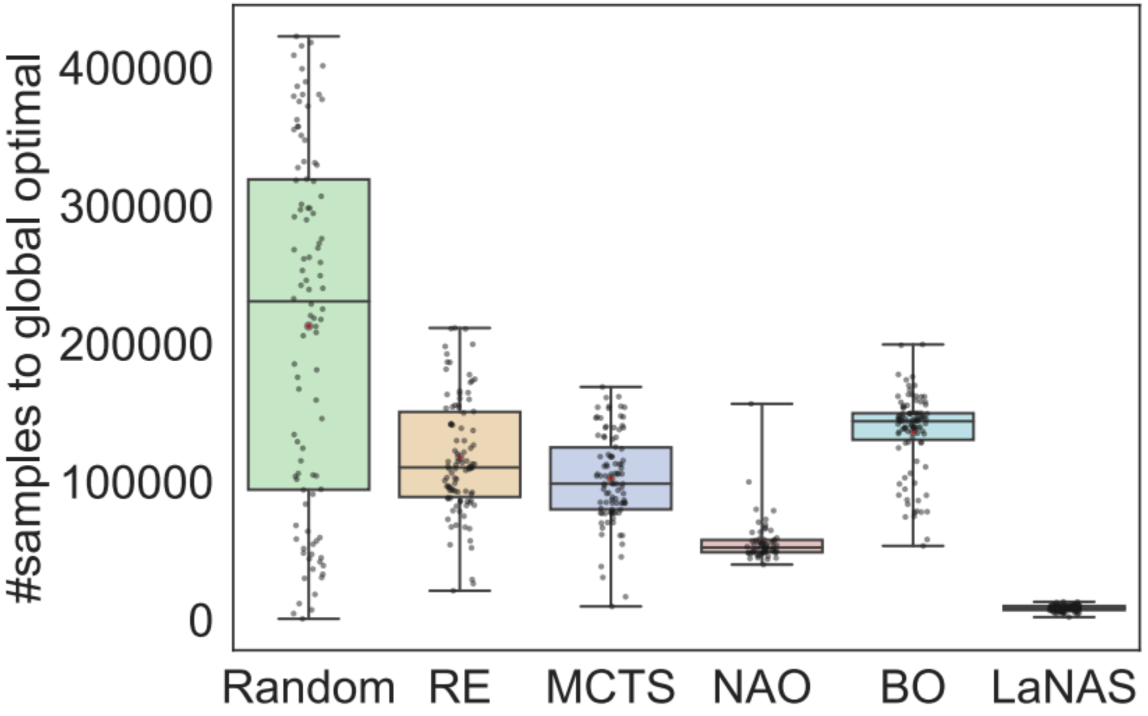
(b) Train the action space.

Network Hyperparameters	Accuracy
(filter=2, depth=5)	85%
(filter=3, depth=7)	92%
(filter=3, depth=2)	30%

Performance

NASBench-101 (CIFAR-10, 420k models, NASNet Search Space)

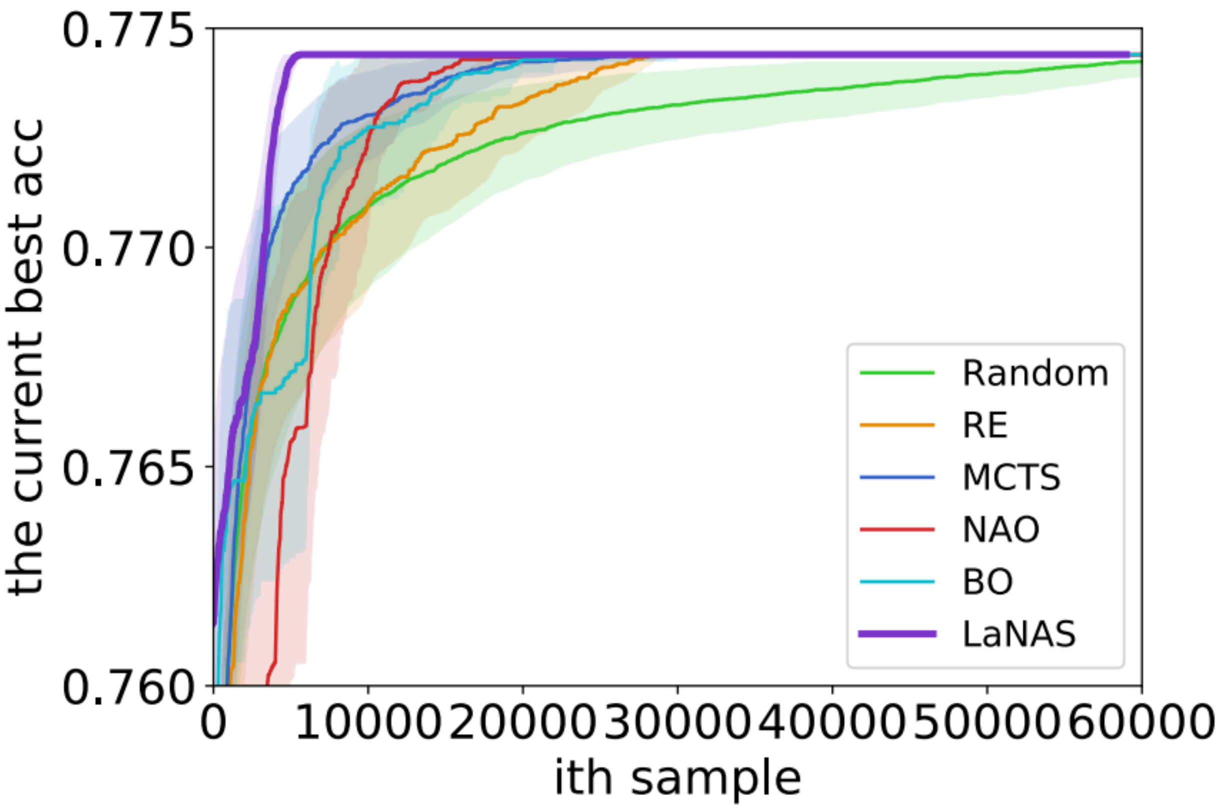


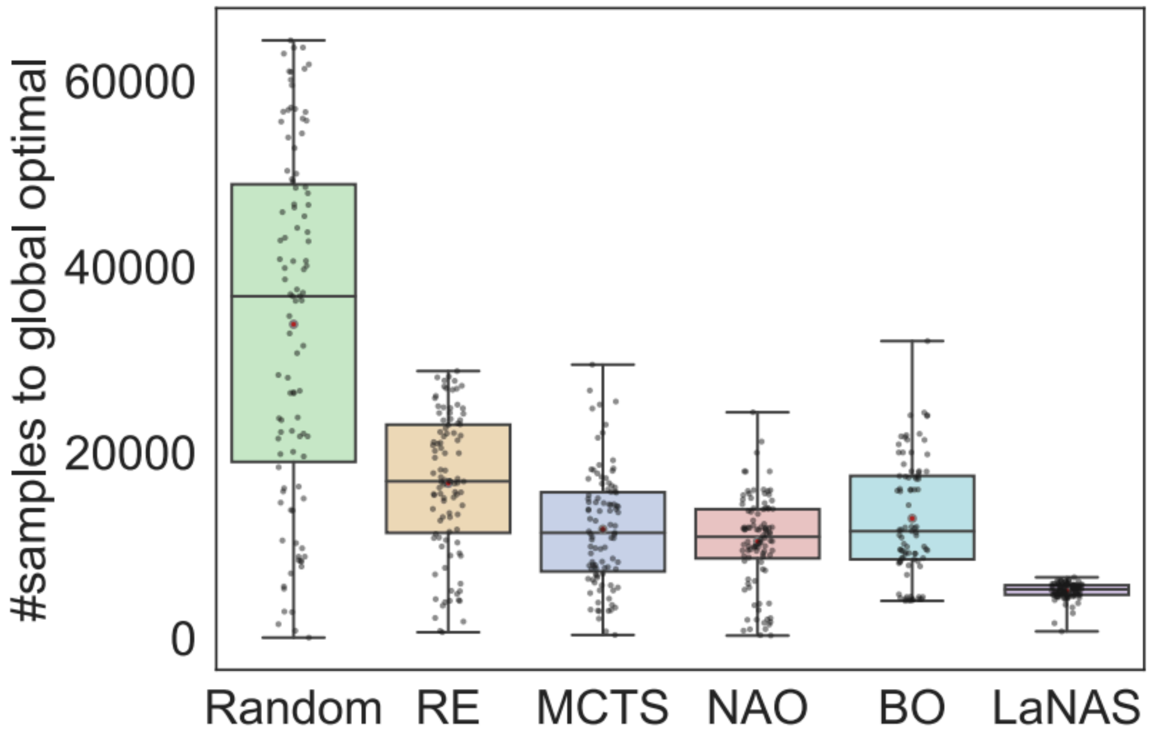


Each curve is repeated 100 times. We randomly pick 2k models to initialize.

Performance

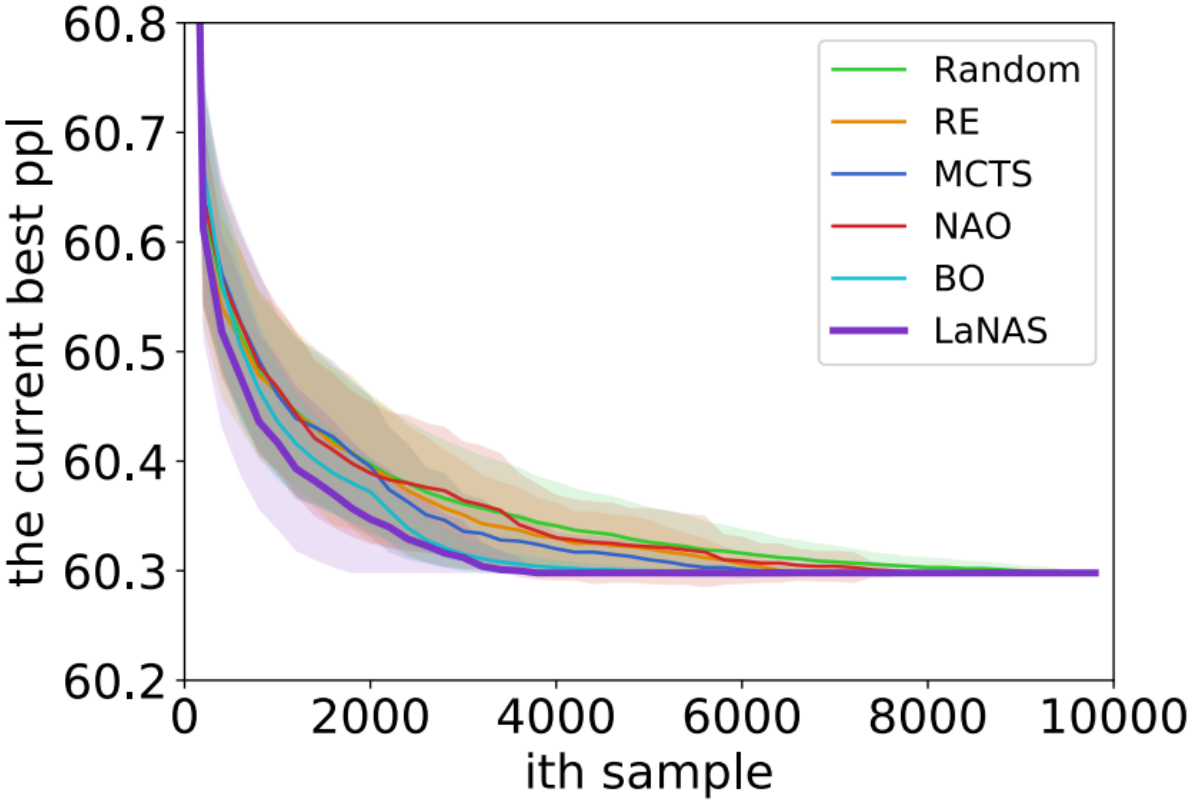
Customized dataset: ConvNet-60K (CIFAR-10, VGG style models)

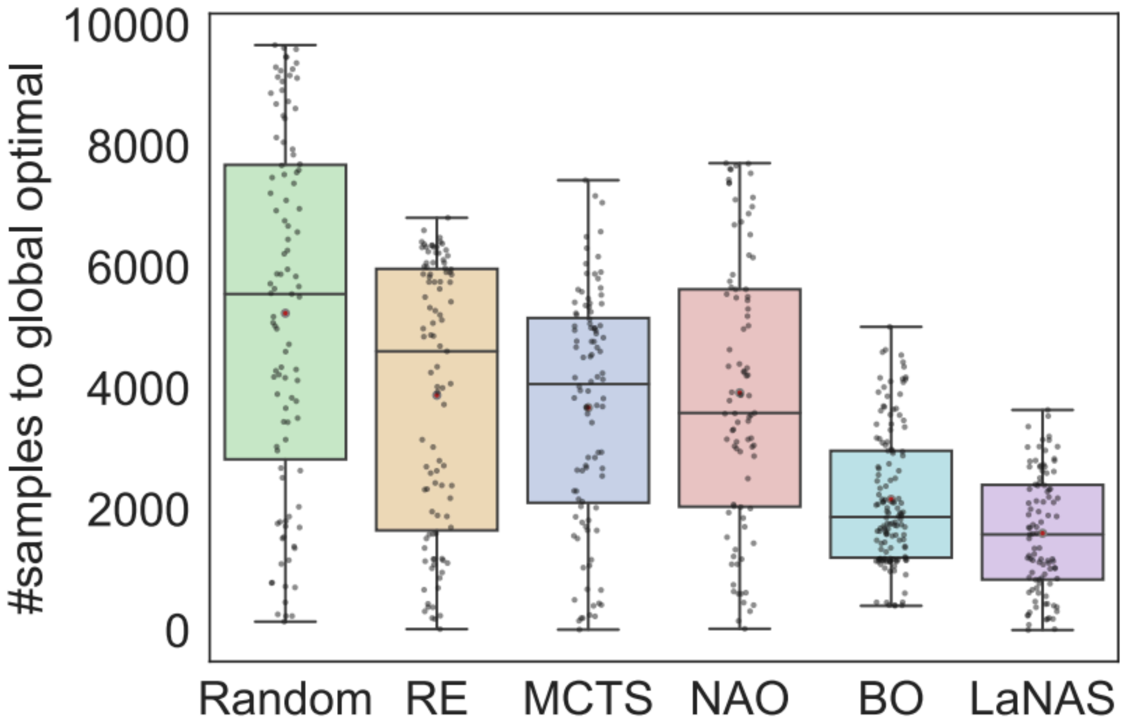




Performance

Customized dataset: LSTM-10K (PTB)





Open Domain

CIFAR-10 (NASNet style architecture)

Model	Using ImageNet	Params	Top1 err	M	GPU days
search based methods					
NASNet-A+c/o [22]	X	3.3 M	2.65	20000	2000
AmoebaNet-B+c/o [10]	X	2.8 M	$2.55_{\pm 0.05}$	27000	3150
PNASNet-5 [29]	X	3.2 M	$3.41_{\pm 0.09}$	1160	225
NAO+c/o [30]	X	128.0 M	2.11	1000	200
AmoebaNet-B+c/o	X	34.9 M	$2.13_{\pm 0.04}$	27000	3150
EfficientNet-B7	√	64M	1.01		
BiT-M	√	60M	1.09		
LaNet+c/o	X	3.2 M	$\mathbf{1.63}_{\pm 0.05}$	800	150
LaNet+c/o	X	44.1 M	$\mathbf{0.99}_{\pm 0.02}$	800	150
one-shot NAS based methods					
ENAS+c/o [18]	X	4.6 M	2.89	-	0.45
DARTS+c/o [20]	X	3.3 M	$2.76_{\pm 0.09}$	-	1.5
BayesNAS+c/o [31]	X	3.4 M	$2.81_{\pm 0.04}$	-	0.2
ASNG-NAS+c/o [32]	X	3.9 M	$2.83_{\pm 0.14}$	-	0.11
XNAS+c/0 [33]	X	3.7 M	1.81		0.3
oneshot-LaNet+c/o	X	3.6 M	$\mathbf{1.68}_{\pm 0.06}$	-	3
oneshot-LaNet+c/o	X	45.3 M	$\mathbf{1.2}_{\pm 0.03}$	-	3

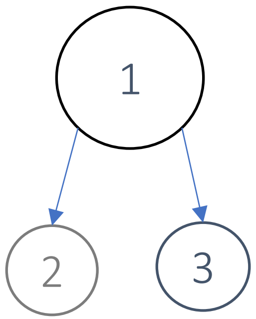
M: number of samples selected.

Open Domain

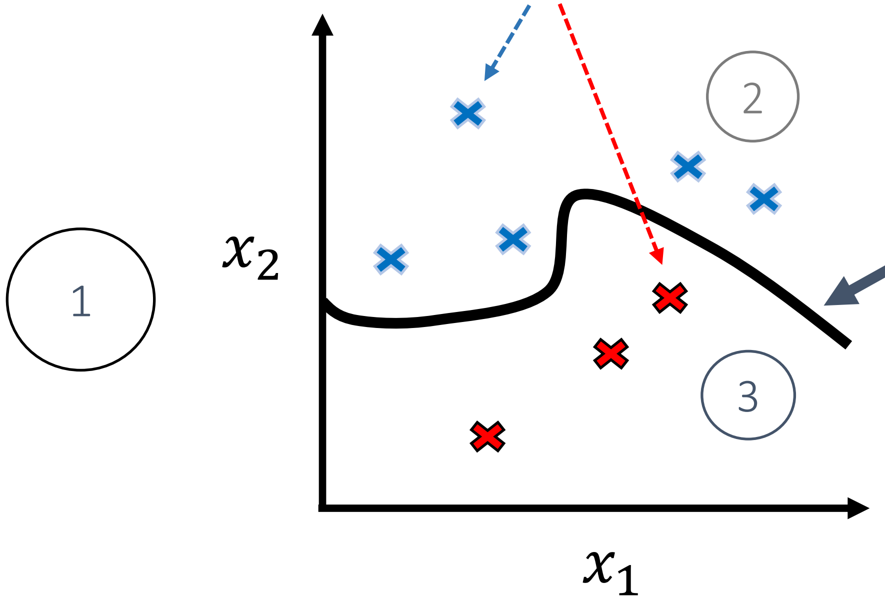
ImageNet
(mobile setting
Flop < 600M)

Model	FLOPs	Params	top1 / top5 err
NASNet-A (Zoph et al. (2018))	564M	5.3 M	26.0 / 8.4
NASNet-B (Zoph et al. (2018))	488M	5.3 M	27.2 / 8.7
NASNet-C (Zoph et al. (2018))	558M	4.9 M	27.5 / 9.0
AmoebaNet-A (Real et al. (2018))	555M	5.1 M	25.5 / 8.0
AmoebaNet-B (Real et al. (2018))	555M	5.3 M	26.0 / 8.5
AmoebaNet-C (Real et al. (2018))	570M	6.4 M	**24.3 / 7.6**
PNASNet-5 (Liu et al. (2018a))	588M	5.1 M	25.8 / 8.1
DARTS (Liu et al. (2018b))	574M	4.7 M	26.7 / 8.7
FBNet-C (Wu et al. (2018))	375M	5.5 M	25.1 / -
RandWire-WS (Xie et al. (2019))	583M	5.6 M	25.3 / 7.8
BayesNAS (Zhou et al. (2019))	-	3.9 M	26.5 / 8.9
LaNet	570M	5.1 M	**25.0 / 7.7**

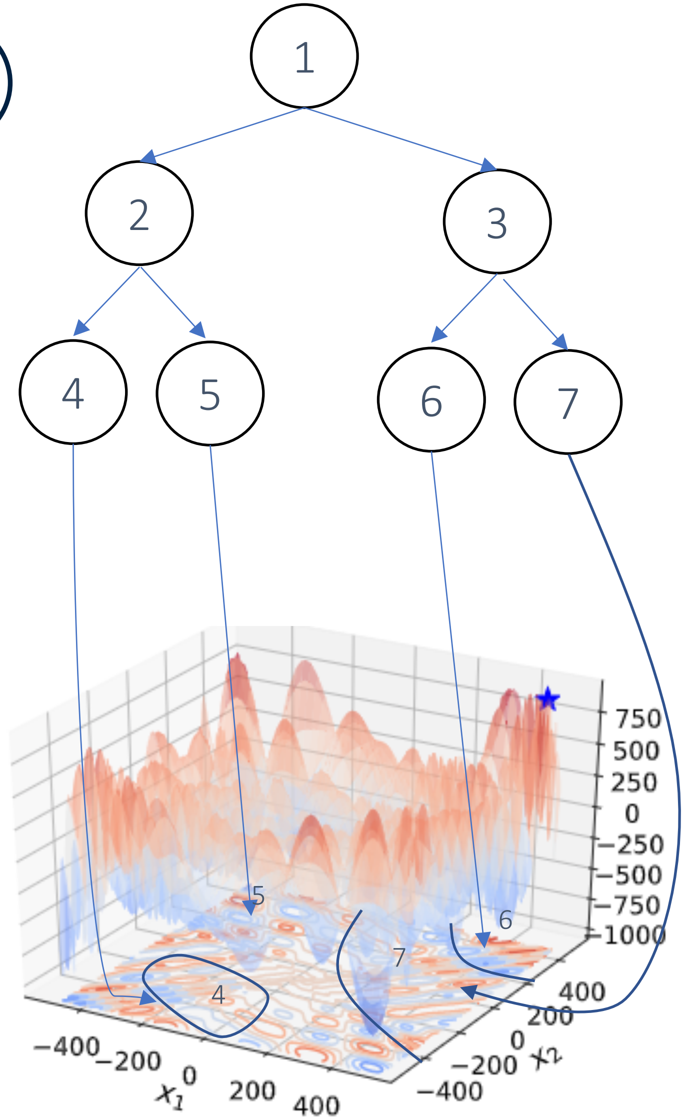
Black-Box Optimization (LaMCTS)



Build local models

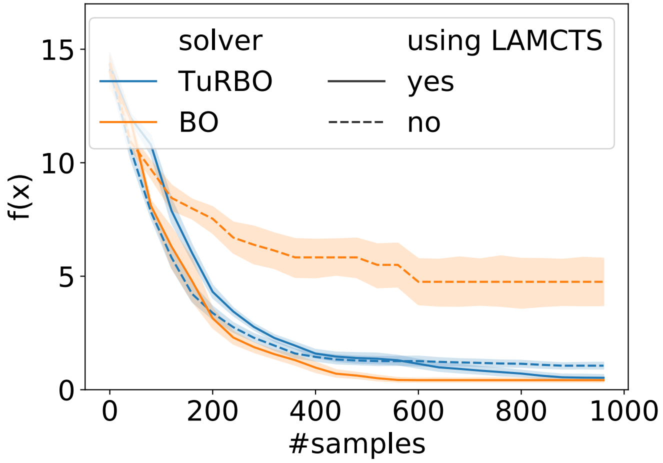


Nonlinear boundary
learnt by SVM

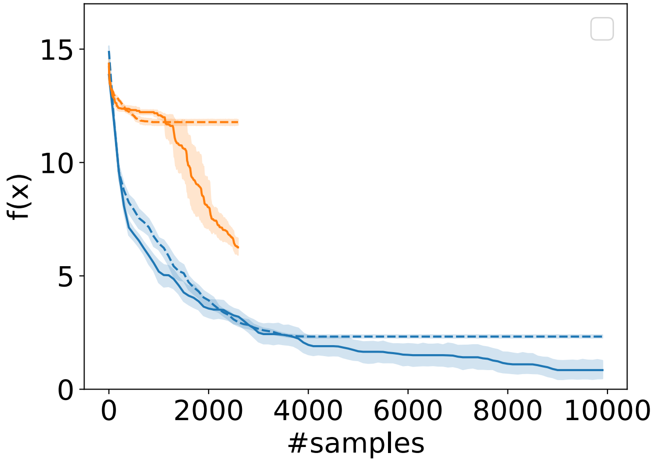


[L. Wang, et al, Learning Search Space Partition for Black-box Optimization using MCTS, NeurIPS 2020]

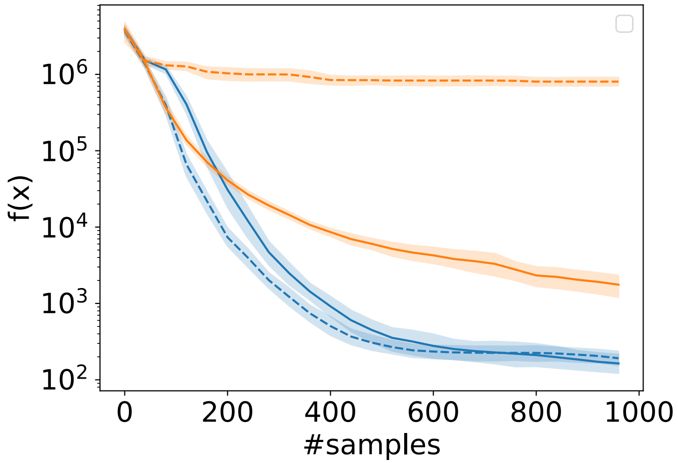
La-MCTS as a meta method



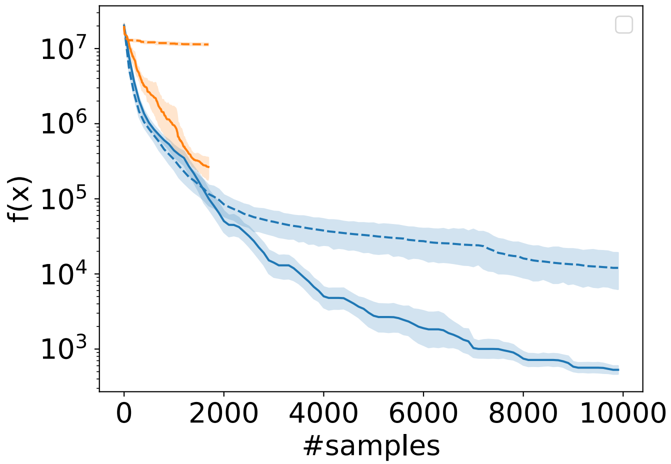
Ackley-20d



Ackley-100d

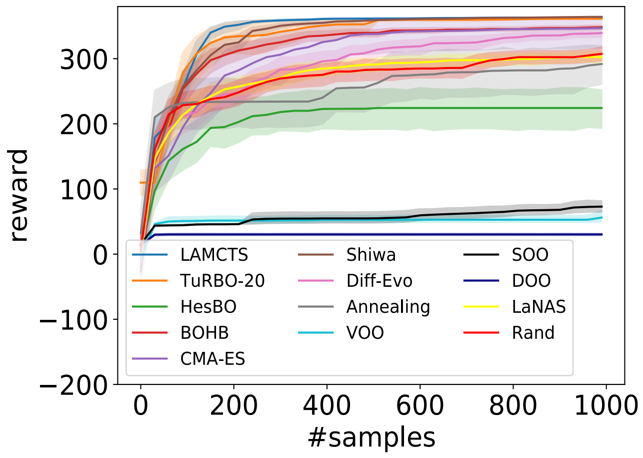


Rosenbrock-20d

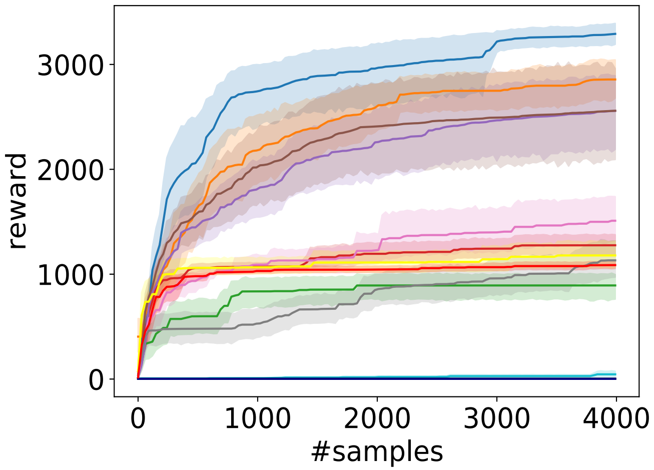


Rosenbrock-100d

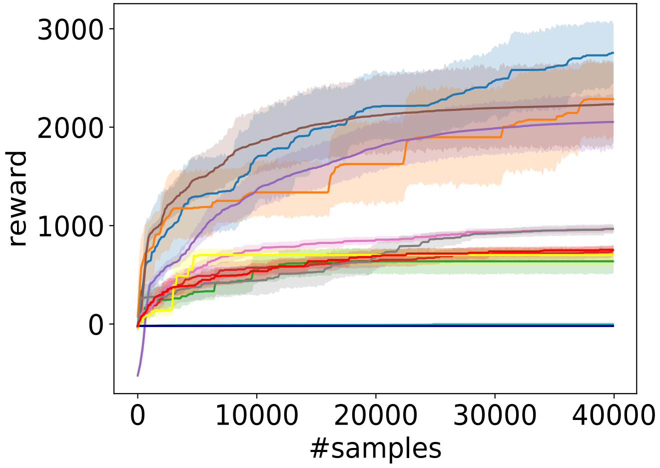
Optimizing linear policy for Mujoco tasks



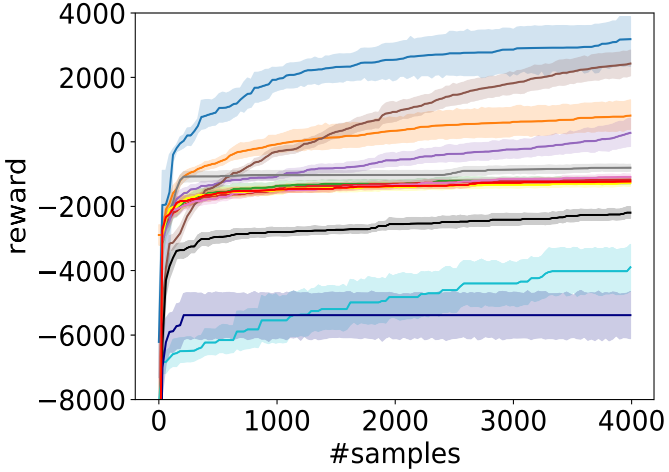
(a) Swimmer, #params = 16



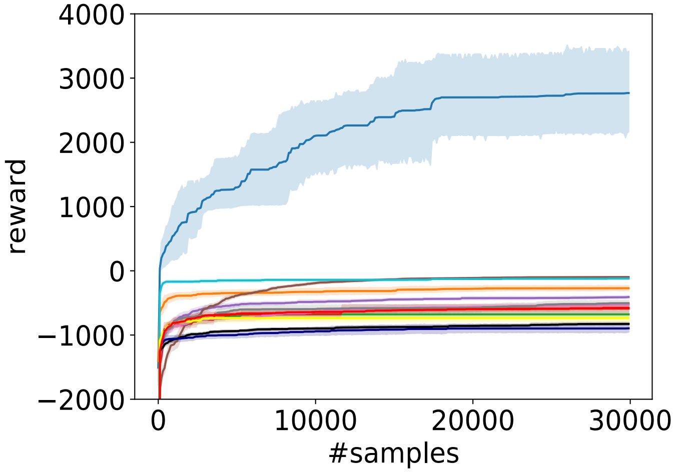
(b) Hopper, #params = 33



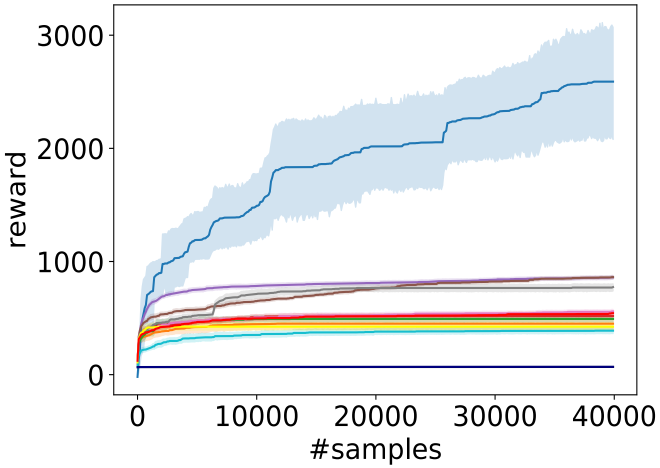
(c) Walker-2d, #params = 102



(d) Half-Cheetah, #params =102



(e) Ant, #params = 888



(f) Humanoid, #params = 6392

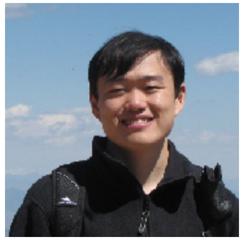
TODO: A theory is needed …

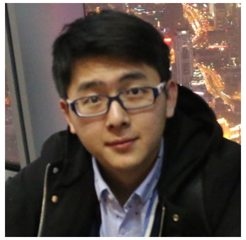
Principled framework
Demystify existing work

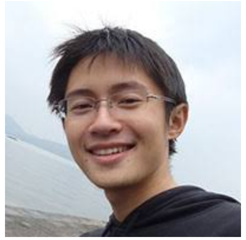
A theoretical framework that explains

1. Why **self-supervised learning** with deep ReLU models works
2. Why a good representation is learned without supervision
3. Why BYOL doesn't need negative samples

Understand Deep ReLU Models


Yuandong Tian

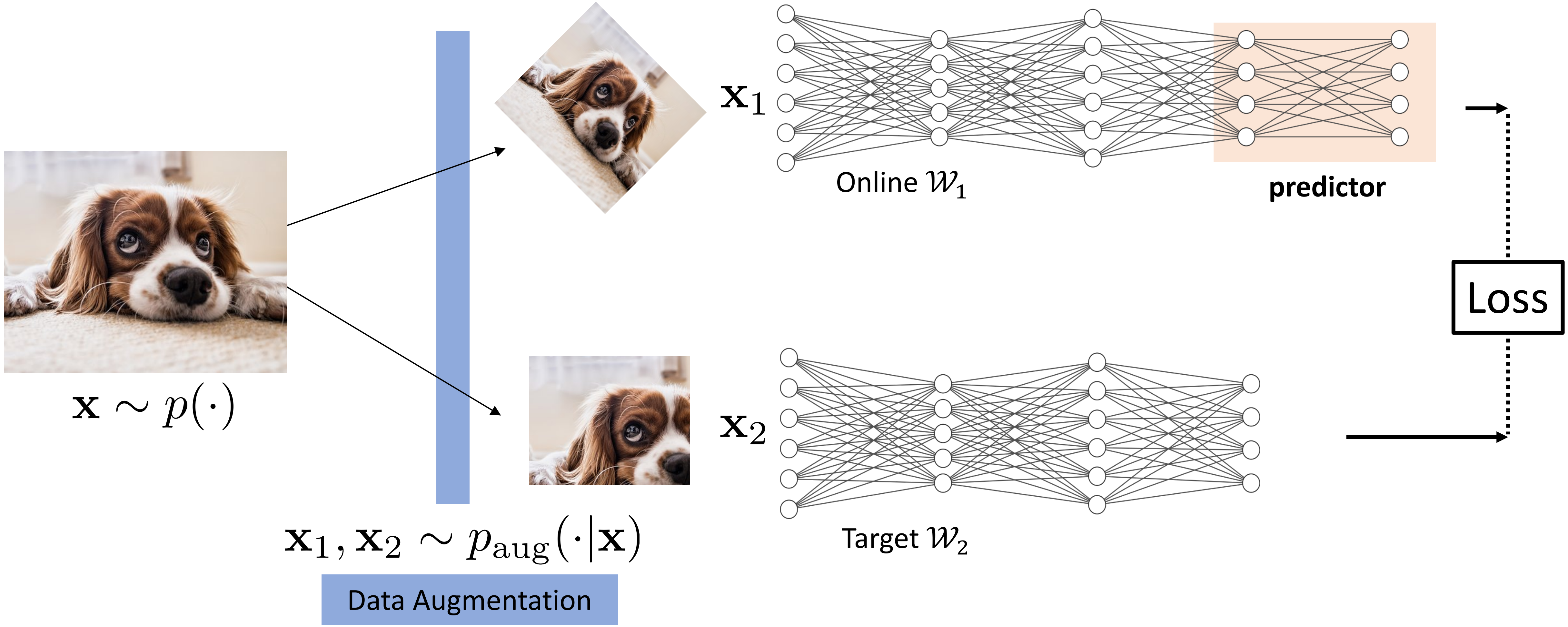

Lantao Yu


Xinlei Chen

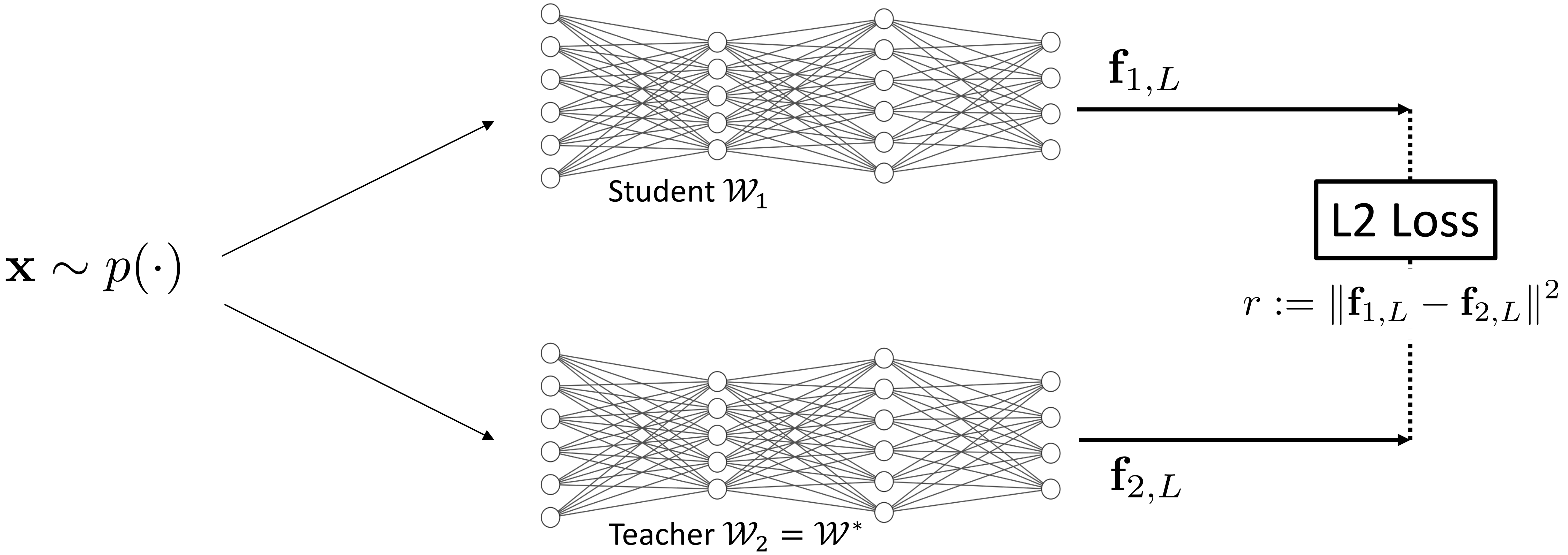

Surya Ganguli

[Y. Tian., Student Specialization in Deep ReLU Networks With Finite Width and Input Dimension, ICML 2020]
[Y. Tian et al., Understanding Self-supervised Learning with Dual Deep Networks, arXiv 2020]

Self-supervised Learning



Similarity with Teacher Student Setting



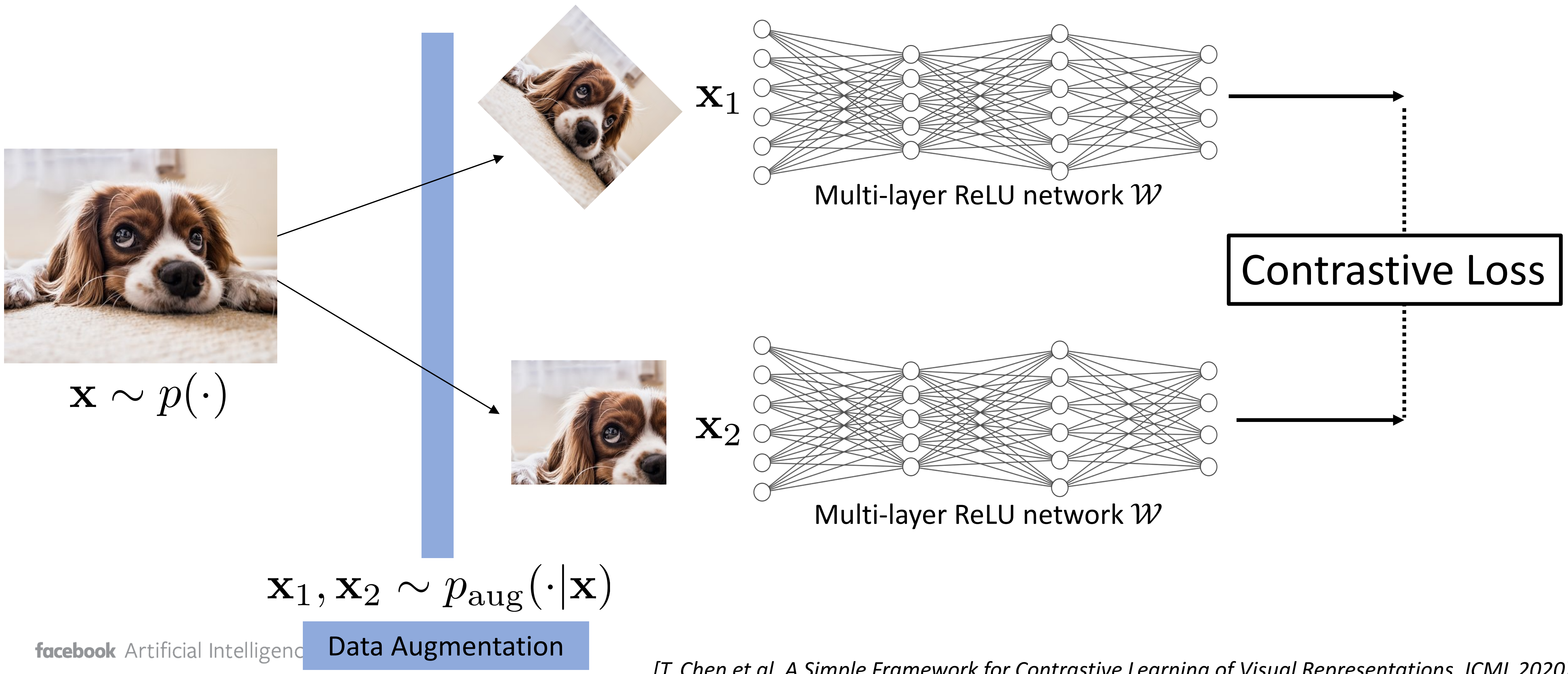
The mathematical framework is similar!

[Y. Tian, Student Specialization in Deep ReLU Networks With Finite Width and Input Dimension, ICML 2020]

Compare with Teacher-Student Setting

	Teacher-Student Setting	SimCLR Setting
Training Setup	Teacher is fixed and assumed to be optimal $\mathcal{W}^*$.	Teacher and student are both under training.
Loss function	L2 loss	Contrastive Loss
Data Augmentation	No	Yes (and critical)
Architectures	Same architecture for Teacher and Student	Same architecture for the two networks

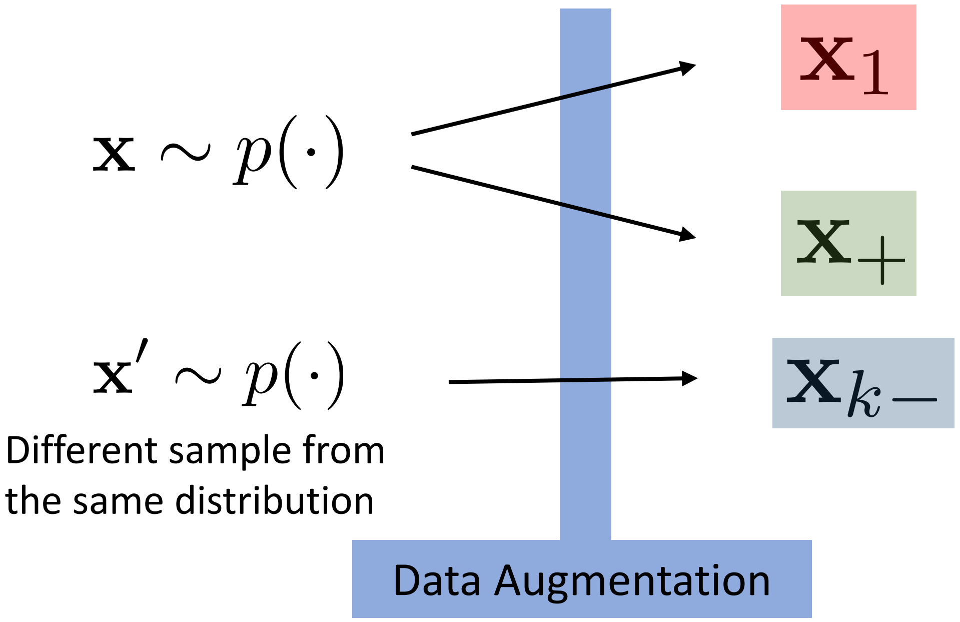
SimCLR Setting



[T. Chen et al, A Simple Framework for Contrastive Learning of Visual Representations, ICML 2020]

SimCLR

Data



Positive pair

$$r_+ := \left\|\mathbf{f}_{1,L} - \mathbf{f}_{+,L}\right\|^2$$

Negative pair

$$r_{k-} := \left\|\mathbf{f}_{1,L} - \mathbf{f}_{k-,L}\right\|^2$$

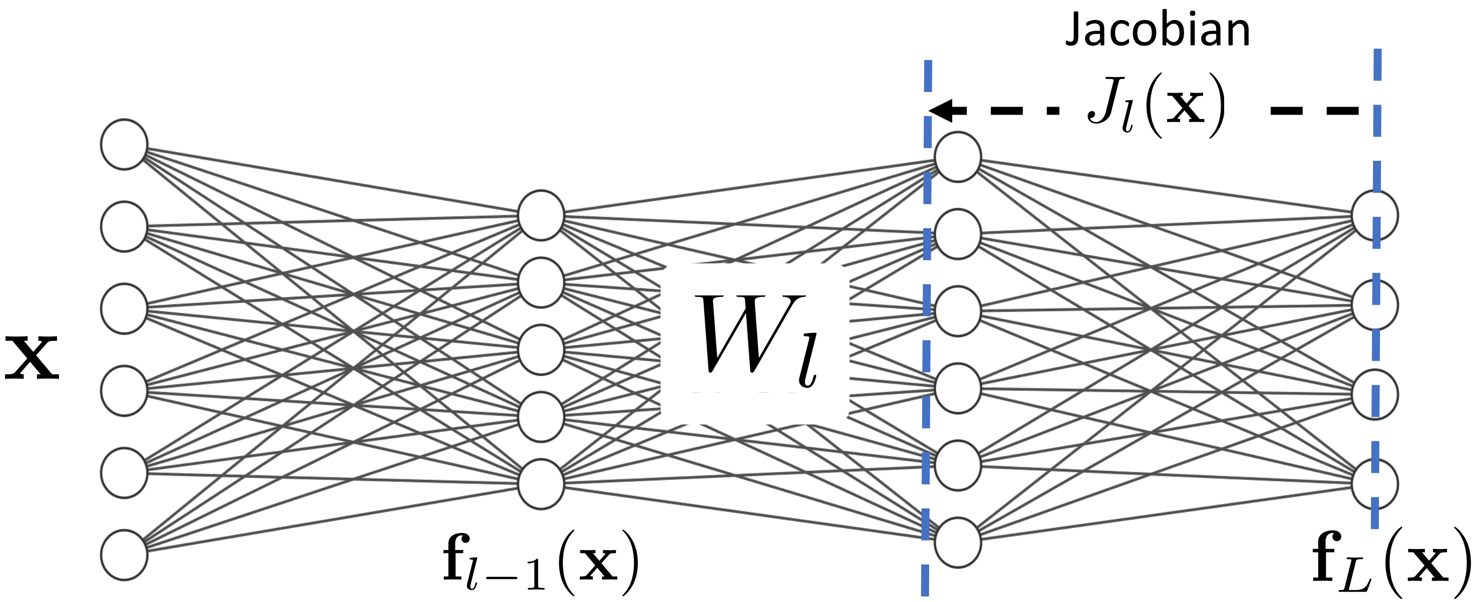
InfoNCE

$$L(r_+, r_{1-}, r_{2-}, \ldots, r_{K-}) \; := \; -\log \frac{e^{-r_+/\tau}}{e^{-r_+/\tau} + \sum_{k=1}^{H} e^{-r_{k-}/\tau}}$$

If $|u| = |v| = 1$, then the formulation is the same as SimCLR's formulation

Since $-r = -|u-v|^2 = 2\mathrm{sim}(u, v) - 2$

The Covariance Operator



Connection

$$K_l(\mathbf{x}) := \mathbf{f}_{l-1}(\mathbf{x}) \otimes J_l^{\intercal}(\mathbf{x})$$

$\otimes$: Kronecker Product

Augment-Average Connection

$$\bar{K}_l(\mathbf{x}) := \mathbb{E}_{\mathbf{x}' \sim p_{\text{aug}}(\cdot|\mathbf{x})}[K_l(\mathbf{x}')]$$

Weight Update for SimCLR at layer *l*:

$$W_l(t+1) = W_l(t) + \alpha \Delta W_l(t)$$

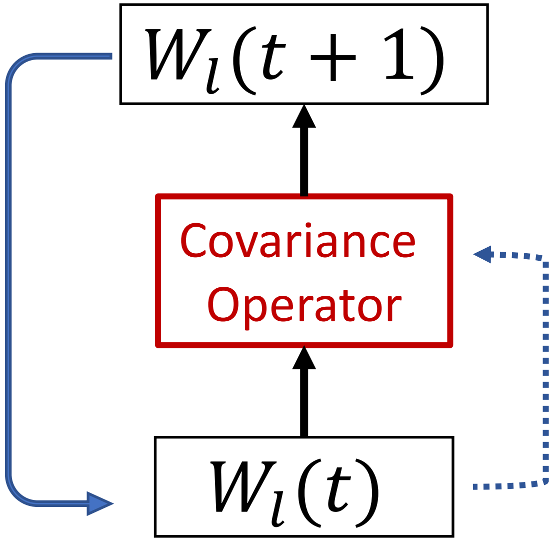
Learning rate

Covariance operator (PSD)

$$\text{vec}(\Delta W_l(t)) = \beta \mathbb{V}_{\boldsymbol{x}}[\bar{K}_l(\boldsymbol{x})] \text{vec}(W_l(t))$$

Positive number related to Contrastive loss

What does it mean?



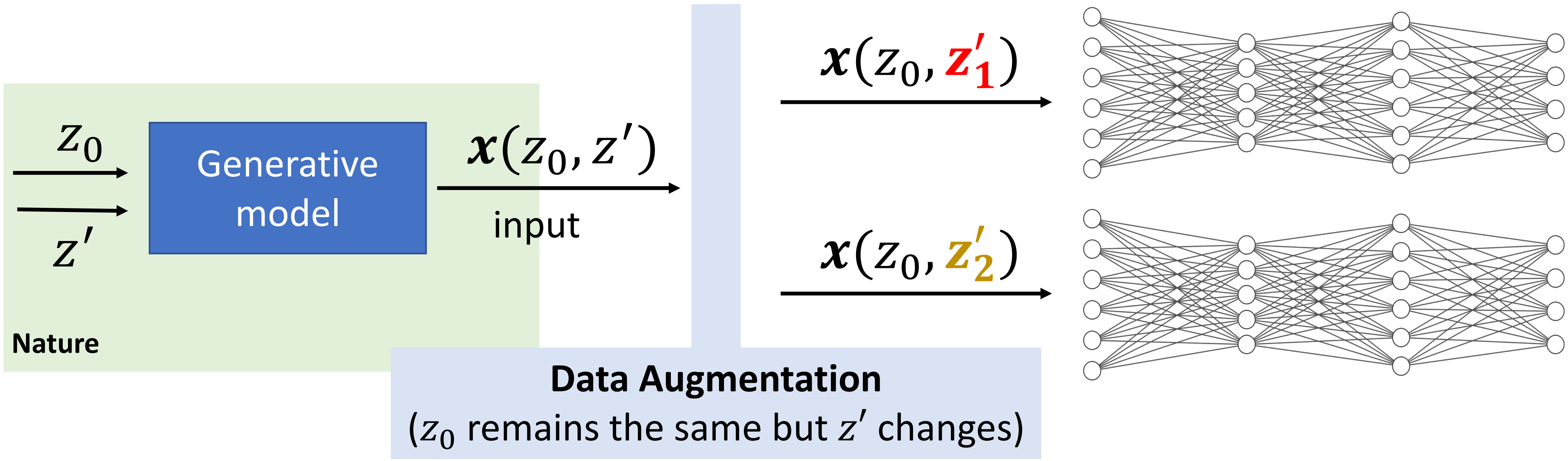
The Covariance Operator $\mathbb{V}_{\mathbf{x}}[\bar{K}_l(\mathbf{x}; \mathcal{W}(t))]$

- Always PSD at any stage of training
- Weight at each layer undergoes a PSD transformation
- Strong eigen mode leads strong weight growth along that direction

What are the strong eigen models in the covariance operator?

To understand that, we need a generative model of the data.

Using Generative Models to understand Covariance Operator



$\boldsymbol{z_0}$: Class (sample) label

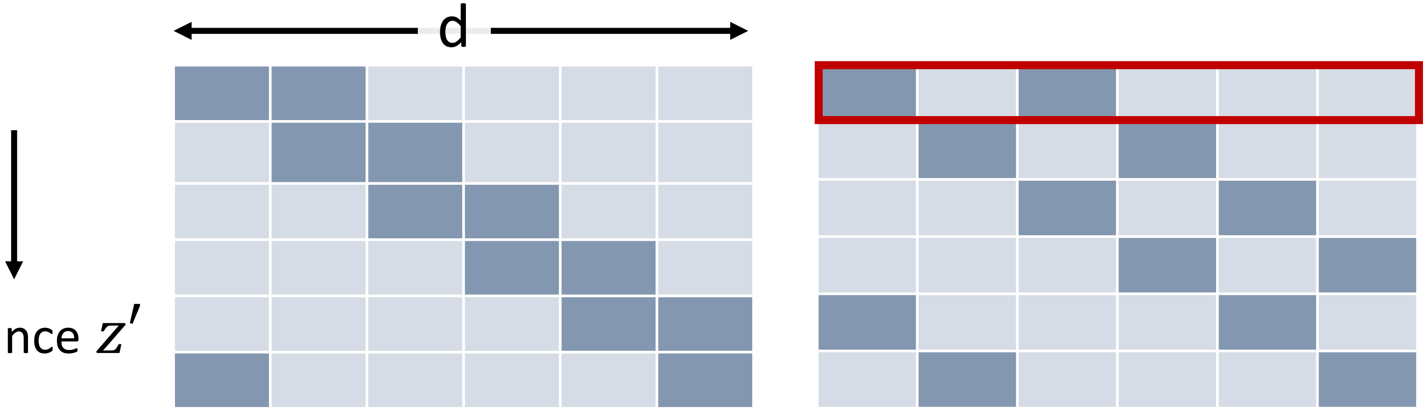
$\boldsymbol{z'}$: Nuisance Transformations given by Data Augmentation

One-layer one-neuron example

Two objects **11** and **101** translating in 1D space

d

Nuisance z'



$\boldsymbol{x}(z_0, z')$

$z_0 = 1$ $z_0 = 2$

$$\boxed{\mathbb{V}_{z_0}\left[\bar{K}(z_0)\right] = \frac{1}{4d^2}\mathbf{u}\mathbf{u}^\intercal}$$

$$\mathbf{u} := \mathbf{x}_{11} + \mathbf{x}_{00} - \mathbf{x}_{01} - \mathbf{x}_{10}$$

Feature to represent pattern 10

Linear neuron: Nothing is learned.

ReLU neuron: Enforce what is initialized!

A two-layer example

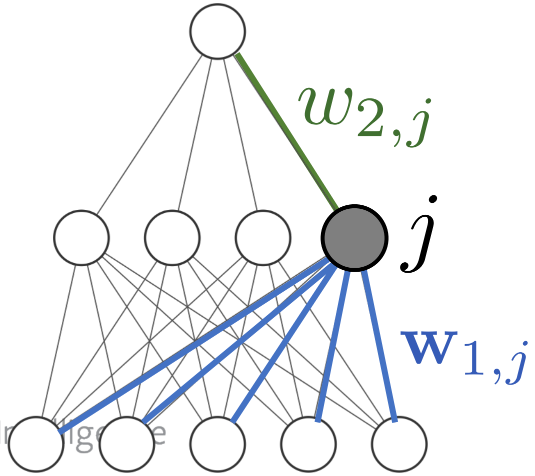
Augment-Average Connection for both layers:

$$\bar{K}_1(z) = [w_{2,1}\mathbf{u}_1, \ldots, w_{2,n_1}\mathbf{u}_{n_1}] \qquad \bar{K}_2(z) = [\mathbf{w}_{1,1}^\intercal\mathbf{u}_1, \ldots, \mathbf{w}_{1,n_1}^\intercal\mathbf{u}_{n_1}]$$

where $\boldsymbol{u}_j(z) := \mathbb{E}_{z'|z}\left[\boldsymbol{x}(z,z')\mathbb{I}(\boldsymbol{w}_{1,j}^\intercal\boldsymbol{x}(z,z') \geq 0)\right]$

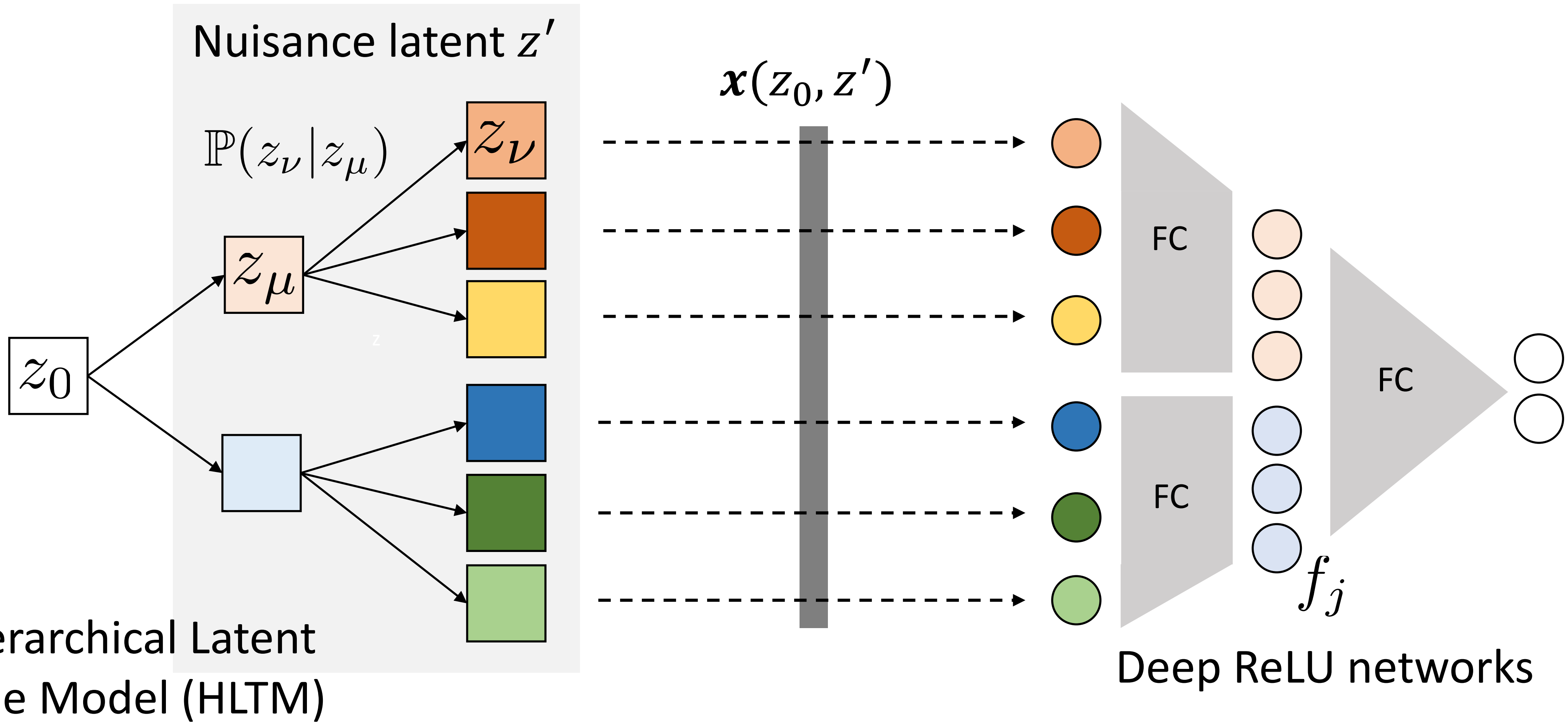
Theorem 4. *If* $\mathrm{Cov}_z\left[\boldsymbol{u}_j, \boldsymbol{u}_k\right] = 0$ *for* $j \neq k$*, then the time derivative of* $w_{2,j}$ *and* $\boldsymbol{w}_{1,j}$ *satisfies:*

$$\boxed{\dot{w}_{2,j} = w_{2,j}\boldsymbol{w}_{1,j}^\intercal A_j \boldsymbol{w}_{1,j}, \quad \dot{\boldsymbol{w}}_{1,j} = w_{2,j}^2 A_j \boldsymbol{w}_{1,j},} \quad \textit{where } A_j := \mathbb{V}_z[\boldsymbol{u}_j(z)].$$

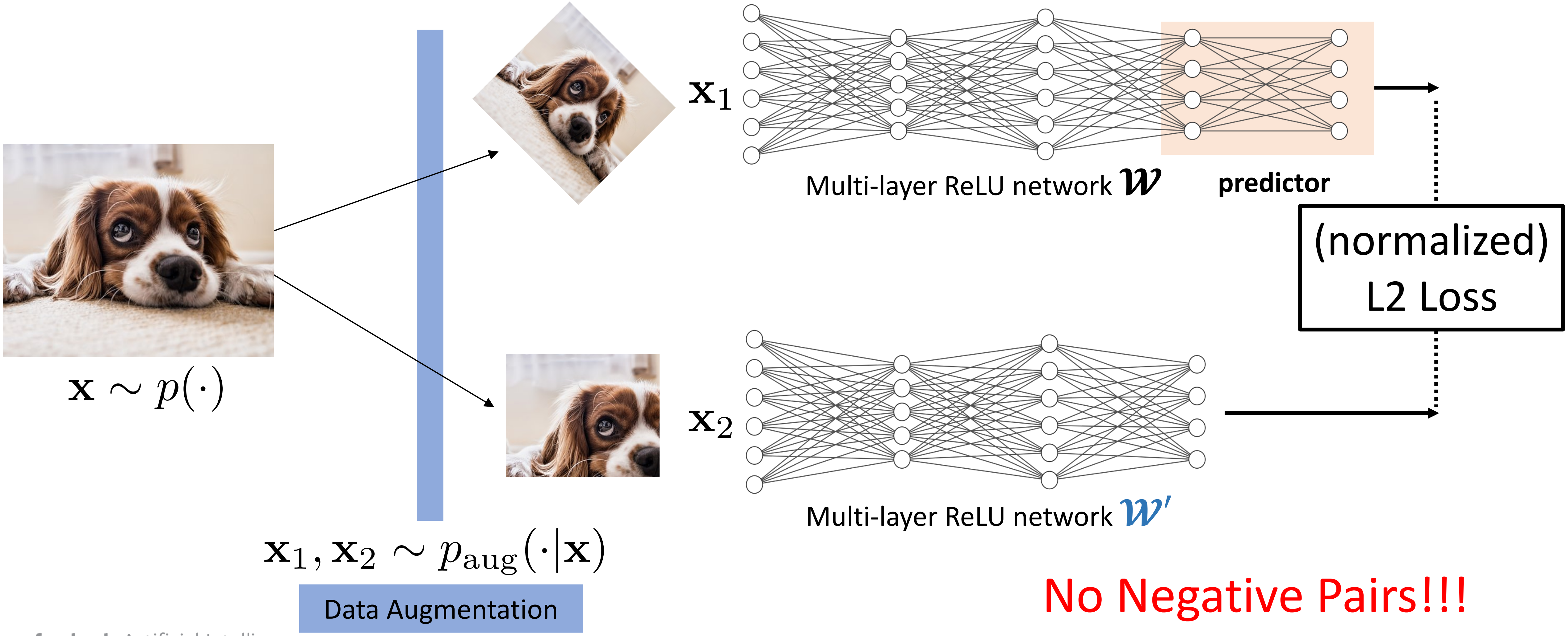


Weights of two layer are ***enforcing*** each other

Hierarchical Latent Tree Models (HLTM)



BYOL Setting



No Negative Pairs!!!

[J. Grill et al, Bootstrap your own latent: A new approach to self-supervised Learning, arXiv]

BYOL Setting

	SimCLR Setting	BYOL Setting
Loss function	Contrastive Loss	(Normalized) L2 Loss
Architectures	Symmetric $\mathcal{W}_1 = \mathcal{W}_2 = \mathcal{W}$	Different Architectures $\mathcal{W} \neq \mathcal{W}'$. $\mathcal{W}$ has an **extra** predictor ***(critical)*** $\mathcal{W}'$ might has Exponential Moving Average (EMA)
BatchNorm in predictor/projector	Optional	Must have BN in predictor/projector ***(critical)***

Why BYOL doesn't need contrastive loss?

Why BYOL needs an extra predictor?

Why BYOL needs to have BN in predictor/projector to work?

BYOL Setting (Top-1 Performance in STL-10)

Using Predictor is critical

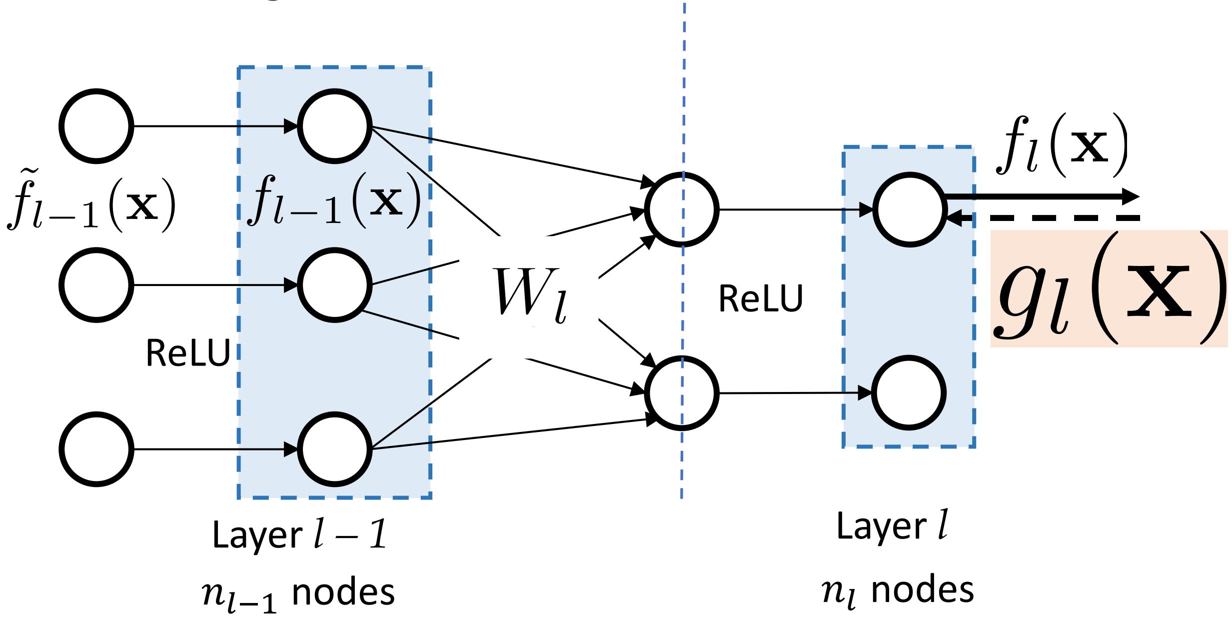
-	EMA	BN	EMA, BN
38.7 ± 0.6	39.3 ± 0.9	33.0 ± 0.3	32.8 ± 0.5

EMA is helping

BN is critical

P	P, EMA	P, BN	P, EMA, BN
39.5 ± 3.1	44.4 ± 3.2	63.6 ± 1.06	$\mathbf{78.1} \pm 0.3$

How to analyze BatchNorm?



Zero-mean property.

After BN, Backpropagated Gradient is zero-mean in each minibatch:

$$\tilde{\boldsymbol{g}}_l^i := \boldsymbol{g}_l^i - \frac{1}{|B|}\sum_{i\in B}\boldsymbol{g}_l^i = \boldsymbol{g}_l^i - \bar{\boldsymbol{g}}_l$$

Zero-mean Gradient matters.

Ablation Study of Batch components

-	μ	σ	μ, σ	$\mu^{\nparallel}$
43.9 ± 4.2	64.8 ± 0.6	72.2 ± 0.9	$\mathbf{78.1} \pm 0.3$	44.2 ± 7.0

$\sigma^{\nparallel}$	$\mu^{\nparallel}, \sigma$	$\mu, \sigma^{\nparallel}$	$\mu^{\nparallel}, \sigma^{\nparallel}$
54.2 ± 0.6	48.3 ± 2.7	76.3 ± 0.4	47.0 ± 8.1

μ x = x - x.mean(0)

σ x = x / x.std(0)

$\mu^{\nparallel}$ x = x - x.mean(0).detach()

$\sigma^{\nparallel}$ x = x / x.std(0).detach()

Explanation with the Framework

Without BN or $\mathcal{W} = \mathcal{W}'$

$$\begin{aligned}\mathrm{vec}\left(\Delta W_l\right) &= \mathrm{vec}(\Delta W_l)_{\mathrm{sym}} \\ &- \mathbb{E}_{\boldsymbol{x}}\left\{\bar{K}_l(\boldsymbol{x})\left[\bar{K}_l^{\intercal}(\boldsymbol{x})\mathrm{vec}(W_l) - \bar{K}_l^{\intercal}(\boldsymbol{x};\mathcal{W}')\mathrm{vec}(W_l')\right]\right\}\end{aligned}$$

With BN and $\mathcal{W} \neq \mathcal{W}'$

$$\begin{aligned}\mathrm{vec}\left(\widetilde{\Delta W_l}\right) &= \mathrm{vec}(\Delta W_l)_{\mathrm{sym}} \\ &- \mathbb{V}_{\boldsymbol{x}}\left[\bar{K}_l(\boldsymbol{x})\right]\mathrm{vec}(W_l) + \mathrm{Cov}_{\boldsymbol{x}}\left[\bar{K}_l(\boldsymbol{x}), \bar{K}_l(\boldsymbol{x};\mathcal{W}')\right]\mathrm{vec}(W_l')\end{aligned}$$

*Some assumption is need to get to here, see paper for the details.

$$\mathrm{vec}(\Delta W_l)_{\mathrm{sym}} = -\mathbb{E}_{\boldsymbol{x}}\left\{\mathbb{V}_{\boldsymbol{x}'}[K_l(\boldsymbol{x}')]\right\}\mathrm{vec}(W_l)$$

Why BatchNorm and Predictor matters

$$-\mathbb{V}_{\boldsymbol{x}}\left[\bar{K}_l(\boldsymbol{x};\mathcal{W})\right]\mathrm{vec}(W_l) + \mathrm{Cov}_{\boldsymbol{x}}\left[\bar{K}_l(\boldsymbol{x};\mathcal{W}), \bar{K}_l(\boldsymbol{x};\mathcal{W}')\right]\mathrm{vec}(W_l')$$

Negated covariance operator

Approximate covariance operator

Small when there is a predictor in $\mathcal{W}$ with small Jacobian

Reinitializing Predictors Works

Table 5: Top-1 performance of BYOL using reinitialization of the predictor every T epochs.

	Original BYOL	ReInit $T = 5$	ReInit $T = 10$	ReInit $T = 20$
STL-10 (100 epochs)	78.1	78.6	**79.1**	79.0
ImageNet (60 epochs)	60.9	61.9	**62.4**	**62.4**

The predictor is not necessarily "optimal" as suggested in the original BYOL paper.

Homework

- What's the best mass ratio in Black powder?
- Is that possible to enumerate all possible states in a game like Go?
- How does AlphaZero work? Does AlphaZero use human knowledge?
- Explain how Monte Carlo Tree Search works?
- Explain how Alpha Beta Pruning works?
- Why do we want to open the black-box for deep models?

Thanks!