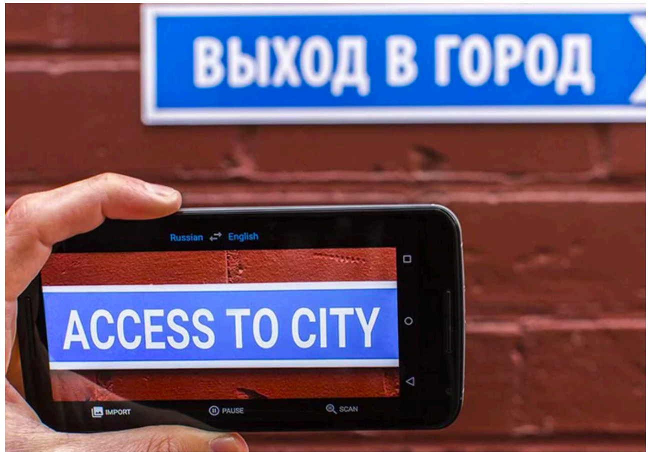
Over-parameterization as a **Catalyst** for Better Generalization of Deep ReLU network

Yuandong Tian

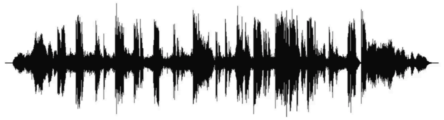
Research Scientist and Manager
Facebook AI Research

Great Empirical Success from Deep Models

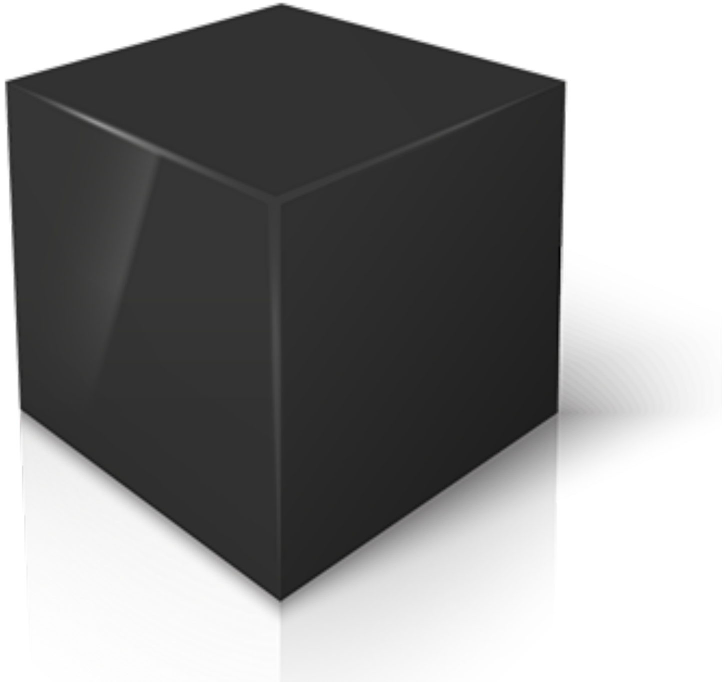


How do deep models work?

Input



This is an apple



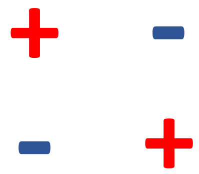
"Some Nonlinear Transformation"

Output

Three Major Problems

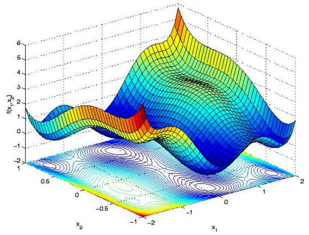
Understanding how Deep Models work

Expressibility



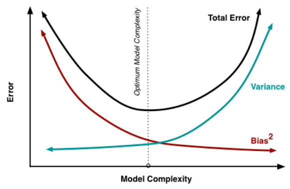
"Neural Network is a universal approximator"
"Deep Models can express functions more efficiently than shallow ones"

Optimization



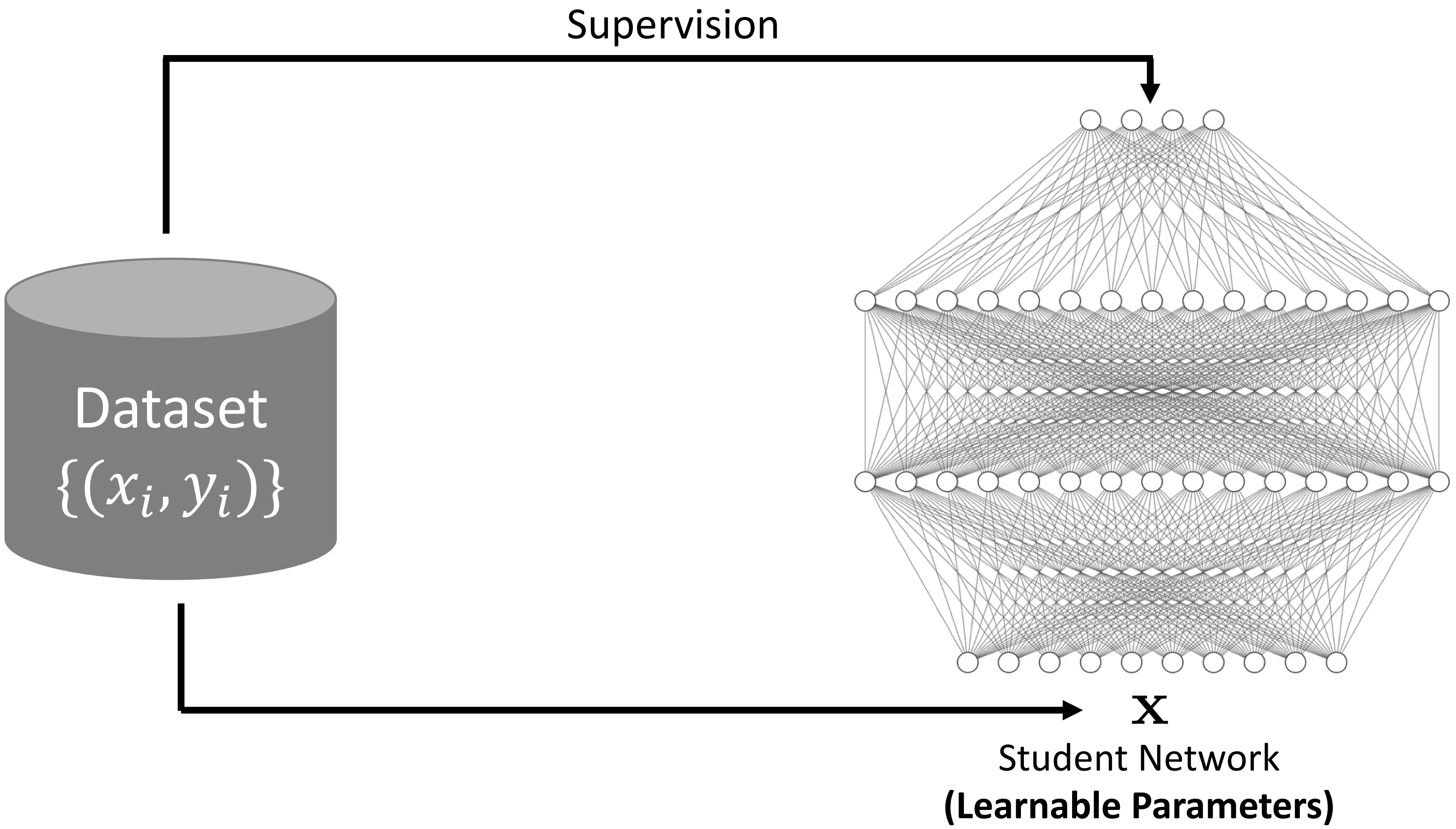
"Gradient vanishing/exploding"
"Gradient Descent might get stuck at saddle point / local minima"
"Can GD/SGD go to global optima? How fast?"

Generalization

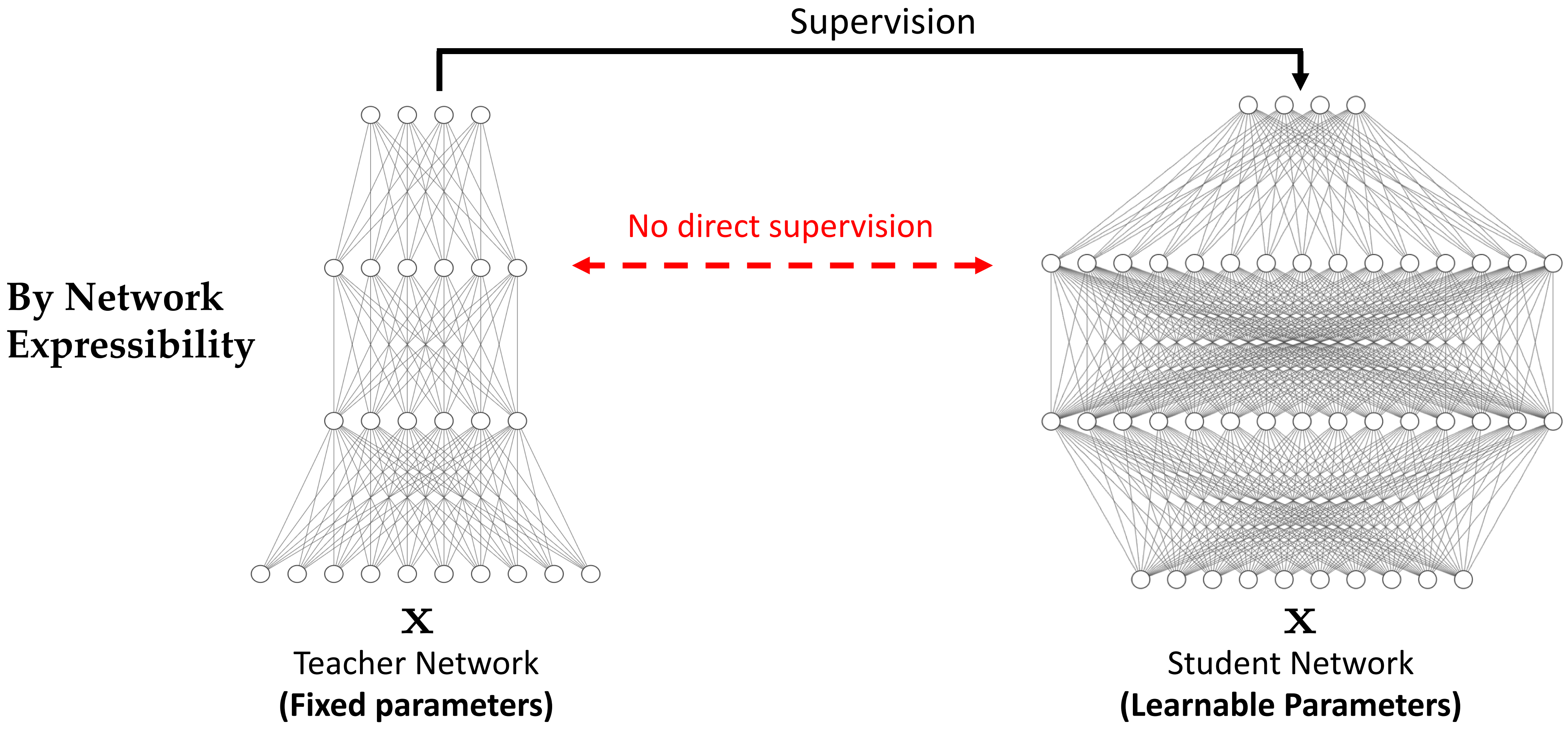


"Does zero training error often lead to overfitting?"
"More parameters might lead to overfitting."

Supervised Learning



Student-Teacher Setting



Why Student-Teacher Setting?

Understanding how Deep Models work

Expressibility

Provide a target function with bounded complexity

Optimization

Study fine dynamics behaviors by comparing with teacher

Generalization

Weight alignment with the teacher yields generalization

Old History of Teacher-Student Setting

$$\epsilon(\mathbf{J}, \boldsymbol{\xi}) \equiv \frac{1}{2} [\ \sigma(\mathbf{J}, \boldsymbol{\xi}) - \zeta\ ]^2 = \frac{1}{2} \left[\sum_{i=1}^{K} g(x_i) - \sum_{n=1}^{M} g(y_n) \right]^2$$

One layer of trainable parameters

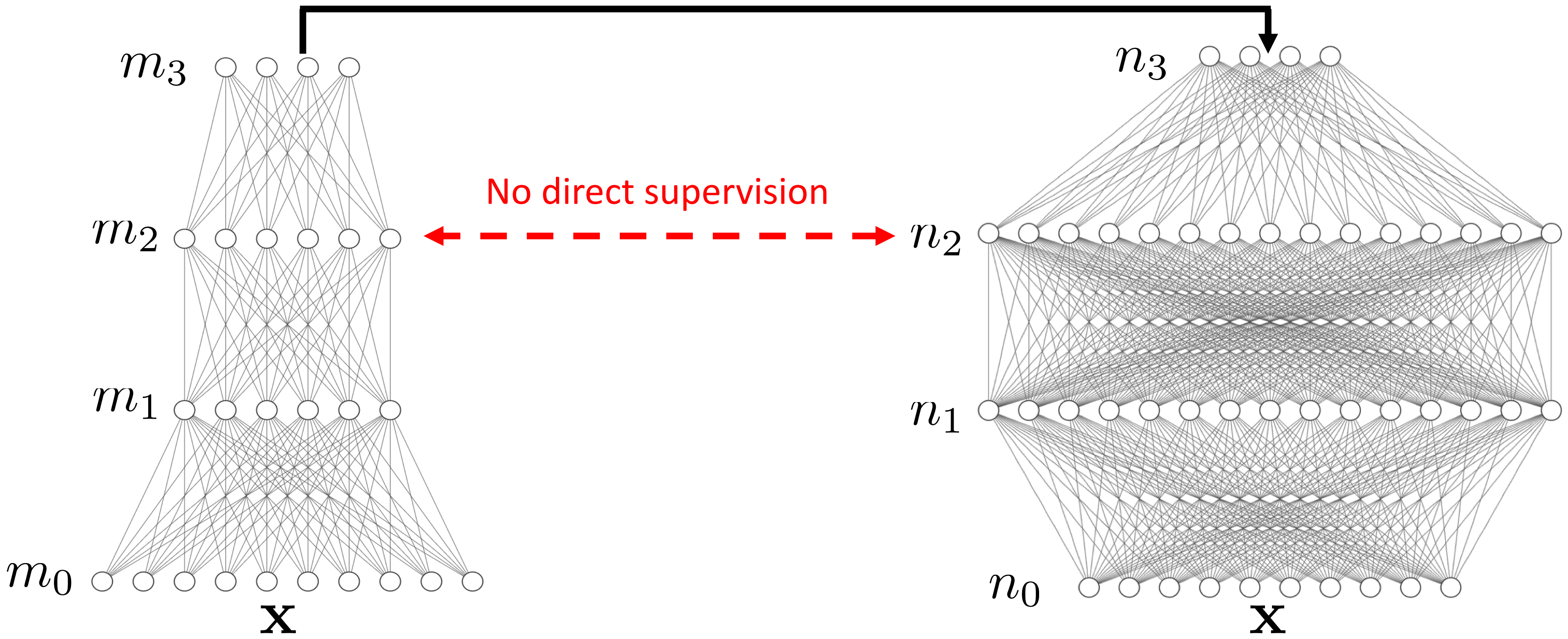
Use Gaussian erf() function as the nonlinearity

Study when the input dimension $d \to +\infty$ (i.e., **thermodynamics limits**)

[On-line learning in soft committee machines, Saad & Solla, Phys. Rev 1995]

Student-Teacher Setting (this paper)

$$\min_{\mathcal{W}} J(\mathcal{W}) = \frac{1}{2}\mathbb{E}_{\mathbf{x}}\left[\|\mathbf{f}_L^*(\mathbf{x}) - \mathbf{f}_L(\mathbf{x})\|^2\right]$$



Teacher Network
(Fixed parameters)

(Over-parameterized) Student Network
(Learnable Parameters)

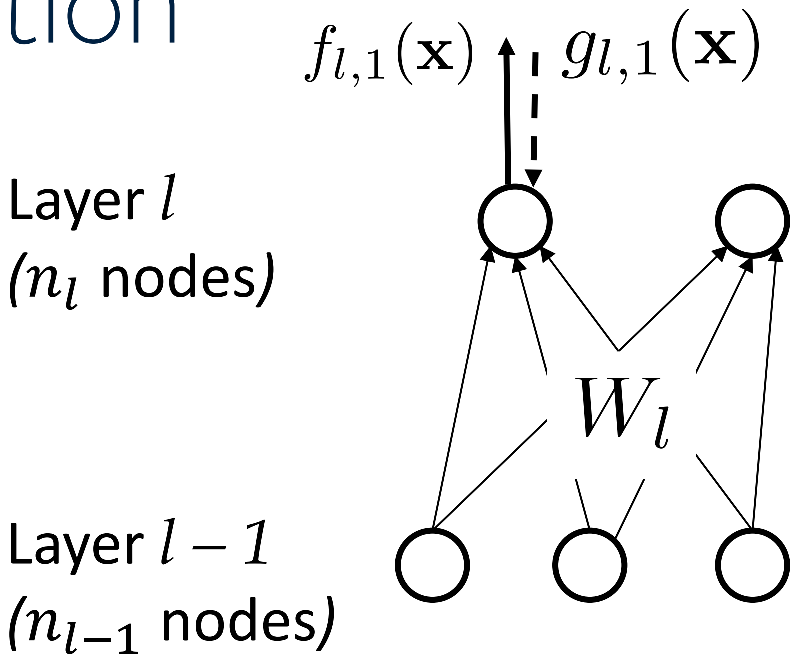
Contributions

Over-parameterization helps in generalization in **two** ways:

1. **Critical point analysis** shows that over-parameterization helps student-teacher alignment.

2. **Training dynamics analysis** shows faster alignment with over-parameterization.

Notation



Activation

$$\mathbf{f}_l(\mathbf{x}) = \begin{bmatrix} f_{l,1}(\mathbf{x}) \\ f_{l,2}(\mathbf{x}) \end{bmatrix}$$

Gradient

$$\mathbf{g}_l(\mathbf{x}) = \begin{bmatrix} g_{l,1}(\mathbf{x}) \\ g_{l,2}(\mathbf{x}) \end{bmatrix}$$

Weight update rule: $\dot{W}_l = \mathbb{E}_{\mathbf{x}}\left[\mathbf{f}_{l-1}(\mathbf{x})\mathbf{g}_l^{\intercal}(\mathbf{x})\right]$

GD: expectation taken over the entire dataset

SGD: expectation taken over a batch

A Trivial Statement

With over-parameterized student network:

Student aligns with the teacher $\longrightarrow$ $$\mathbf{g}_l(\mathbf{x}; \mathcal{W}) = \mathbf{0}, \quad \forall \mathbf{x} \in R_0$$

The Inverse Problem

With over-parameterized student network:

Student aligns
with the teacher

$$\mathbf{g}_l(\mathbf{x}; \mathcal{W}) = \mathbf{0}, \quad \forall \mathbf{x} \in R_0$$

? ? ?

➔ Zero training error leads to good generalization

Lemma1: Recursive Gradient Rule

For layer l, there exists $A_l(x)$ and $B_l(x)$ so that:

$$\mathbf{g}_l(\mathbf{x}) = D_l(\mathbf{x})\left[A_l(\mathbf{x})\mathbf{f}_l^*(\mathbf{x}) - B_l(\mathbf{x})\mathbf{f}_l(\mathbf{x})\right]$$

Student gradient

Student gating

Teacher mixture

Student mixture

$A_l(x)$ and $B_l(x)$ are **piece-wise constant.**

Lemma1: Recursive Gradient Rule

$$D_l(\mathbf{x}) \in \mathbb{R}^{n_l \times n_l}$$
$$A_l(\mathbf{x}) \in \mathbb{R}^{n_l \times m_l}$$
$$B_l(\mathbf{x}) \in \mathbb{R}^{n_l \times n_l}$$

For layer l, there exists $A_l(x)$ and $B_l(x)$ so that:

n_l: number of student nodes at layer l

m_l: number of teacher nodes at layer l

$$\mathbf{g}_l(\mathbf{x}) = D_l(\mathbf{x})\left[A_l(\mathbf{x})\mathbf{f}_l^*(\mathbf{x}) - B_l(\mathbf{x})\mathbf{f}_l(\mathbf{x})\right]$$

Student gradient

Student gating

Teacher mixture

Student mixture

$$\mathbf{f}_l^*(\mathbf{x}) \in \mathbb{R}^{m_l}$$
$$\mathbf{f}_l(\mathbf{x}) \in \mathbb{R}^{n_l}$$
$$\mathbf{g}_l(\mathbf{x}) \in \mathbb{R}^{n_l}$$

$A_l(x)$ and $B_l(x)$ are **piece-wise constant.**

Recursive Formula for $A_l(x)$ and $B_l(x)$

$V_l(\mathbf{x}) \in \mathbb{R}^{C\times n_l}$
$V_l^*(\mathbf{x}) \in \mathbb{R}^{C\times m_l}$
C: output dimension

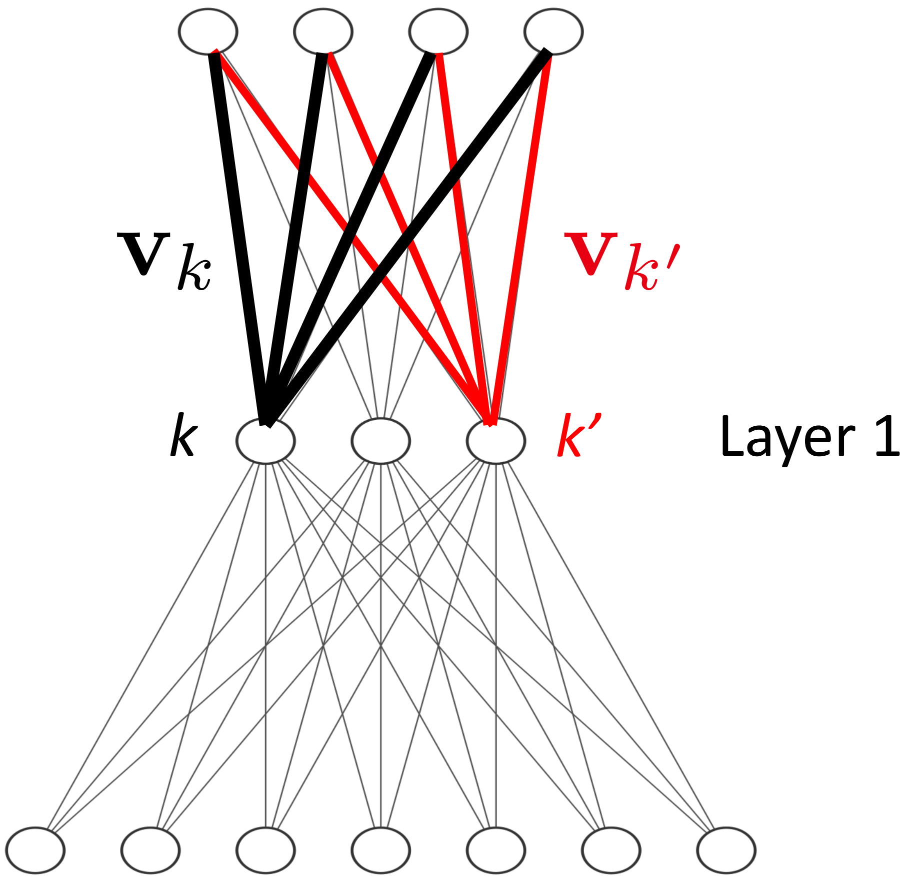
$$A_l(\mathbf{x}) = V_l^\intercal(\mathbf{x}) V_l^*(\mathbf{x})$$
$$B_l(\mathbf{x}) = V_l^\intercal(\mathbf{x}) V_l(\mathbf{x})$$

Recursive Formula for *V*:

$$V_{l-1}^*(\mathbf{x}) = V_l^*(\mathbf{x}) D_l^*(\mathbf{x}) W_l^{*\intercal}$$
$$V_{l-1}(\mathbf{x}) = V_l(\mathbf{x}) D_l(\mathbf{x}) W_l^\intercal$$

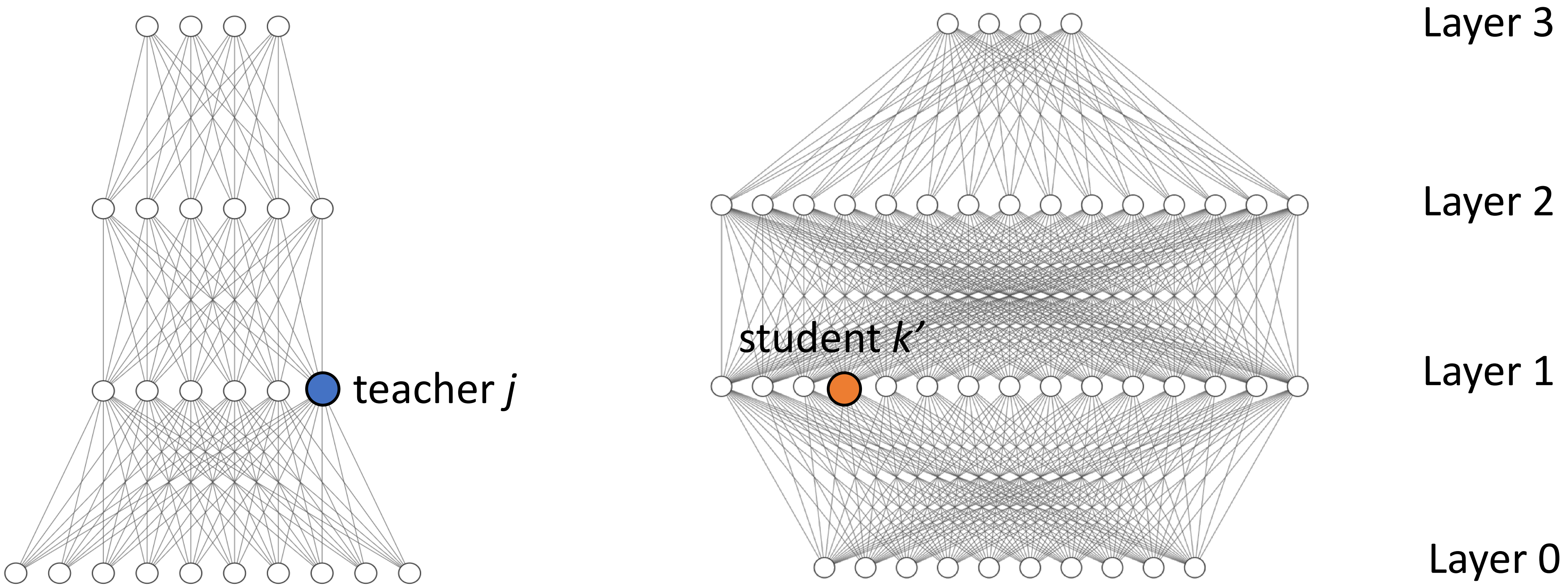
Base case:

$$V_L(\mathbf{x}) = V_L^*(\mathbf{x}) = I_{C\times C}$$

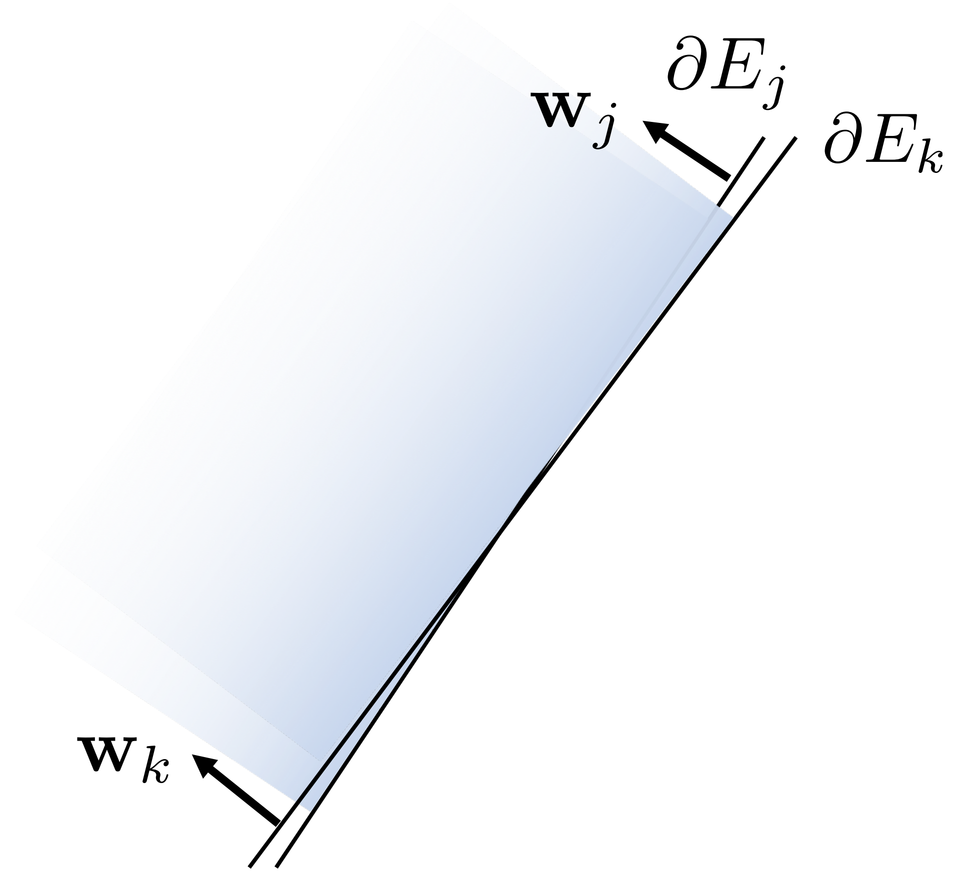


$$V_1 = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{n_1}]$$

Main results: Alignment could happen!



Definition of Alignment



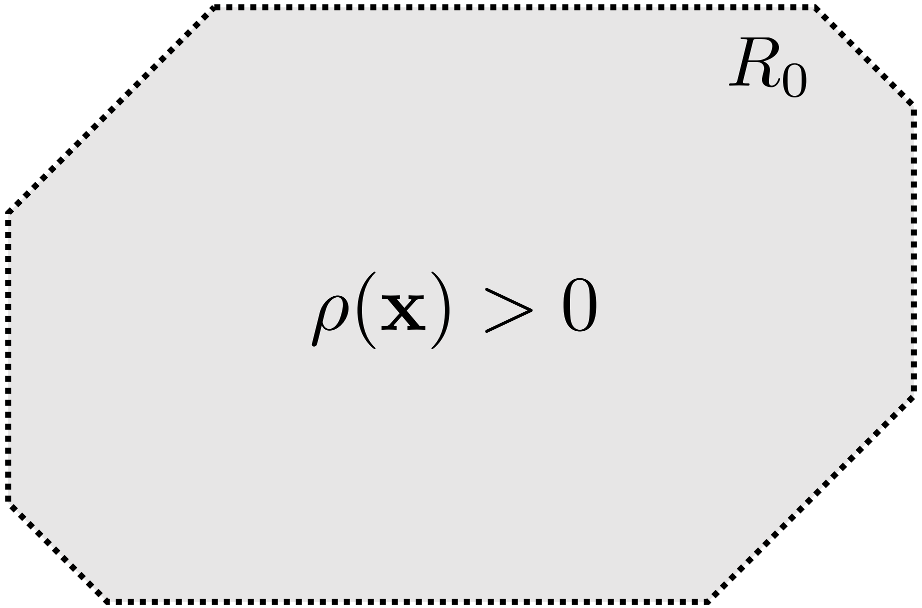
An example of “rough” alignment

E_j Activated Region of node *j*

∂E_j Boundary of node *j*

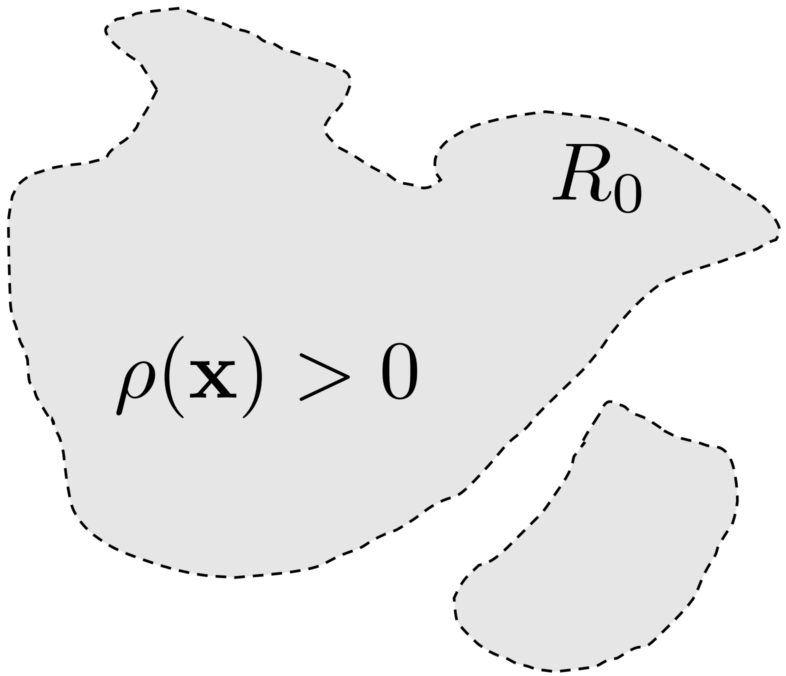
∂E_k Boundary of node *k*

Assumption of the dataset



Infinite dataset!

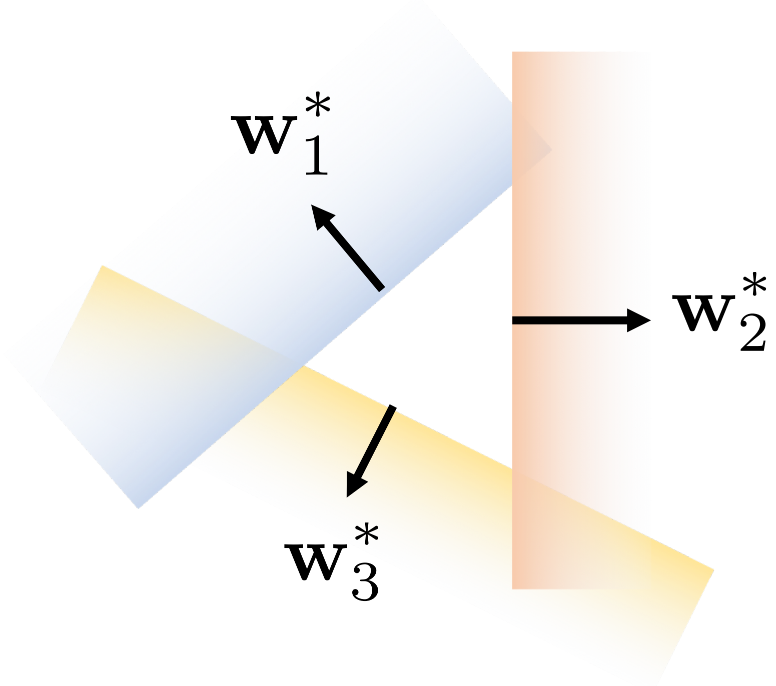
Assumption of the dataset



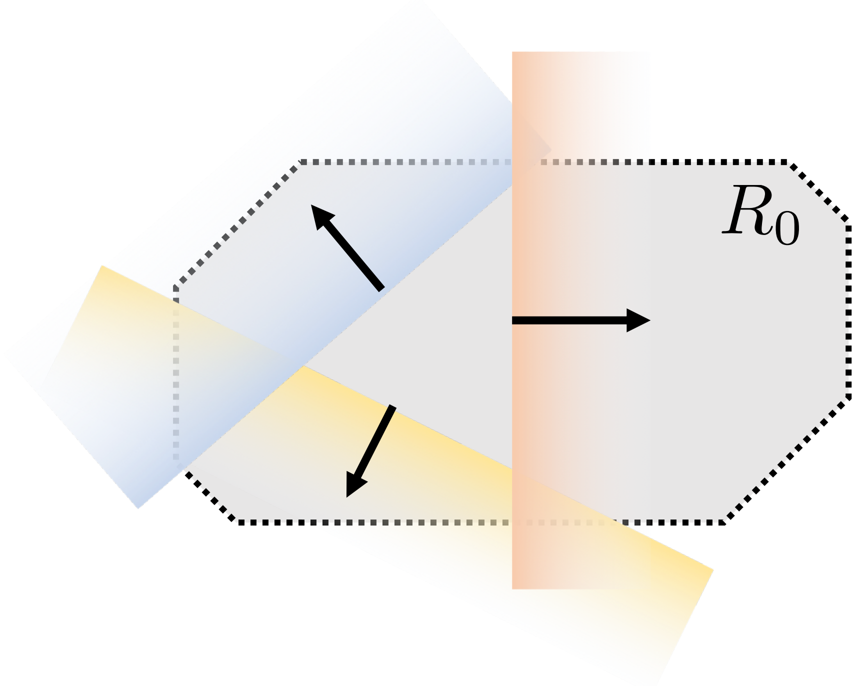
Infinite dataset!

Assumptions on Teacher Network

- Cannot reconstruct arbitrary teachers
  - e.g., all ReLU nodes are dead

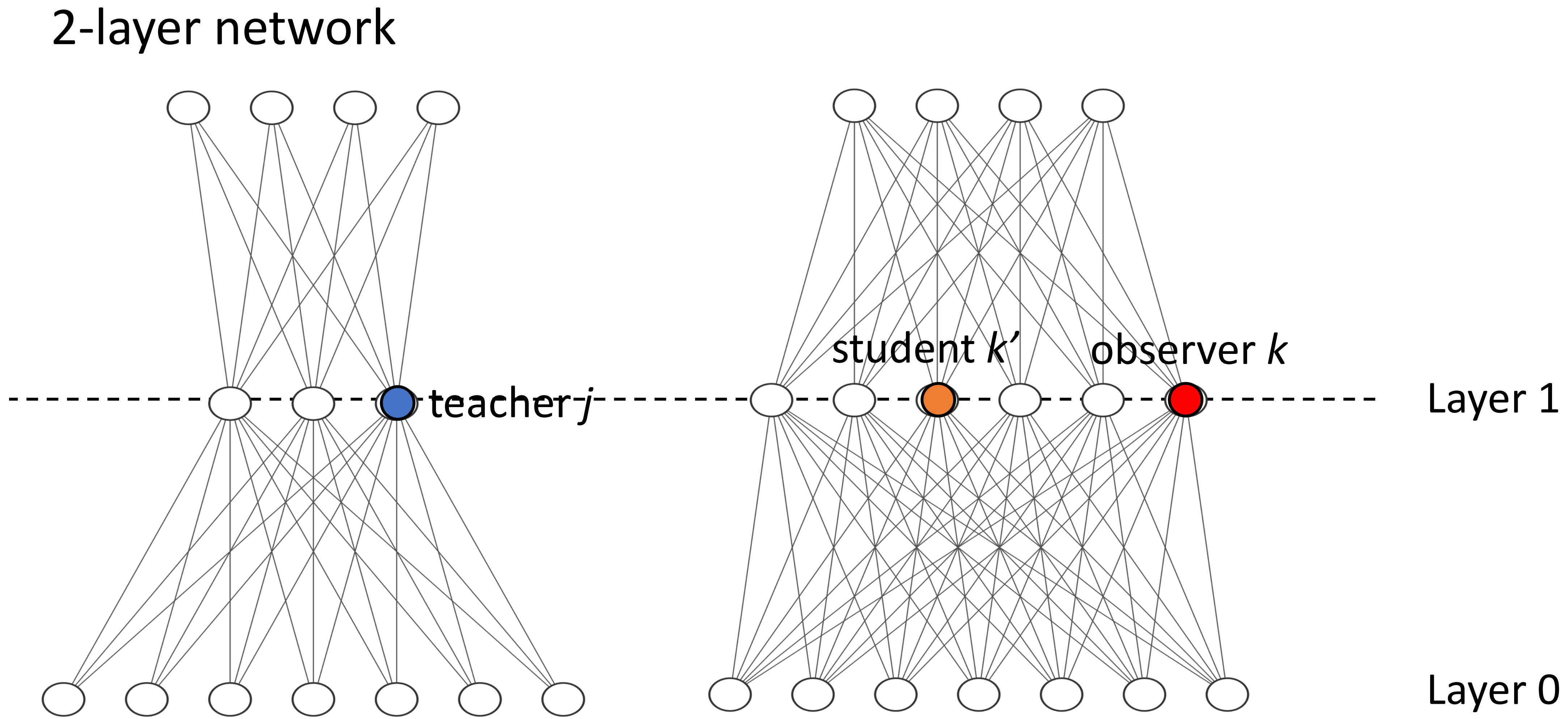


Distinct teacher nodes

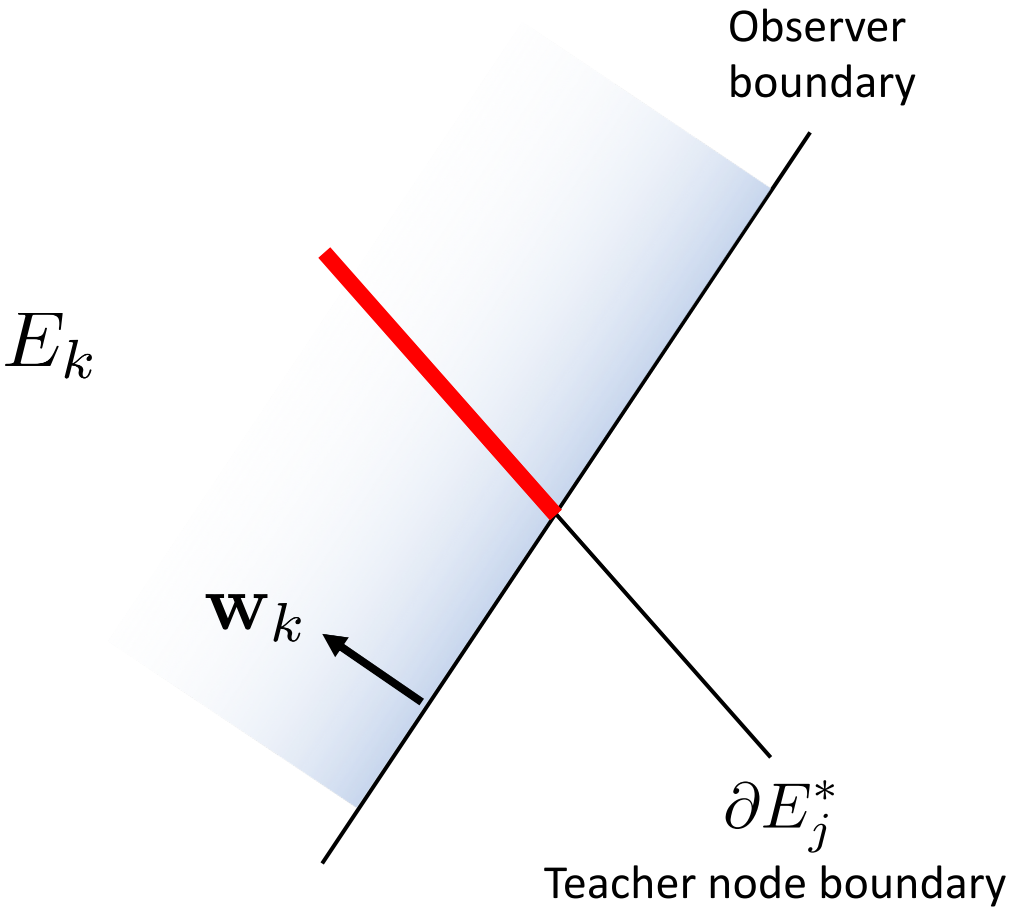


Teacher's boundary are visible in the dataset

Main results: Alignment could happen!



Definition of "Observation"



$$\partial E_j^* \cap E_k \neq \emptyset$$

Teacher *j* is **observed** by a student *k*

Main results: Alignment could happen!

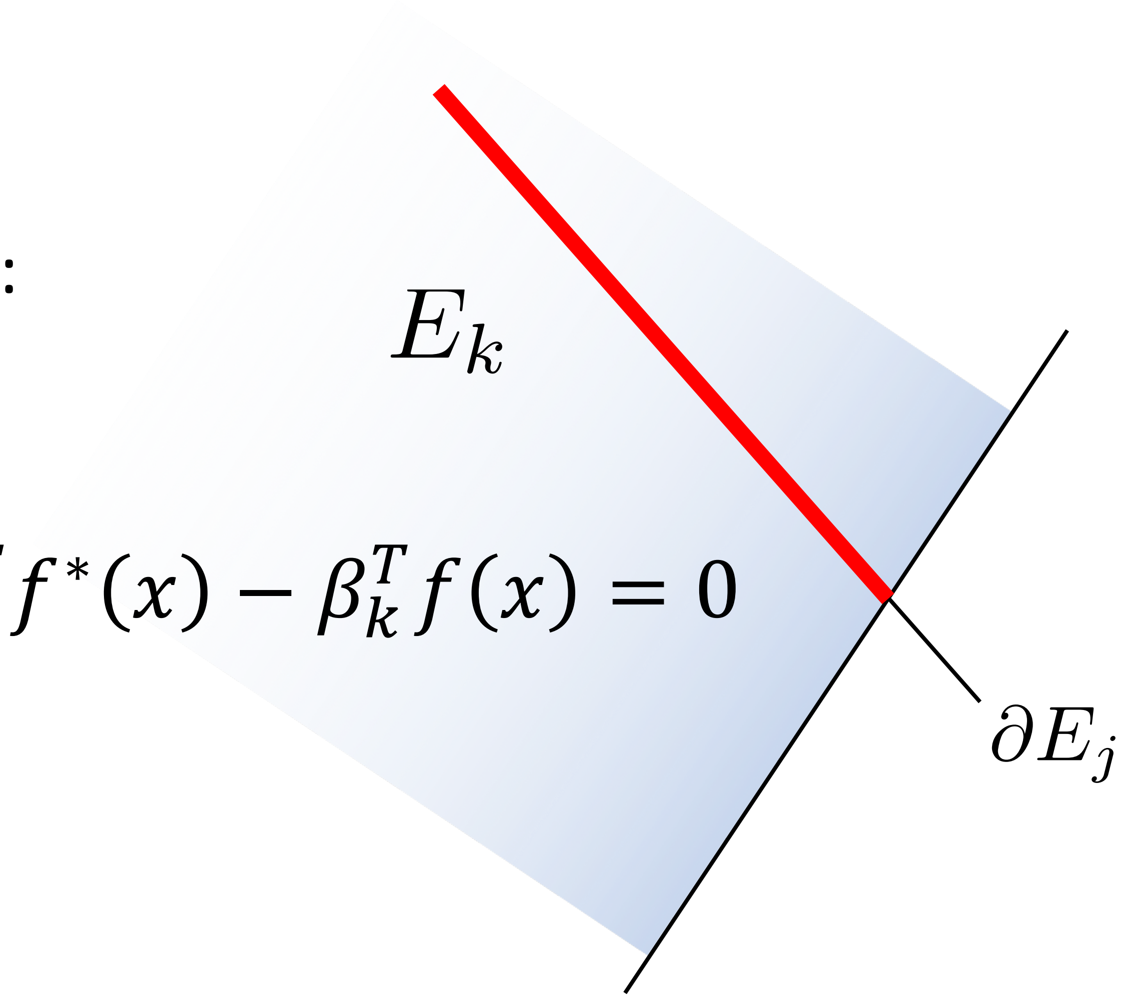
Teacher node *j* is **observed** by a student node *k*

→

Teacher *j* is **aligned with** at least one student *k'*

Why?

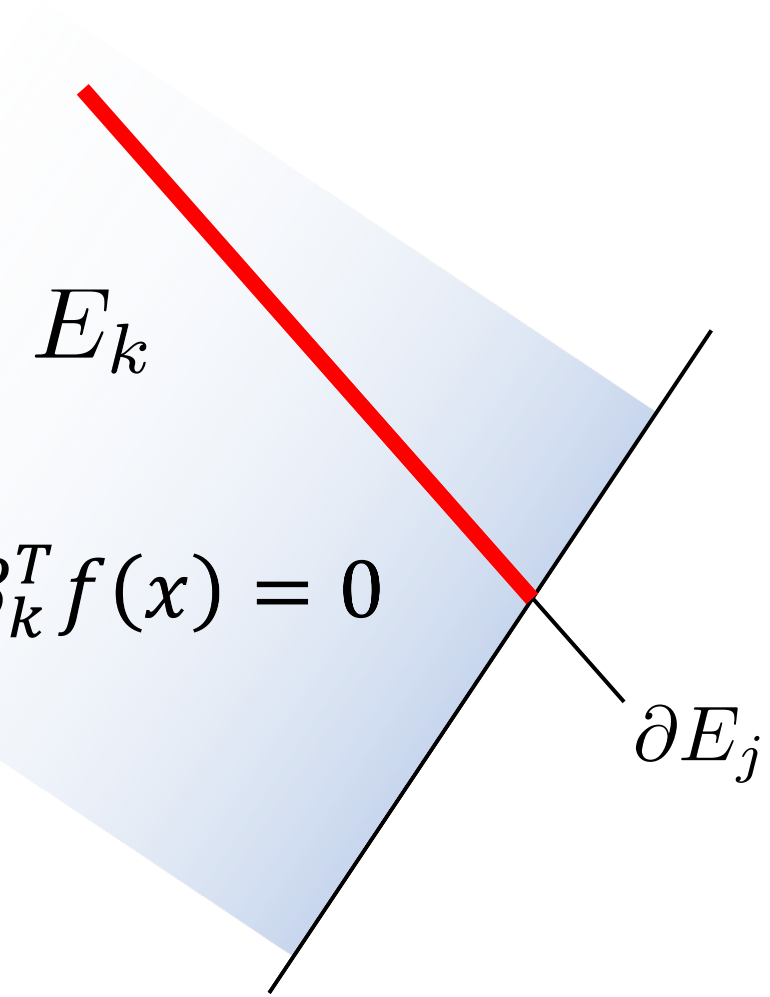
The gradient of observer *k* is 0:



From Lemma 1, $g_k(x) = \alpha_k^T f^*(x) - \beta_k^T f(x) = 0$
If $x \in E_k$

Why?

The gradient of observer *k* is 0:



From Lemma 1, $g_k(x) = \alpha_k^T f^*(x) - \beta_k^T f(x) = 0$

If $x \in E_k$

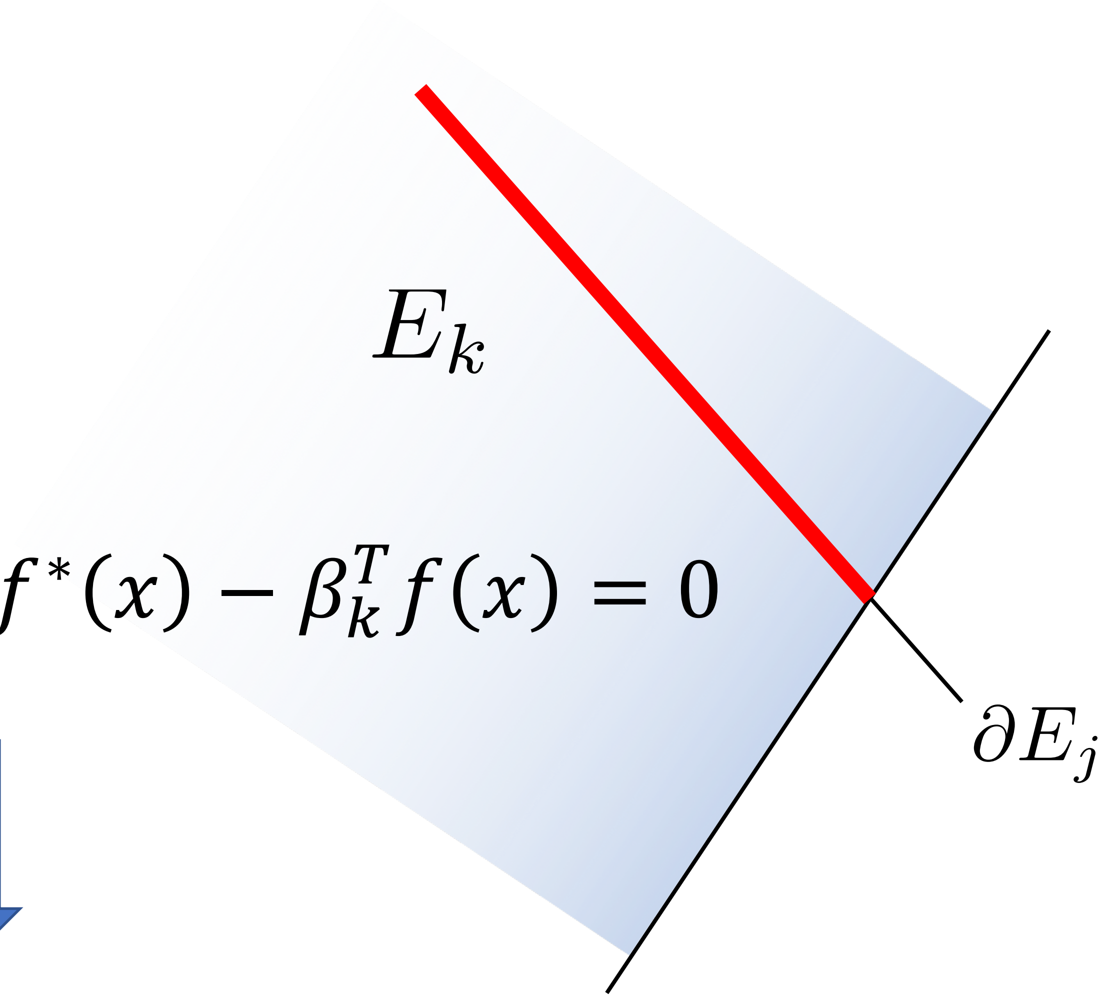
ReLUs are linear independent!

⬇

Coefficients for teacher *j* direction must be 0

Why?

The gradient of observer *k* is 0:



From Lemma 1, $g_k(x) = \alpha_k^T f^*(x) - \beta_k^T f(x) = 0$

If $x \in E_k$

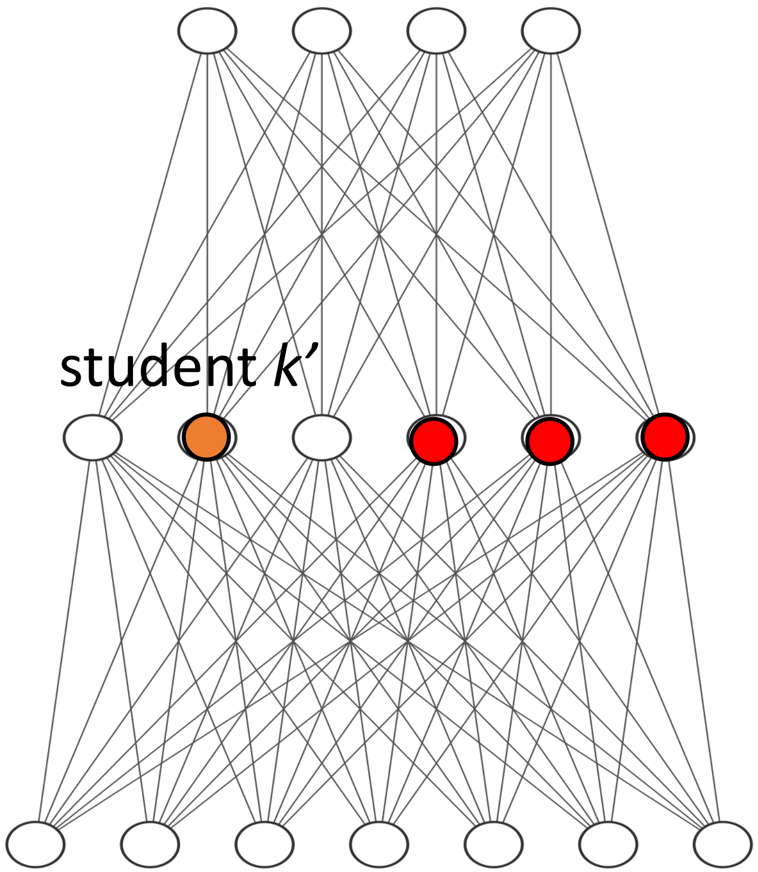
ReLUs are linear independent!

Coefficients for teacher *j* direction must be 0

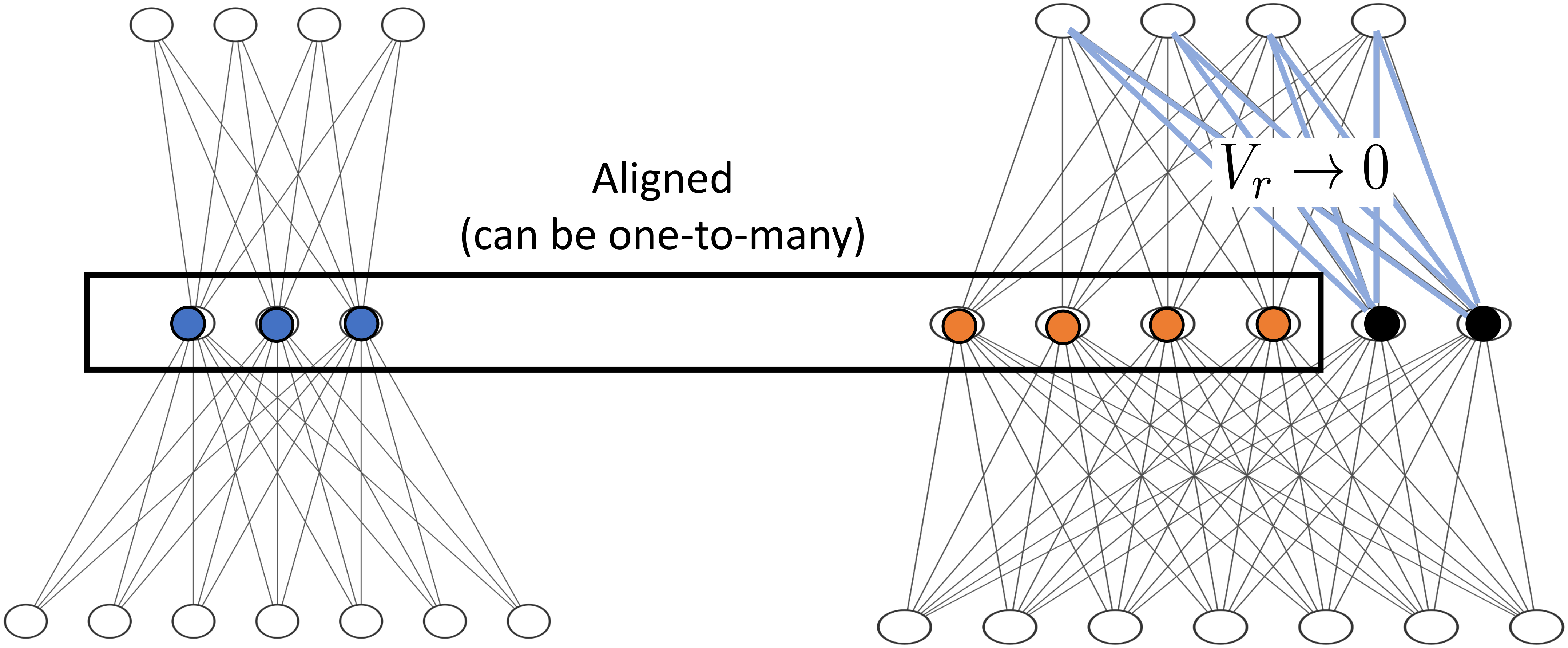
Teacher *j* is aligned with at least one student *k'* (sum of coefficients = 0)

Why Over-parameterization helps?

More observers!

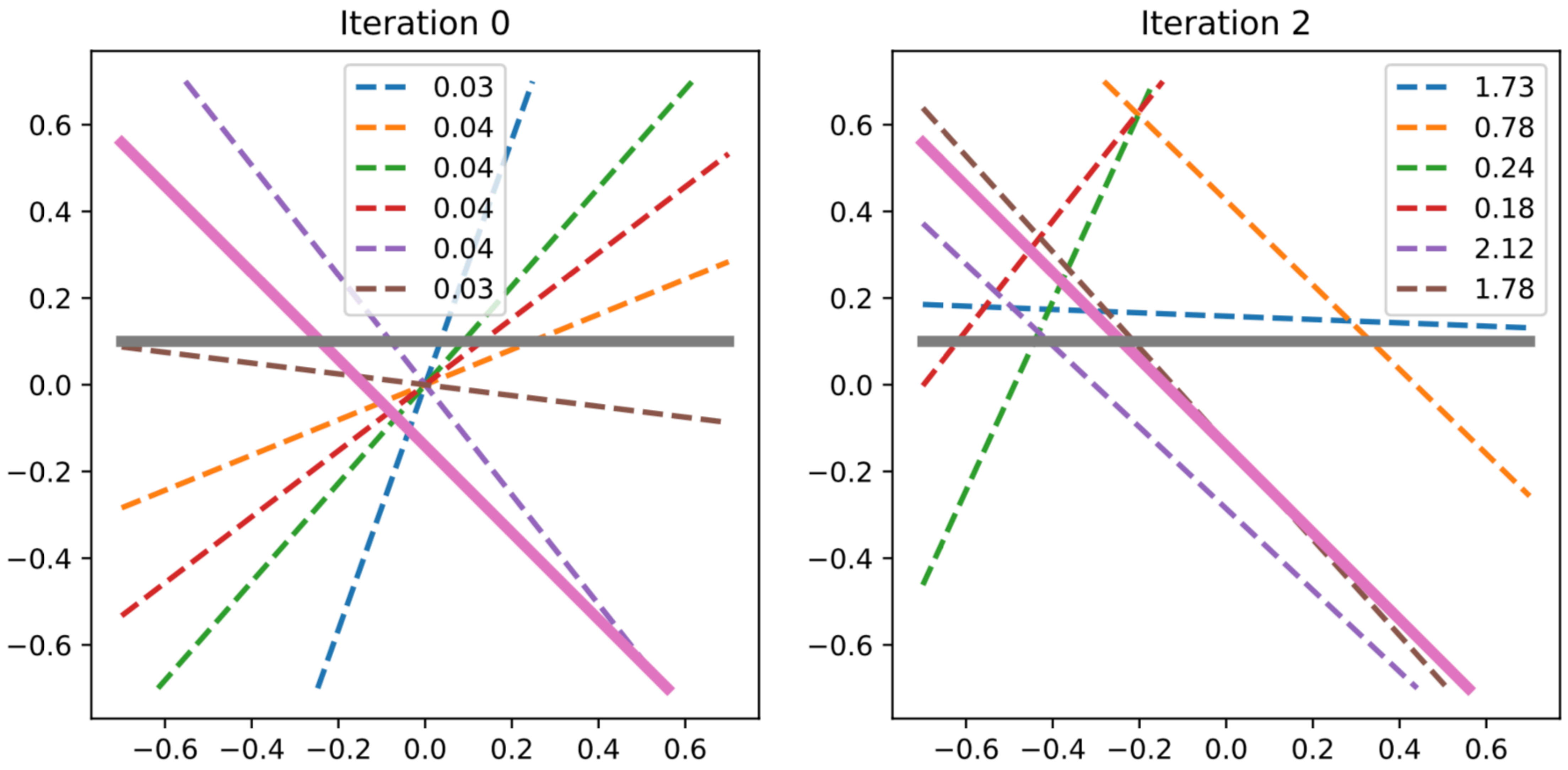


What happens to unaligned students?

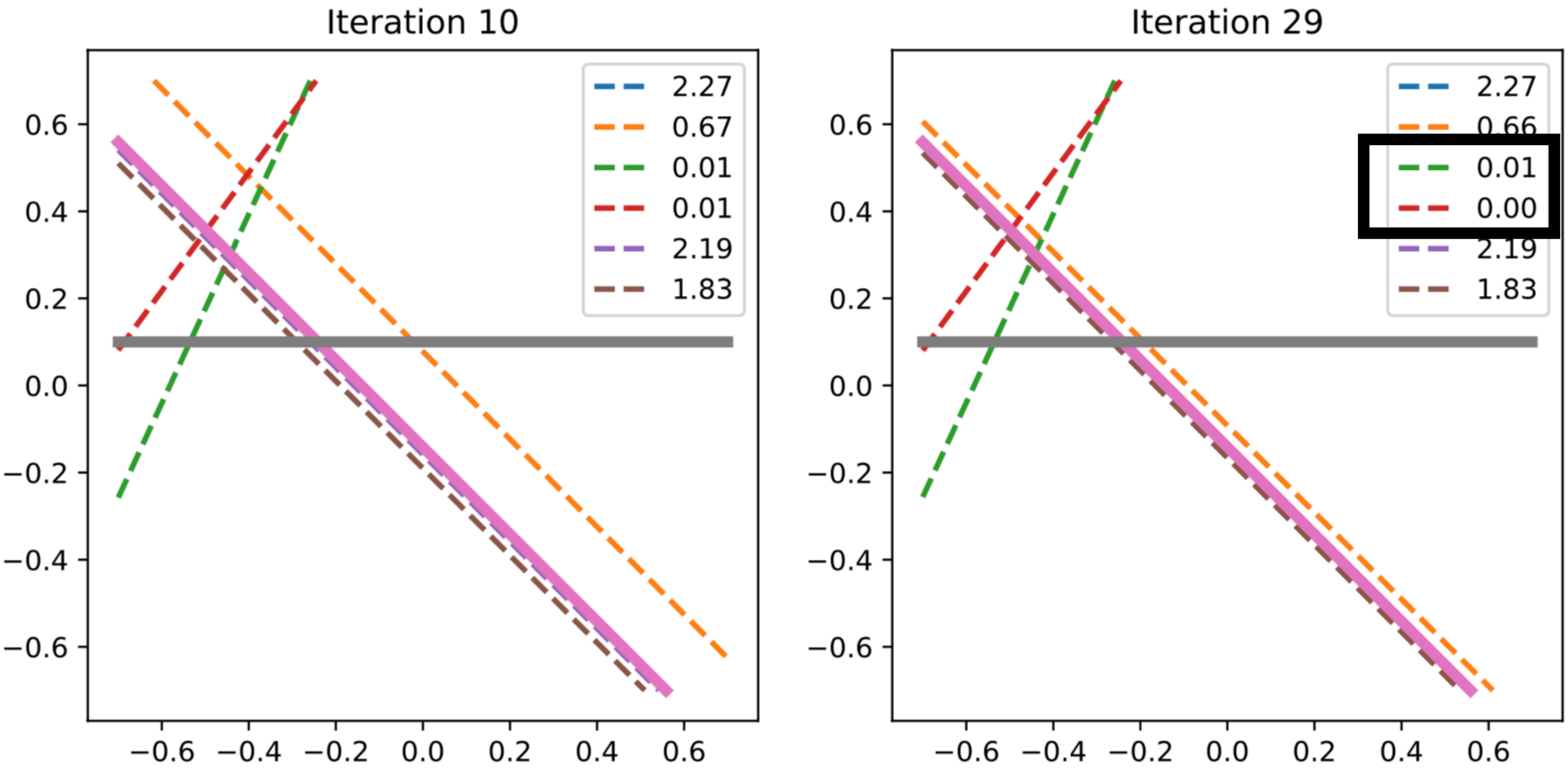


Simple 2D experiments

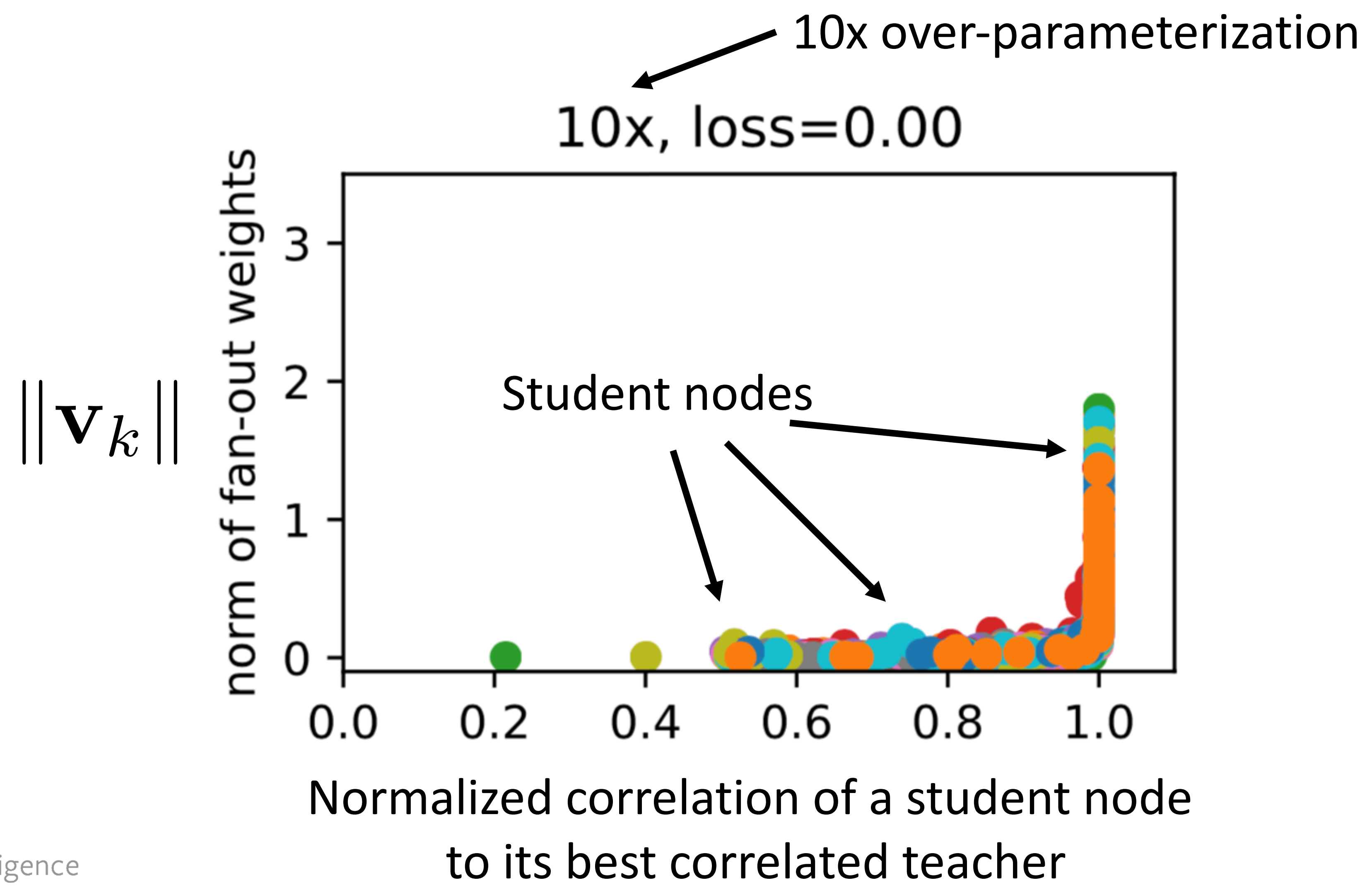
Student Boundary
Teacher Boundary



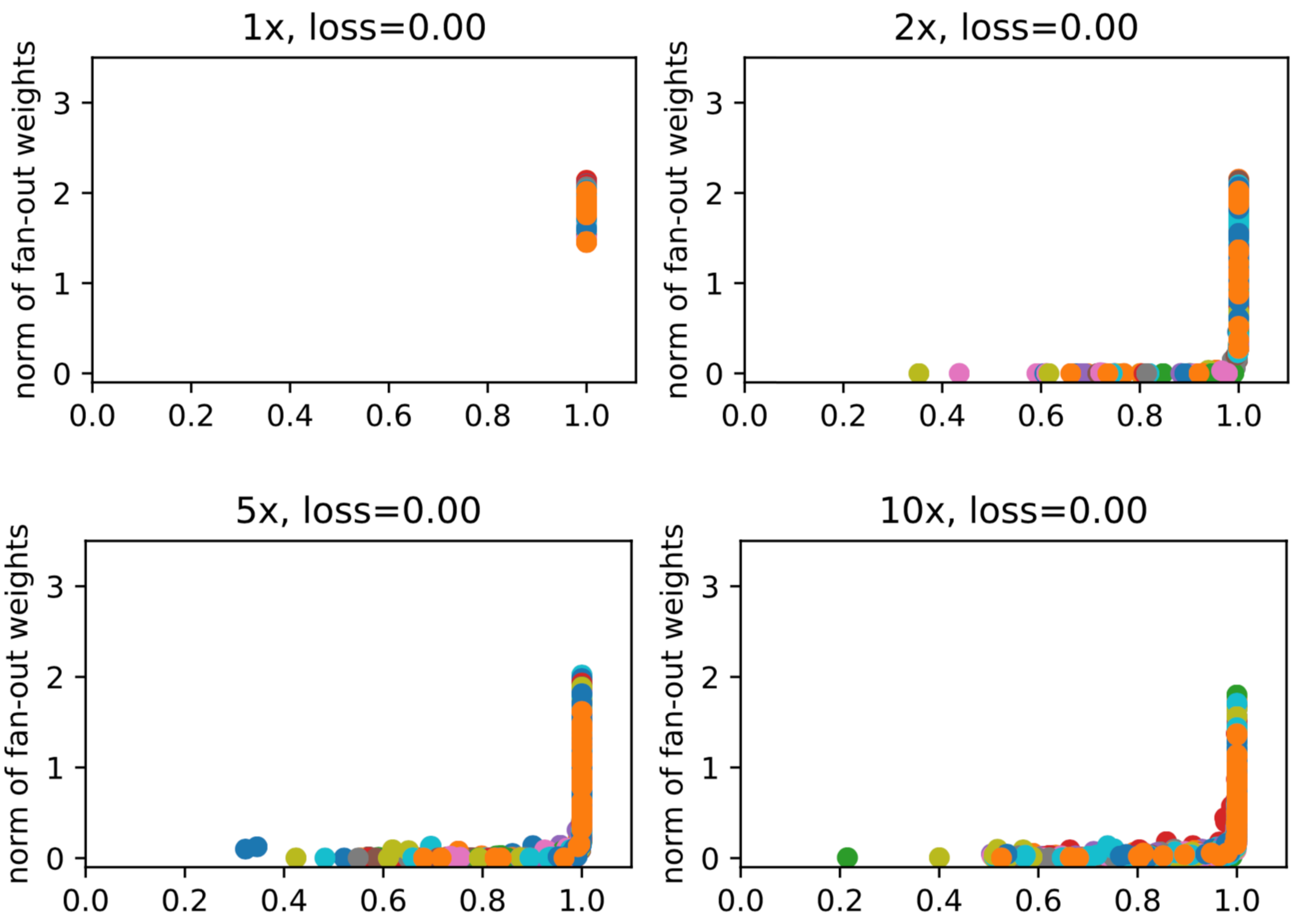
Simple 2D experiments



L-shape curve at convergence



L-shape curve at convergence

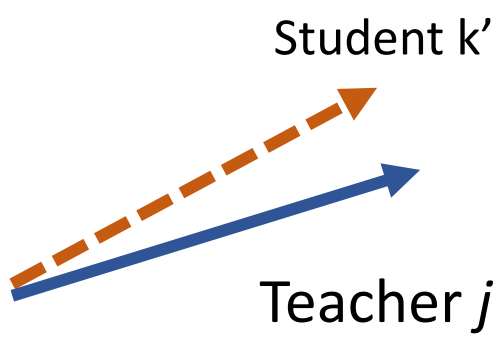


Noisy Case $\|\mathbf{g}_1(\mathbf{x};\mathcal{W})\|_\infty \leq \epsilon$

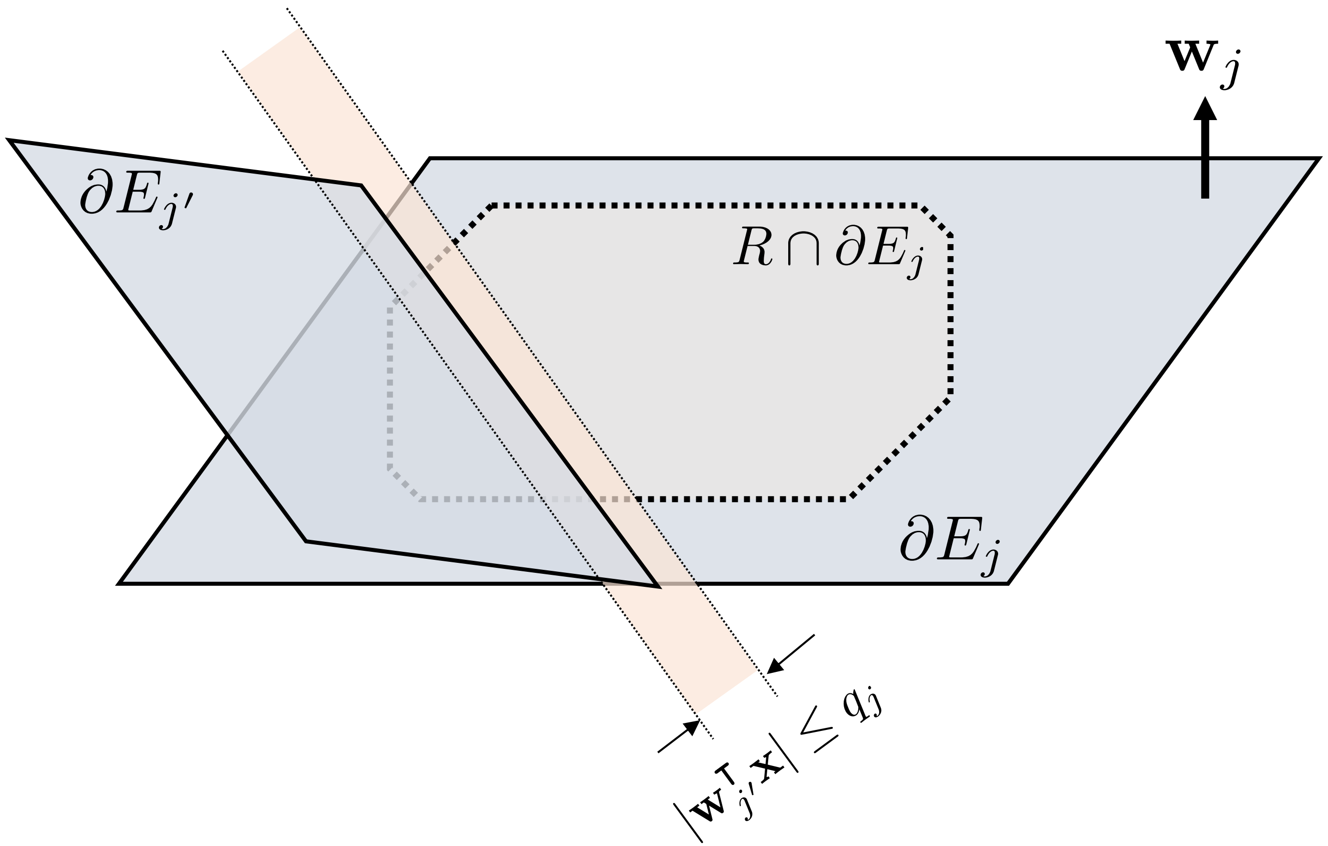
For teacher *j*, there exists student *k'*:

weights $$\sin\theta_{jk'} = \mathcal{O}\left(\frac{\epsilon^{1-\delta}}{|\alpha_{kj}|}\right)$$

bias $$|b_j^* - b_{k'}| = \mathcal{O}\left(\frac{\epsilon^{1-2\delta}}{|\alpha_{kj}|}\right)$$

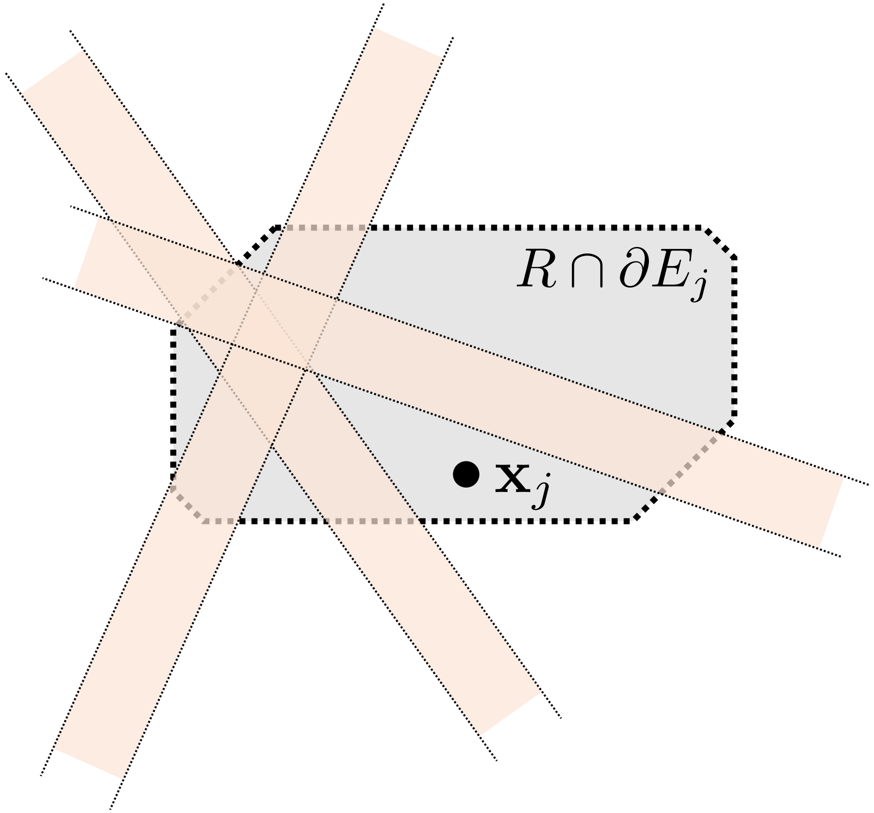


How to Prove?



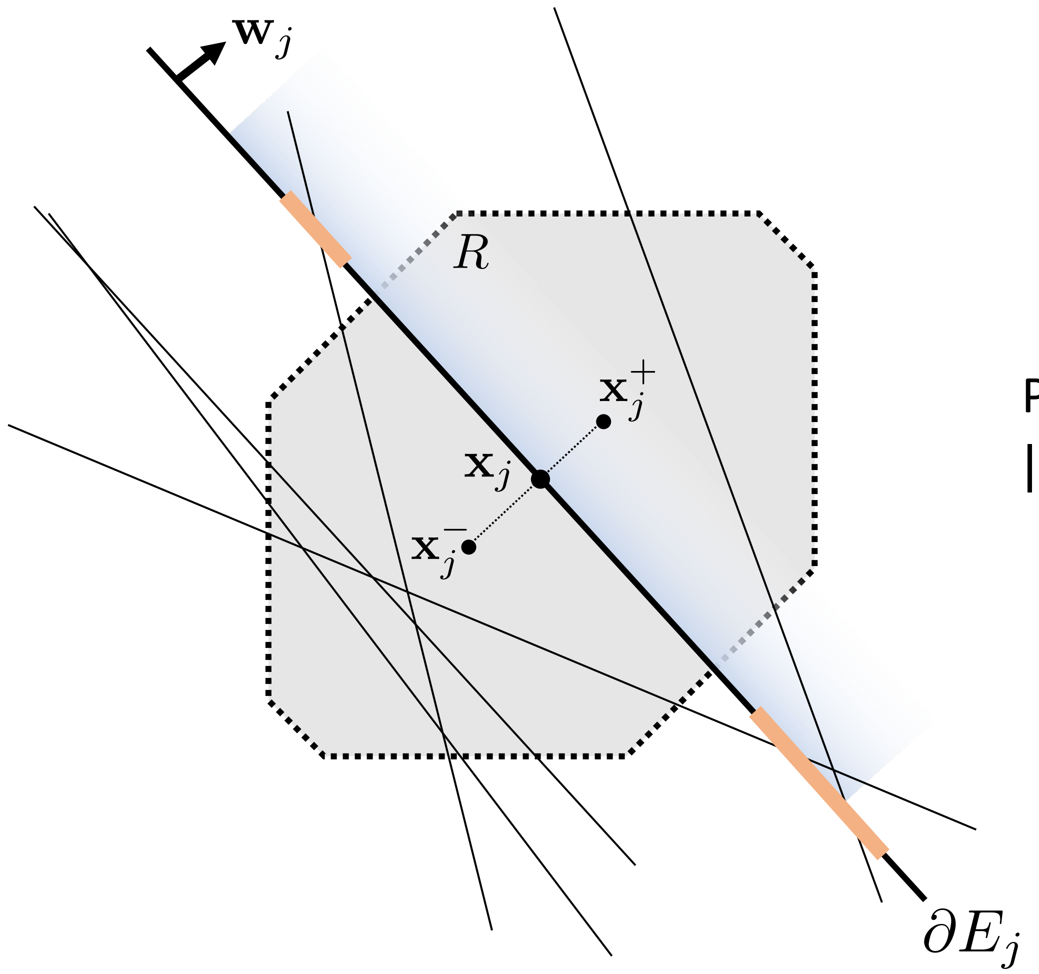
Misalignment leads to small overlap

How to Prove?



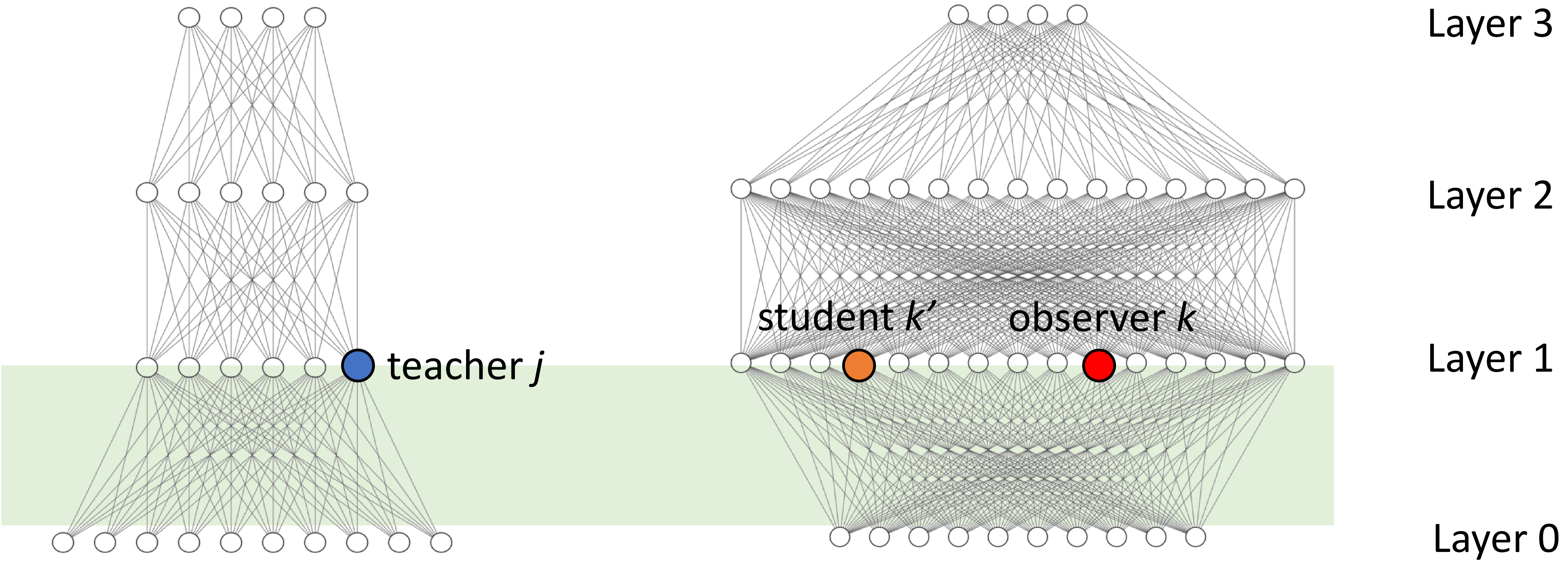
Small overlap → There exists a datapoint that is far away from all boundaries.

How to Prove?



Pick three points x_j, x_j^+, x_j^- and there will be one with $|g_j(x)| > \epsilon$, which is a contradiction.

Multi-Layer case: Alignment could happen!



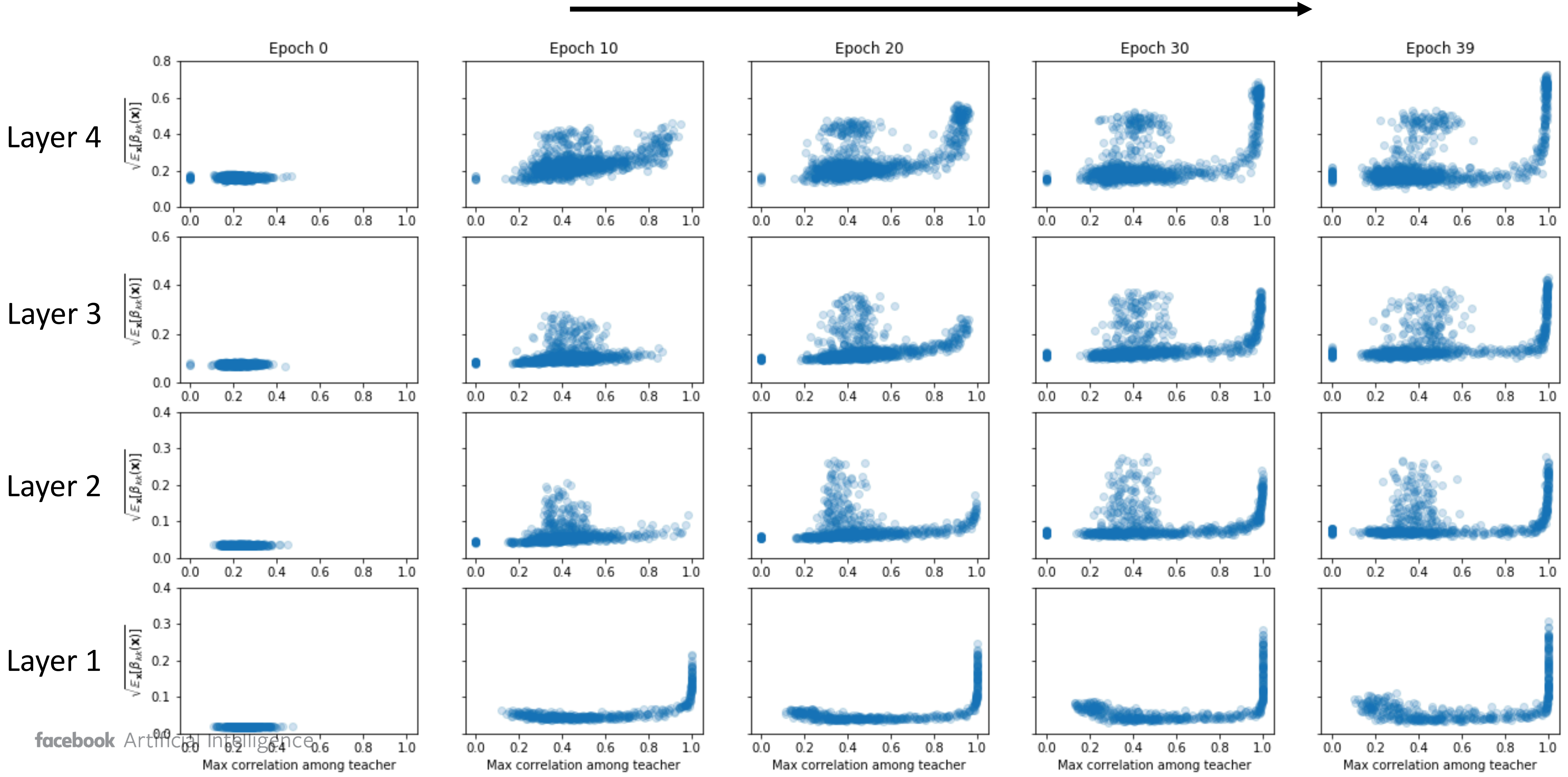
$$\boldsymbol{\alpha}_k^\intercal(\mathbf{x})\mathbf{f}^*(\mathbf{x}) - \boldsymbol{\beta}_k^\intercal(\mathbf{x})\mathbf{f}(\mathbf{x}) = \mathbf{0}$$

Piece wise constant, apply the same logic **per region!**

For 2-layer:

$$\sqrt{\mathbb{E}_{\mathbf{x}}\left[\beta_{kk}(\mathbf{x})\right]} = \|\mathbf{v}_k\|$$

Training Progresses

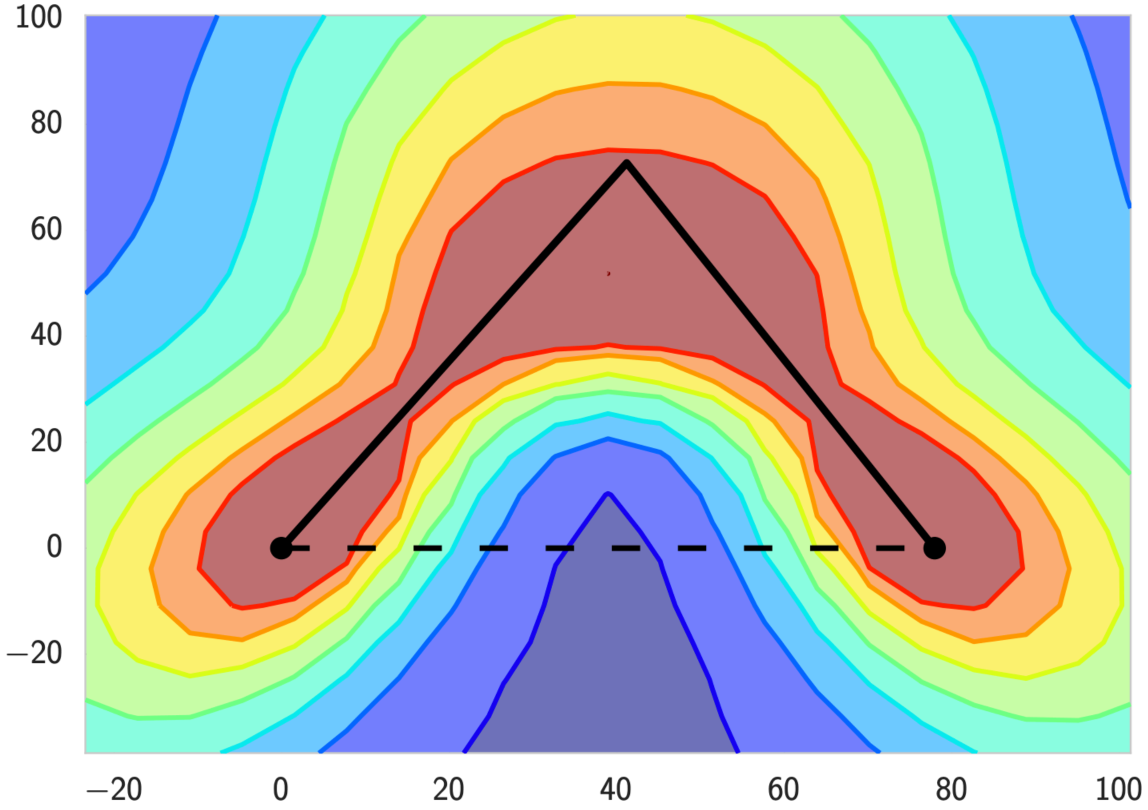


Different initialization, Similar Solutions

VGG-19 on CIFAR-100

[All Neural Networks are Created Equal, Hacohen et al, 2019]

Solutions can be connected by line segments

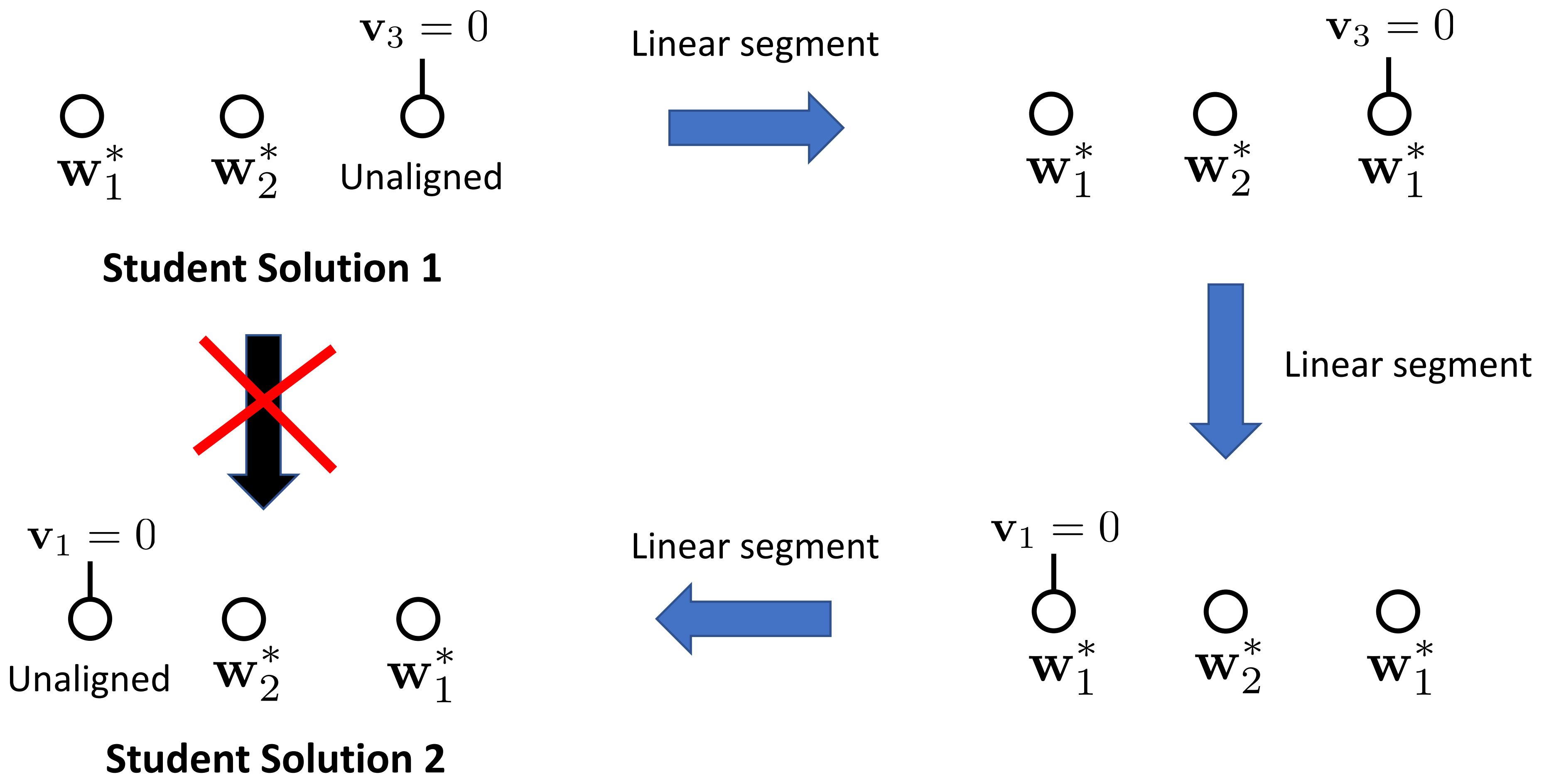


[Loss Surfaces, Mode Connectivity, and Fast Ensembling of DNNs, Garipov et al. NeurIPS 2018]

[Essentially No Barriers in Neural Network Energy Landscape, Draxler et al, 2018]

[Explaining Landscape Connectivity of Low-cost Solutions for Multilayer Nets, Kuditipudi et al, 2019]

Our Explanation



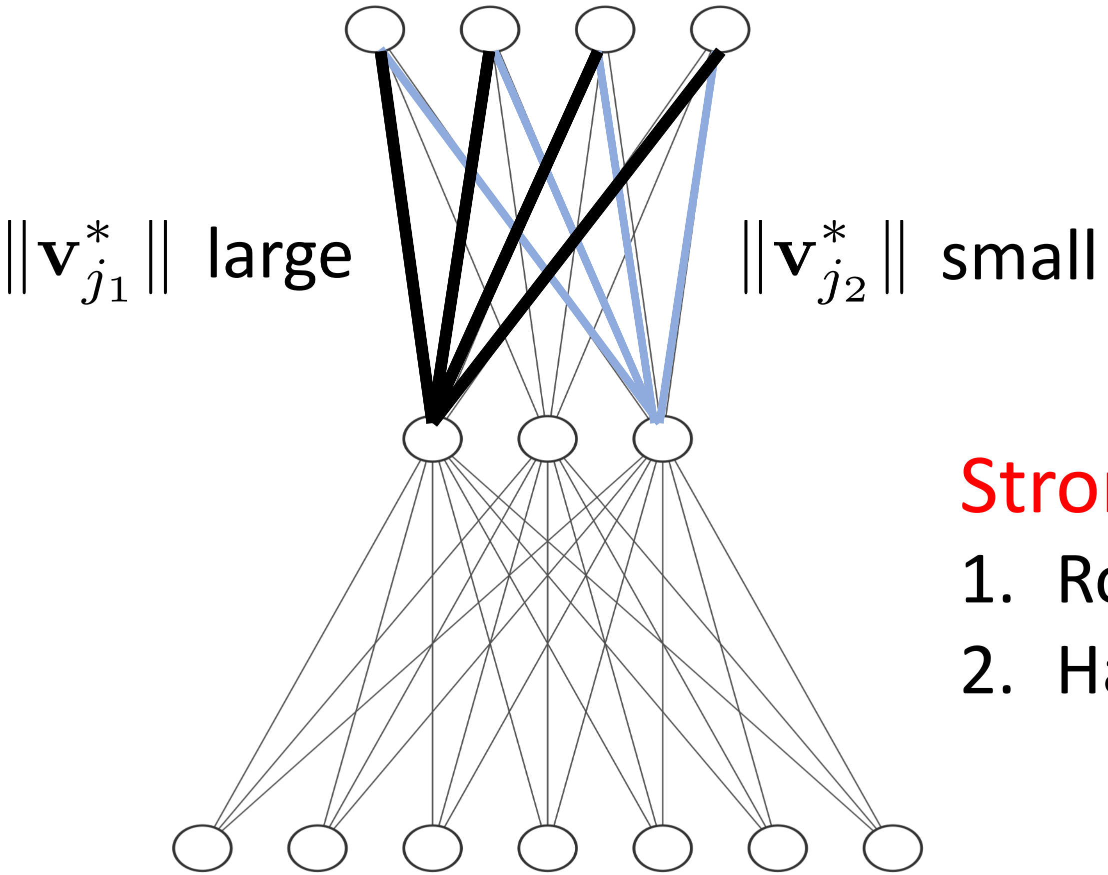
Training Dynamics

Critical Points have nice properties!

Can we achieve that via training with SGD?

Not Easy

Strong/weak teacher nodes

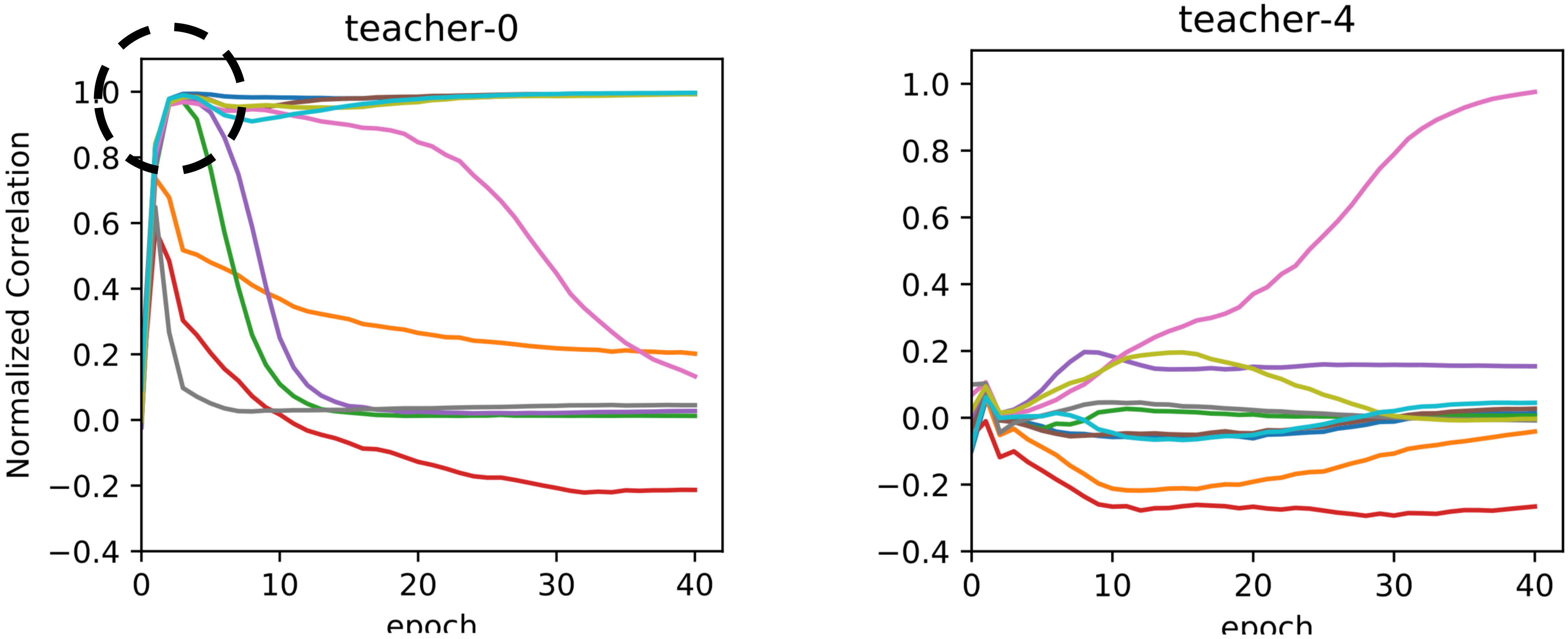


Strong teacher nodes are learned faster

1. Robust to Noise! 😃
2. Hard to learn weak teacher nodes 😢

Training Dynamics

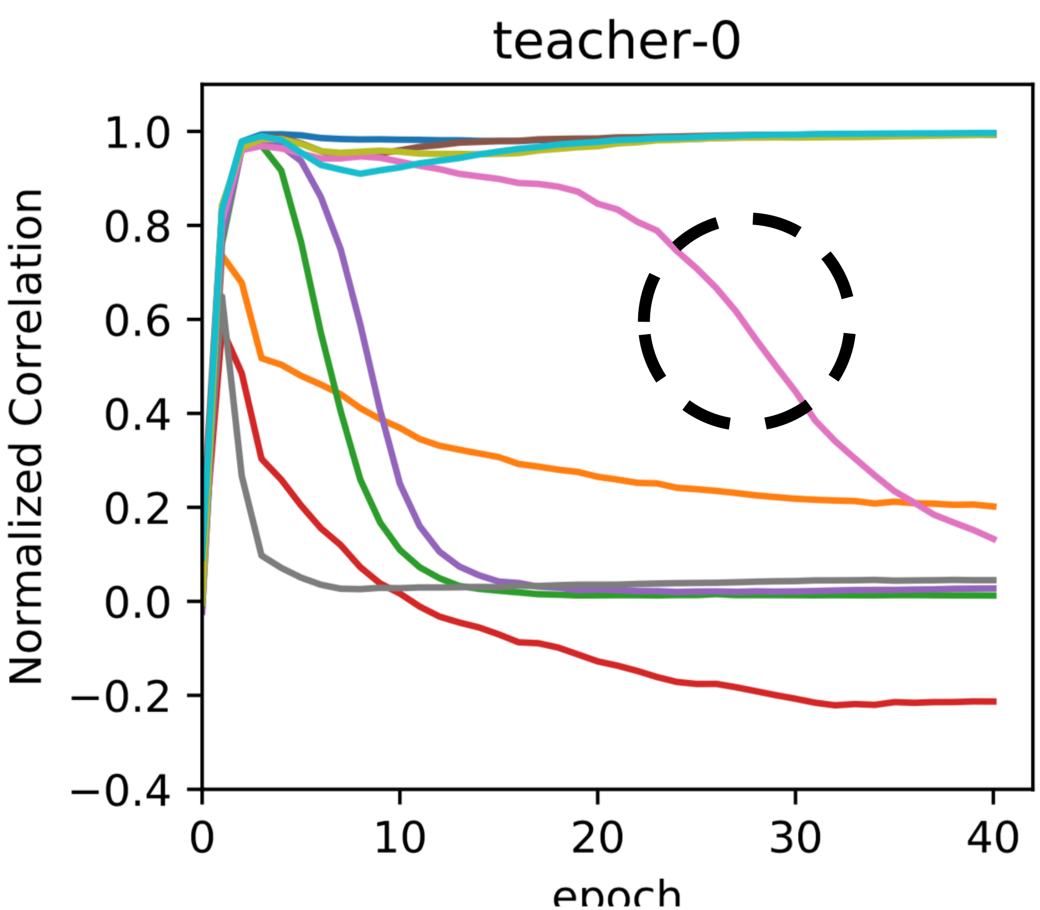
Teacher *j*: $\|\mathbf{v}_j^*\| \propto 1/j^p$

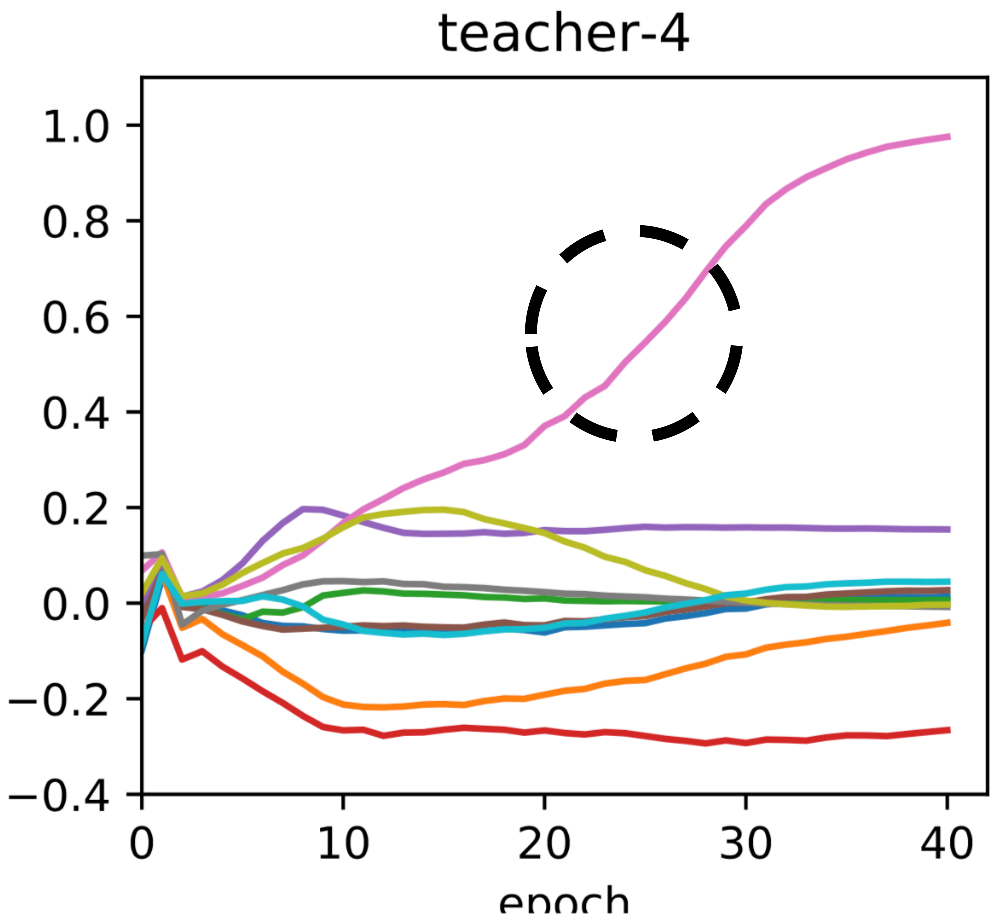


Strong teacher node attracts many students!

Training Dynamics

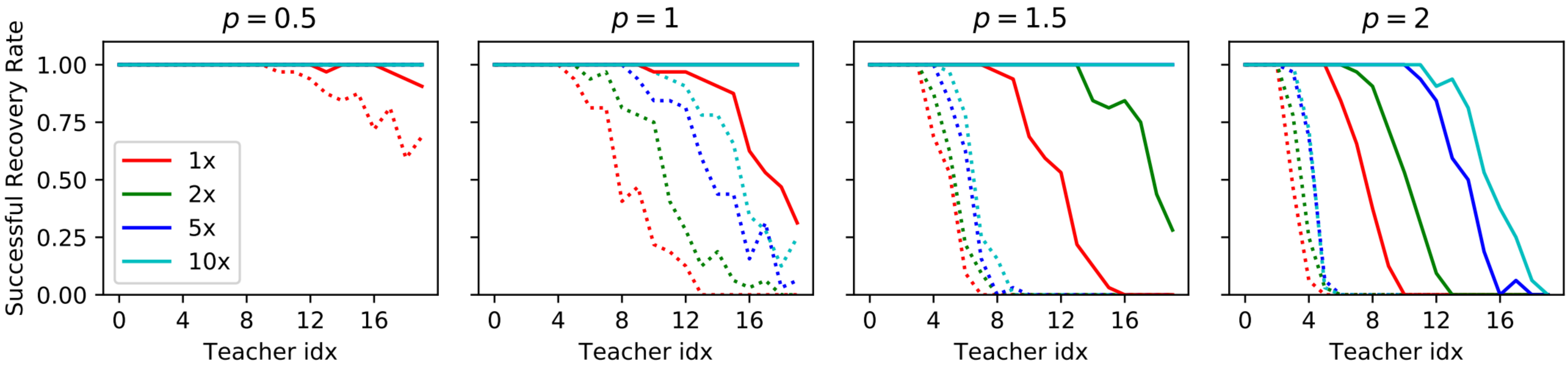
Teacher *j*: $\|\mathbf{v}_j^*\| \propto 1/j^p$





Losing student node shifts focus.

Successful Rate of Teacher Node Reconstruction



- - - - - - 5 epochs

——— 100 epochs

Teacher *j*: $\|\mathbf{v}_j^*\| \propto 1/j^p$

Analysis of (approx.) Training Dynamics

For each node *k*, we have:

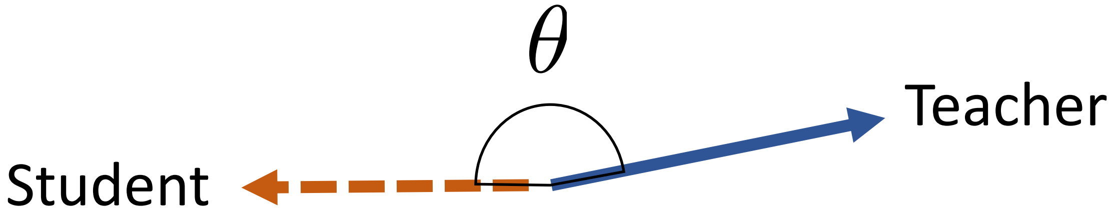
$$\dot{\mathbf{w}}_k = \|\mathbf{w}_k\|\mathbf{r}_k$$

Where:

$$\mathbf{r}_k = \sum_j \alpha_{jk}\psi(\theta_{jk})\mathbf{w}_j^* - \sum_{k'} \beta_{k'k}\psi(\theta_{k'k})\mathbf{w}_{k'} - \nu\mathbf{w}_k$$

$$\psi(\pi) = 0$$

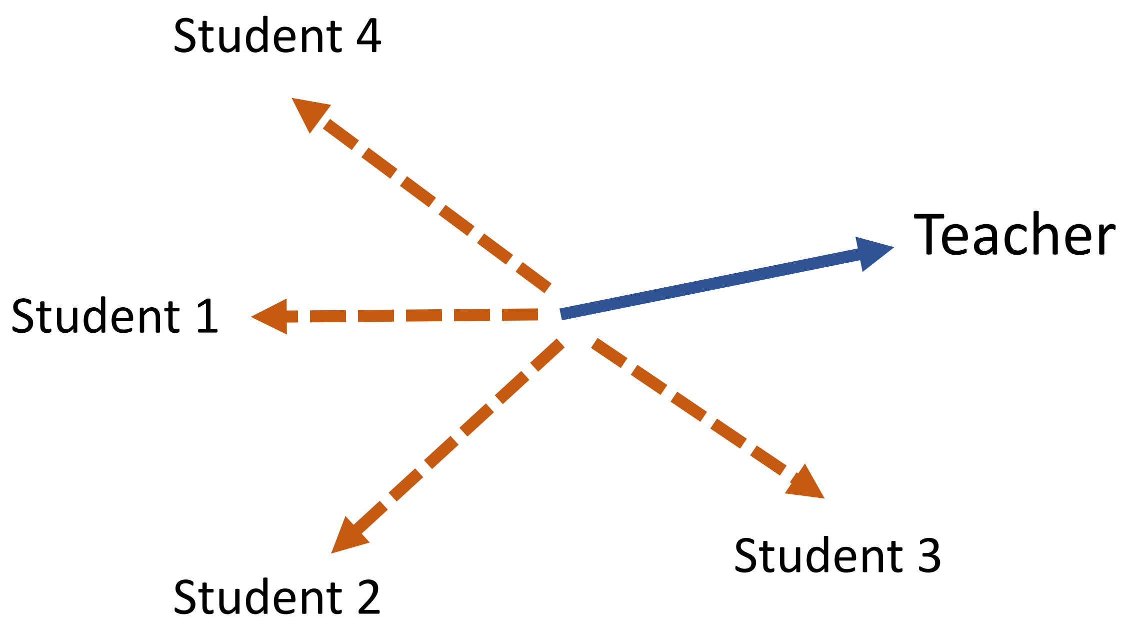
Worst case scenario



$$\boxed{\theta_0 \to \pi, \quad t \to +\infty}$$

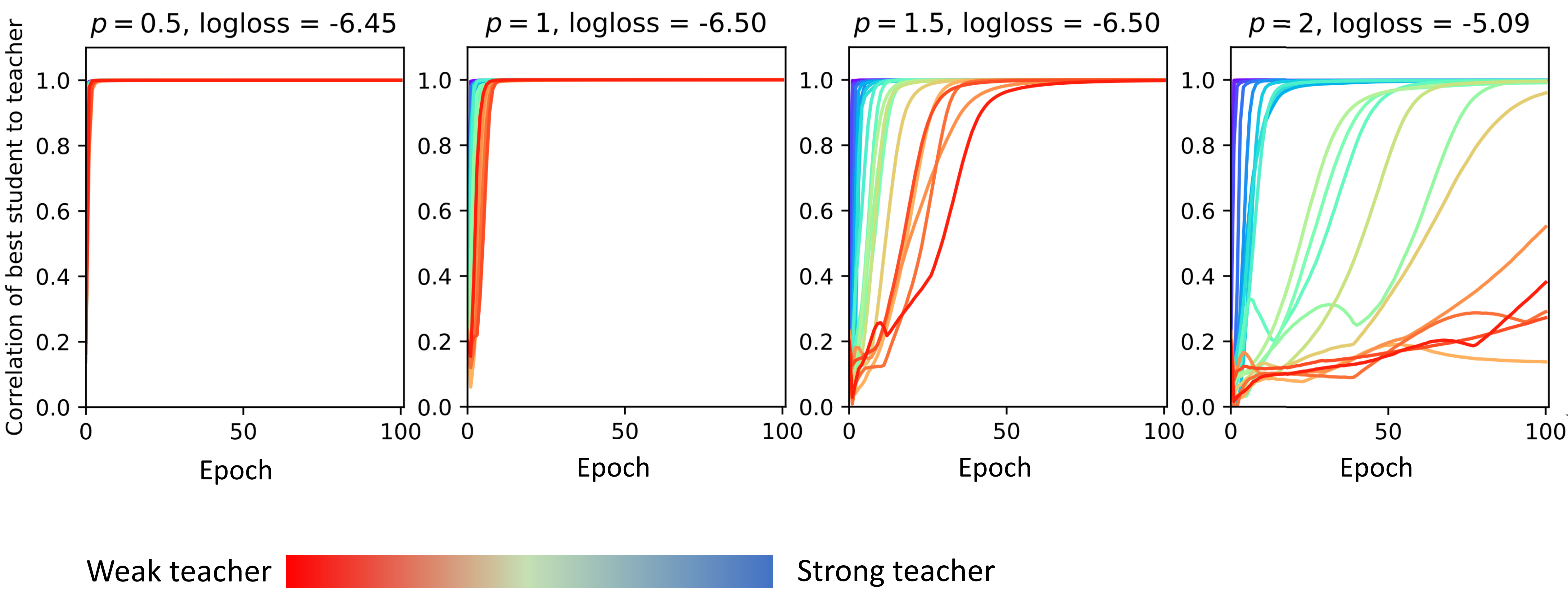
$$\dot{\theta} \propto -\psi(\theta) \sin\theta \propto -(\pi - \theta)^2$$

How Over-parameterization can help?



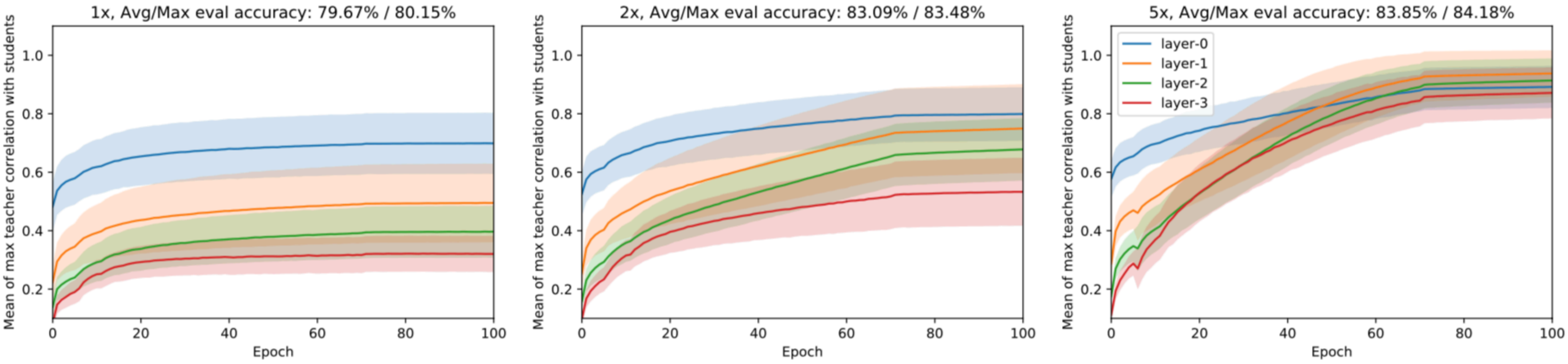
More students yield better coverage.

Weak teacher nodes are Slow to train

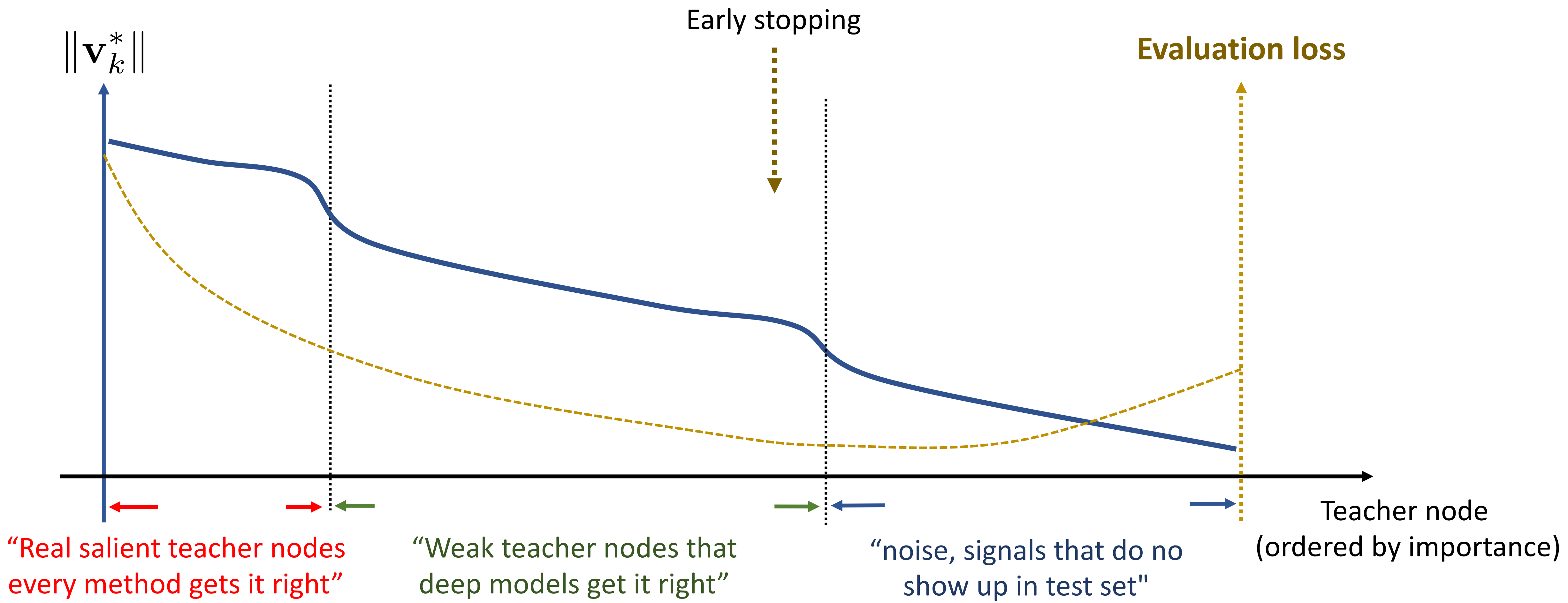


CIFAR 10

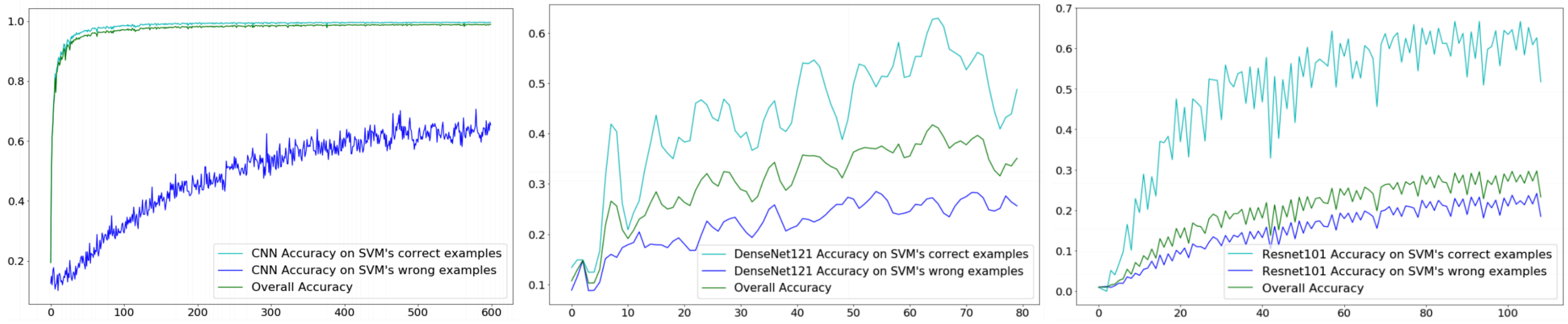
1. Train a teacher network 64-64-64-64.
2. Then prune the teacher network with [0.3,0.5,0.5,0.7] rate.
3. Then train a student network to mimic teacher‘s output (before softmax)



Hypothesis: What does a real dataset look like?



Some Evidences



[Do deep neural networks learn shallow learnable examples first? Mangalam et al, ICML 2019 Workshop]

Future Work

Empirical:

- Large Scale Experiments (ImageNet)
- Relate what analysis tells versus what we see empirically

Theoretical:

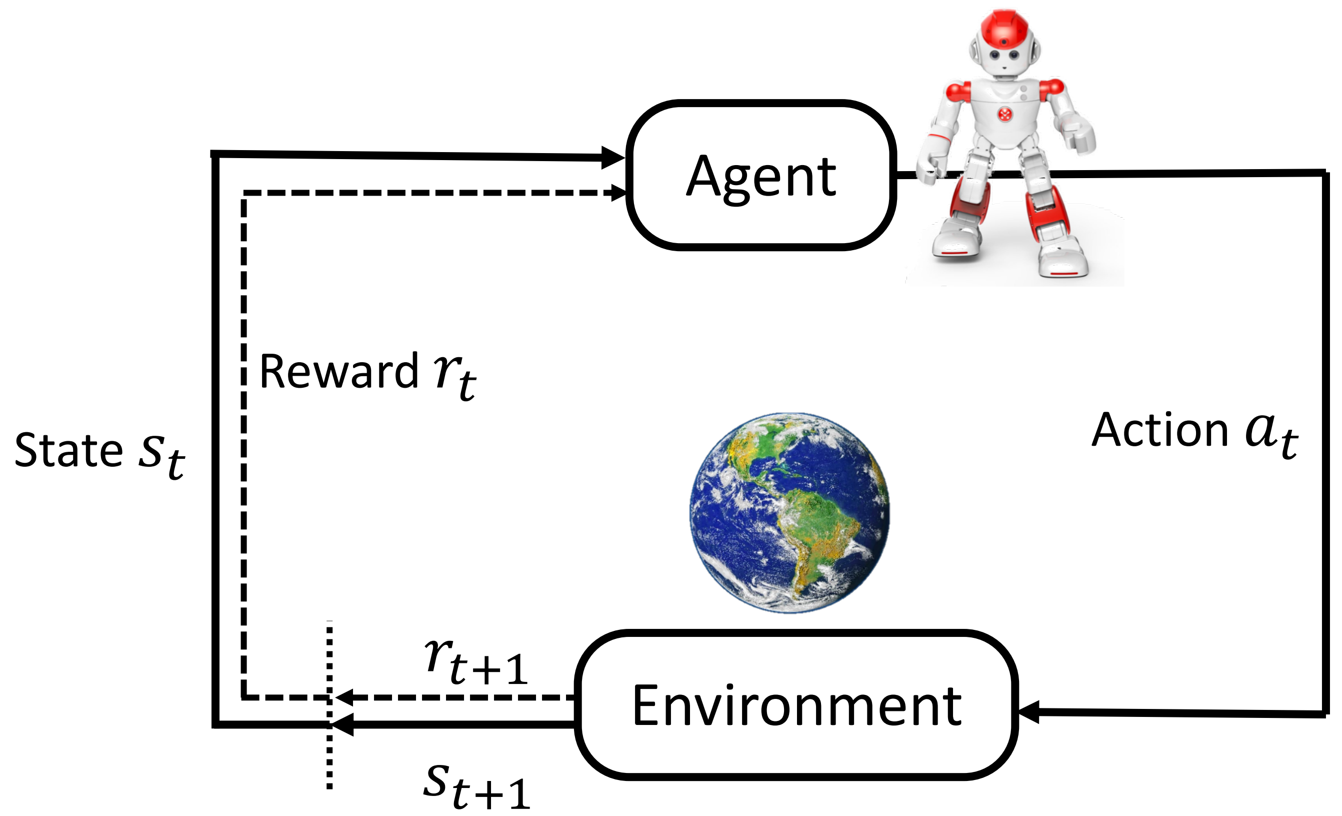
- Finite Sample Analysis to achieve a formal generalization bound
- Bottom-up Training Dynamics of deep ReLU networks
- Training Dynamics of student nodes competing against each other (Competitive Lotka–Volterra equations)

Building Scalable Systems for Reinforcement Learning

Presented by Yuandong Tian

Research Scientist and Manager
Facebook AI Research

Crash Course of Reinforcement Learning



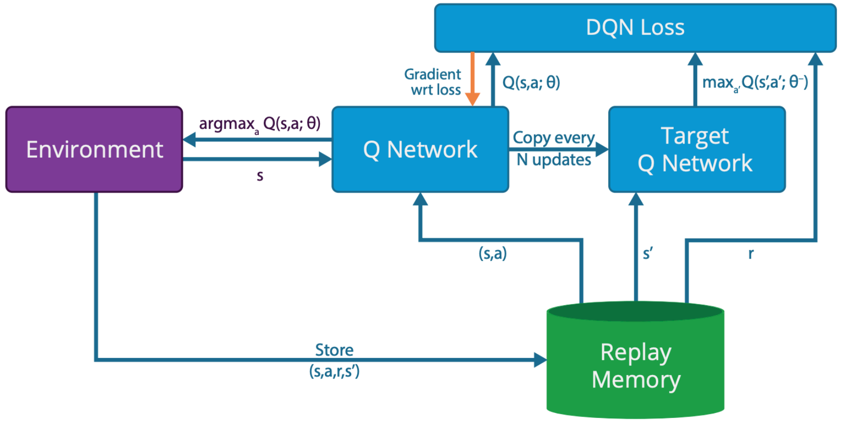
Reinforcement Learning works, but expensive

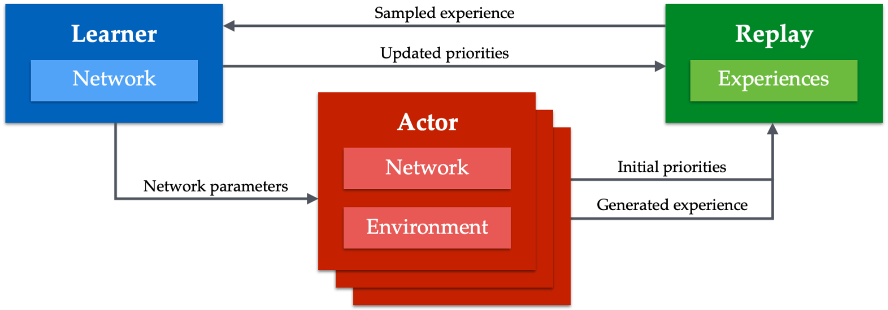
Year	Projects	Human Data	Training Resource	Training time
2016	DeepMind's AlphaGo	Yes	~50 GPUs + ? CPUs	~1 week
2017	DeepMind's AlphaGo Zero (20 blocks)	No	~2000 TPUs	3 days
2017	DeepMind's AlphaZero (20 blocks)	No	~5000 TPUs	8 hours
2018	OpenAI Five	No	128,000 CPUs + 256 GPUs	Several months
2019	DeepMind's AlphaStar	Yes	16,000 CPUs + 3072 TPUv3 cores	44 days

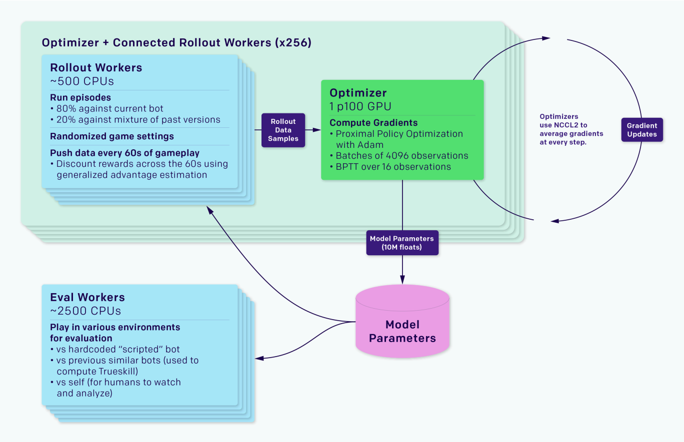
Challenges in large-scale RL Training System

- Trade-offs in a *heterogenous* system
  - **Different kind of objects**: Actor / Environment / Trainer / Replay buffer
  - CPUs / GPUs Allocations
  - Multi-threading versus Multiple Processes, Batching issues
  - Local versus Distributed
  - Synchronization / Asynchronization.
    - On-policy versus off-policy methods
    - Perfect synchronization might NOT give you the best performance
- Mingled Algorithm Design and System Design
  - New System design ←→ New RL algorithm

Distributed System for training RL agent







GORILLA

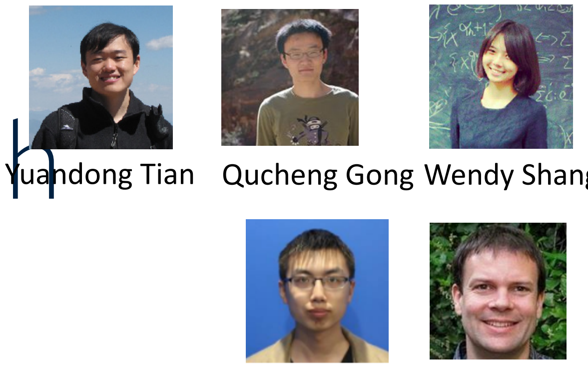
[Massively Parallel Methods for Deep Reinforcement Learning, AAAI 2015]

Ape-X / R2D2

[Distributed Prioritized Experience Replay, Horgan et al, ICLR 2018]
[Recurrent Experience Replay in Distributed Reinforcement Learning Kapturowski et al, ICLR 2019]

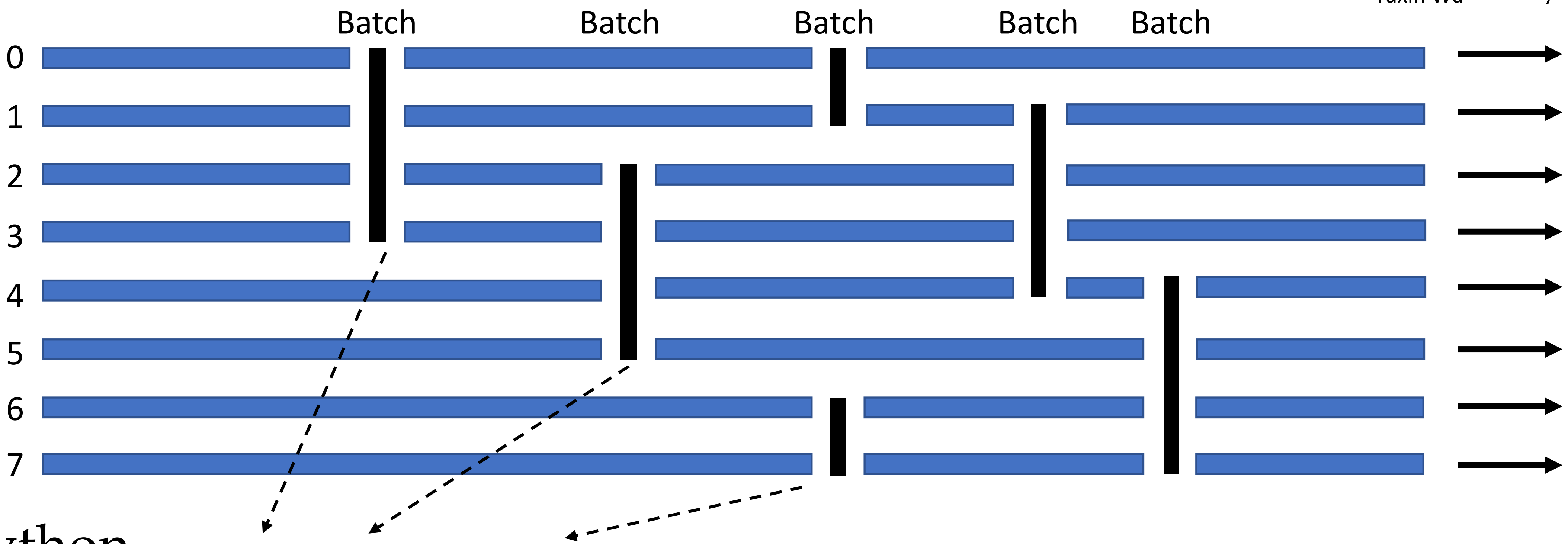
OpenAI Rapid

ELF: RL Framework for Game Research



C++

Game Threads

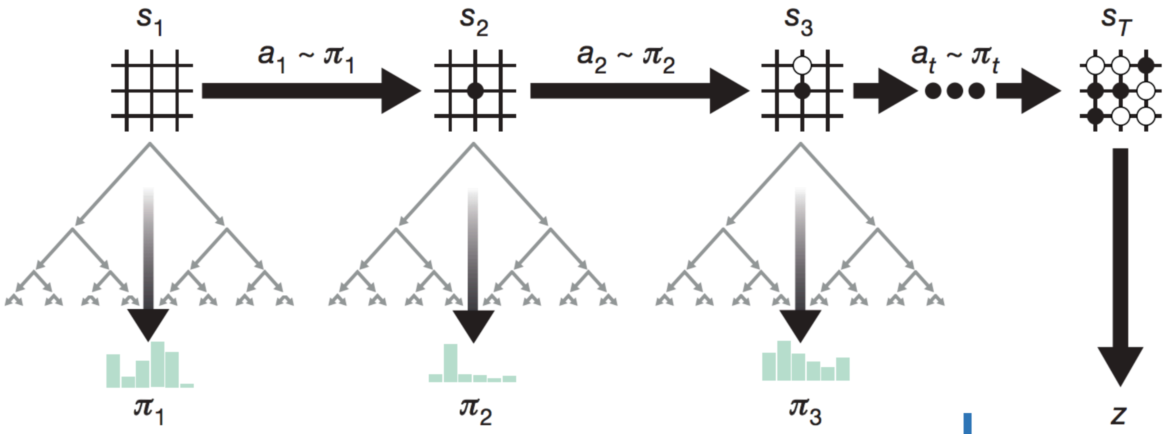


Python

```python
while True:
    batched_states = GameContext.Wait()
    replies = model(batched_states)
    GameContext.Steps(replies)
```

[ELF: Extensive, Lightweight and Flexible Framework for Game Research, Tian et al, NeurIPS 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]

Generate Self-play Games

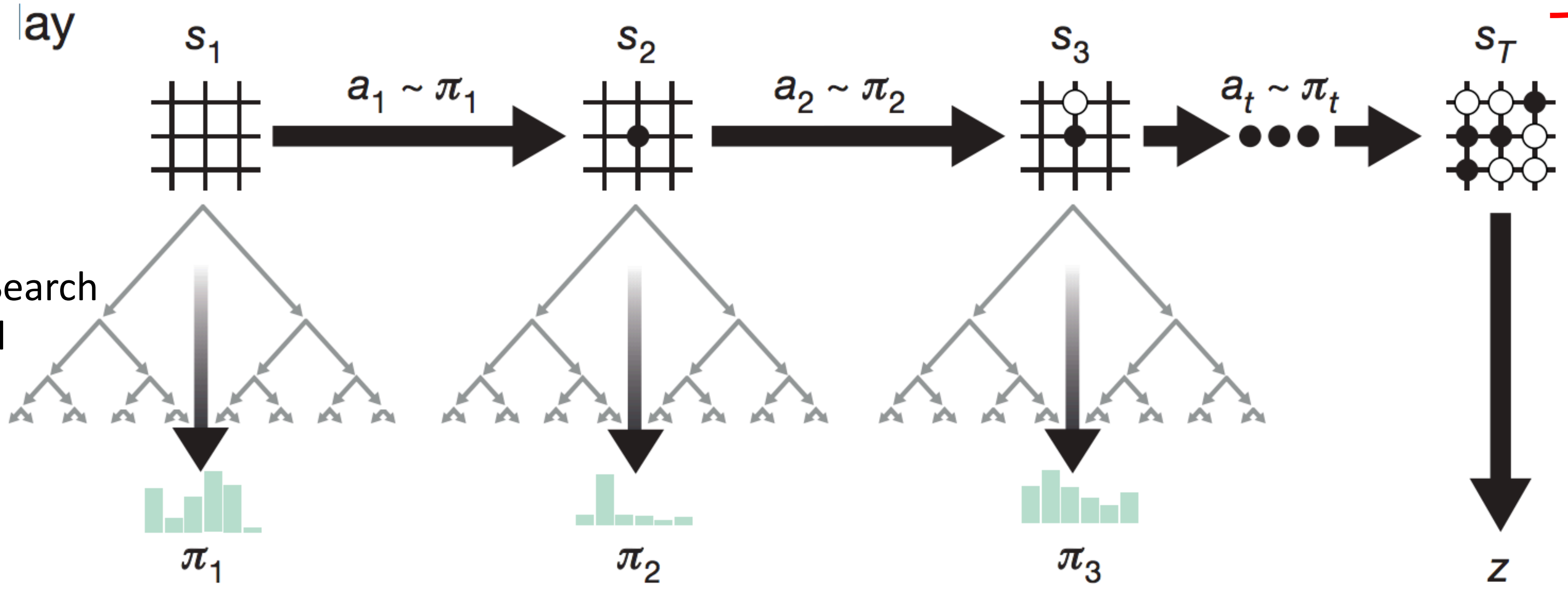
$\boxed{(s_1, \pi_1, z)}$

$\boxed{(s_2, \pi_2, z)}$

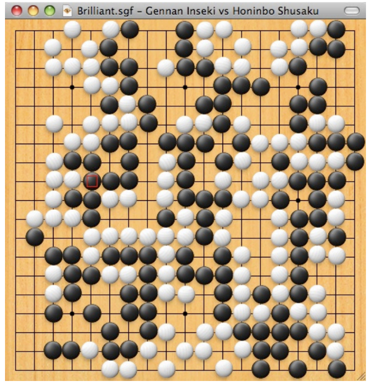
$\boxed{(s_3, \pi_3, z)}$

Training samples for θ_{i+1}

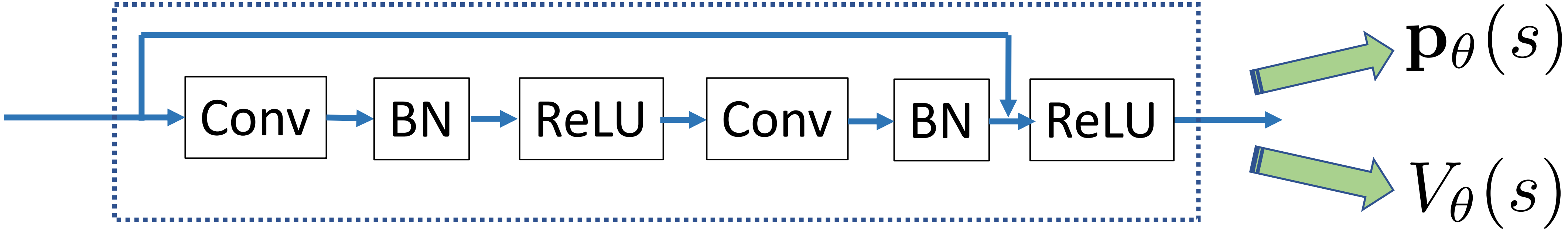
Monte Carlo Tree Search with current model θ_i



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

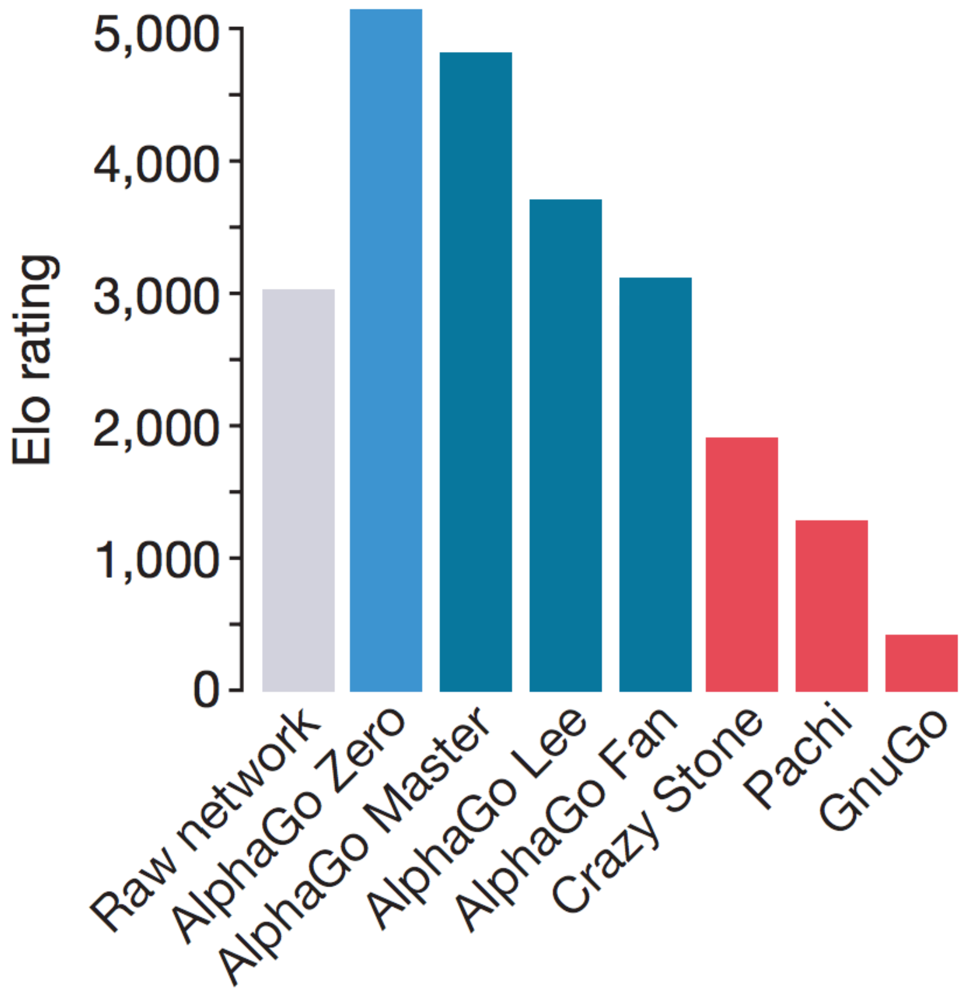
Objective: $(s, \boldsymbol{\pi}, z)$

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

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



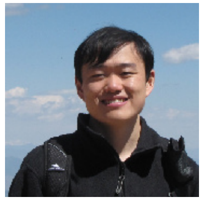
The Mystery of AlphaZero

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

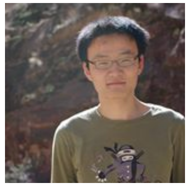
- Lack of Ablation Studies
  - What factor is critical for the performance?
  - Is the algorithm robust to random initialization and changes of hyper parameters?
  - Any adversarial samples?

Impressive Results, No code, No model

ELF OpenGo

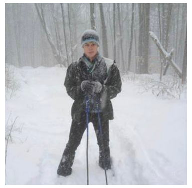

Yuandong Tian


Jerry Ma*


Qucheng Gong*

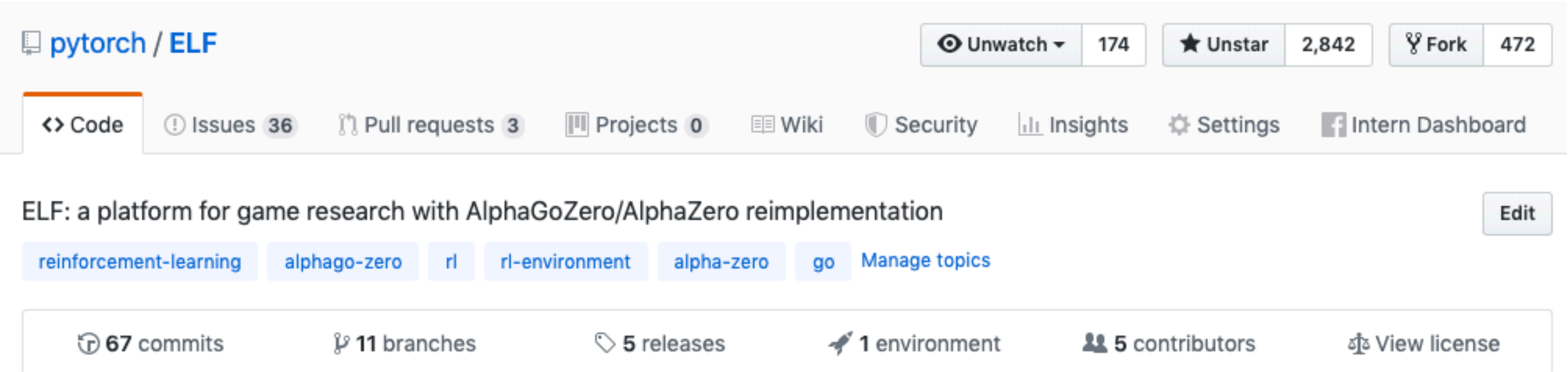

Shubho Sengupta*


Zhuoyuan Chen


James Pinkerton


Larry Zitnick

- System can be trained with 2000 GPUs in 2 weeks (20 block version)
- Superhuman performance against professional players and strong bots.
- Abundant ablation analysis.



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

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

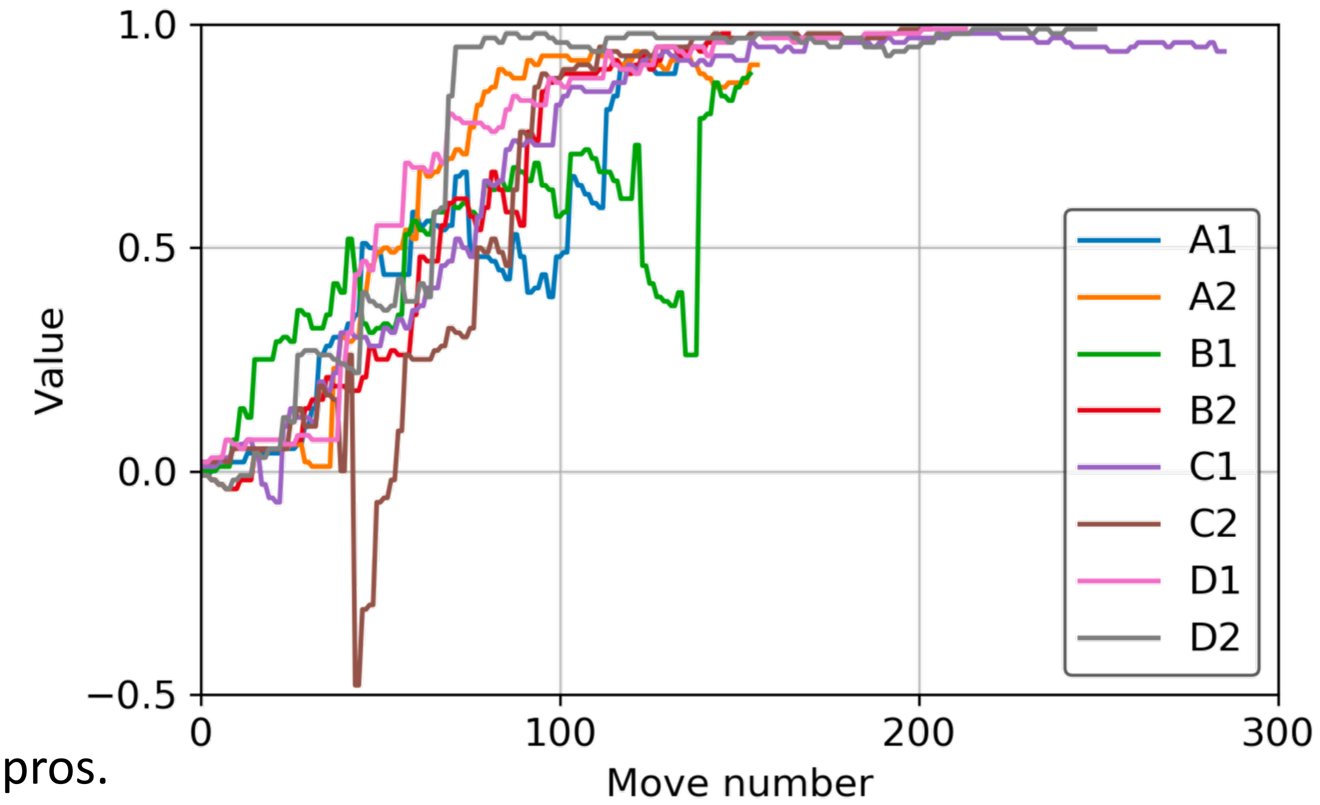
ELF OpenGo Performance

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



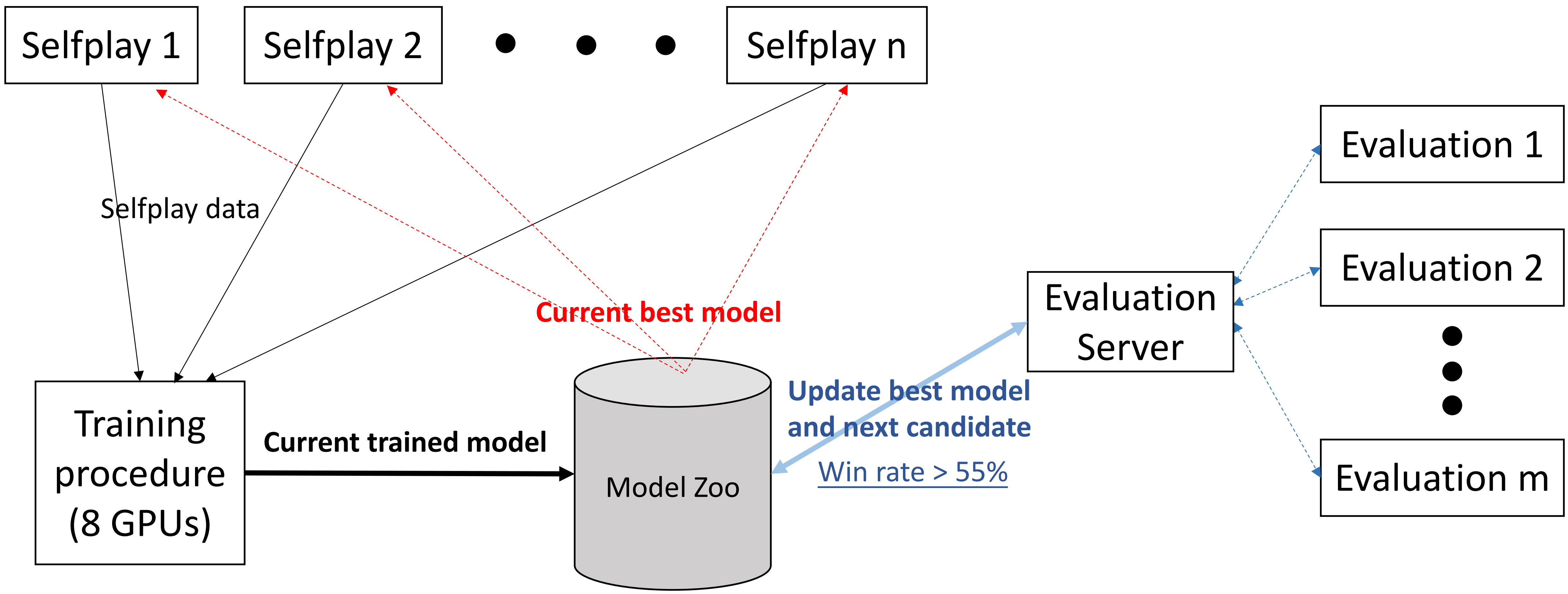
Vs professional players

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

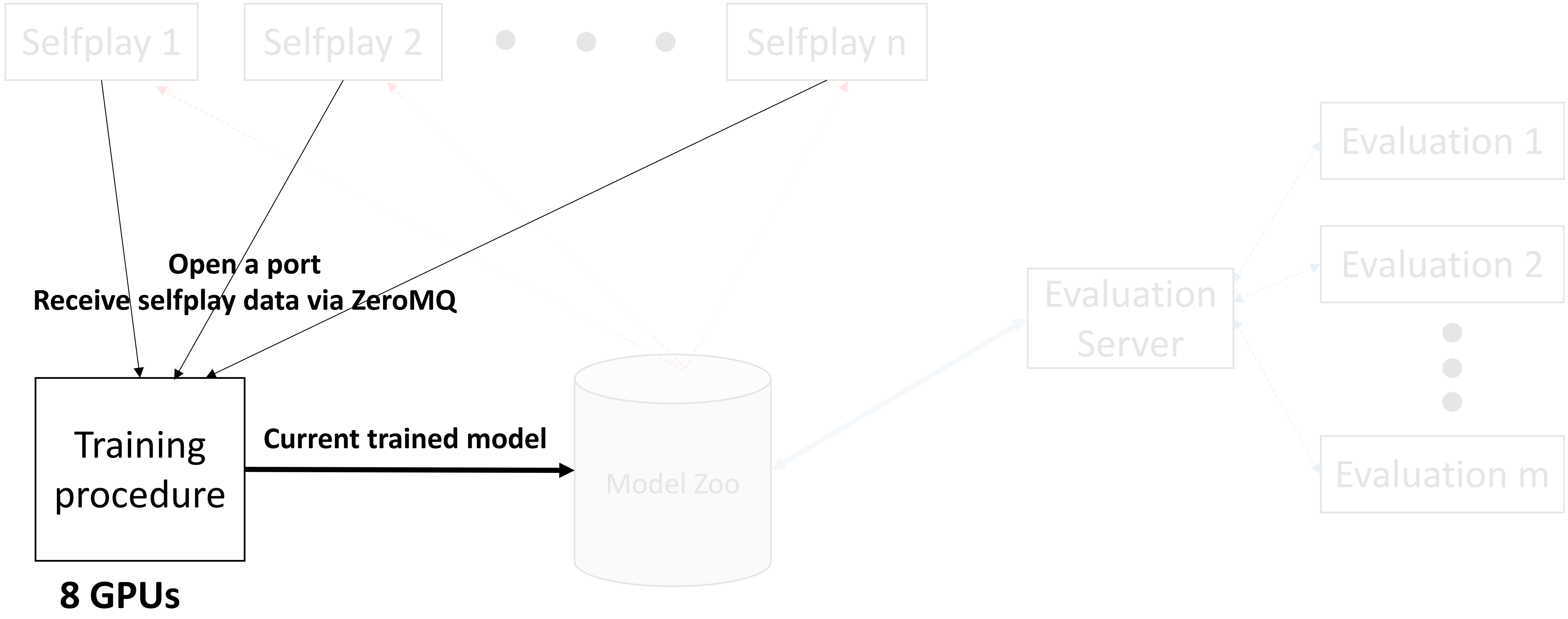
Vs strong bot (LeelaZero)

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

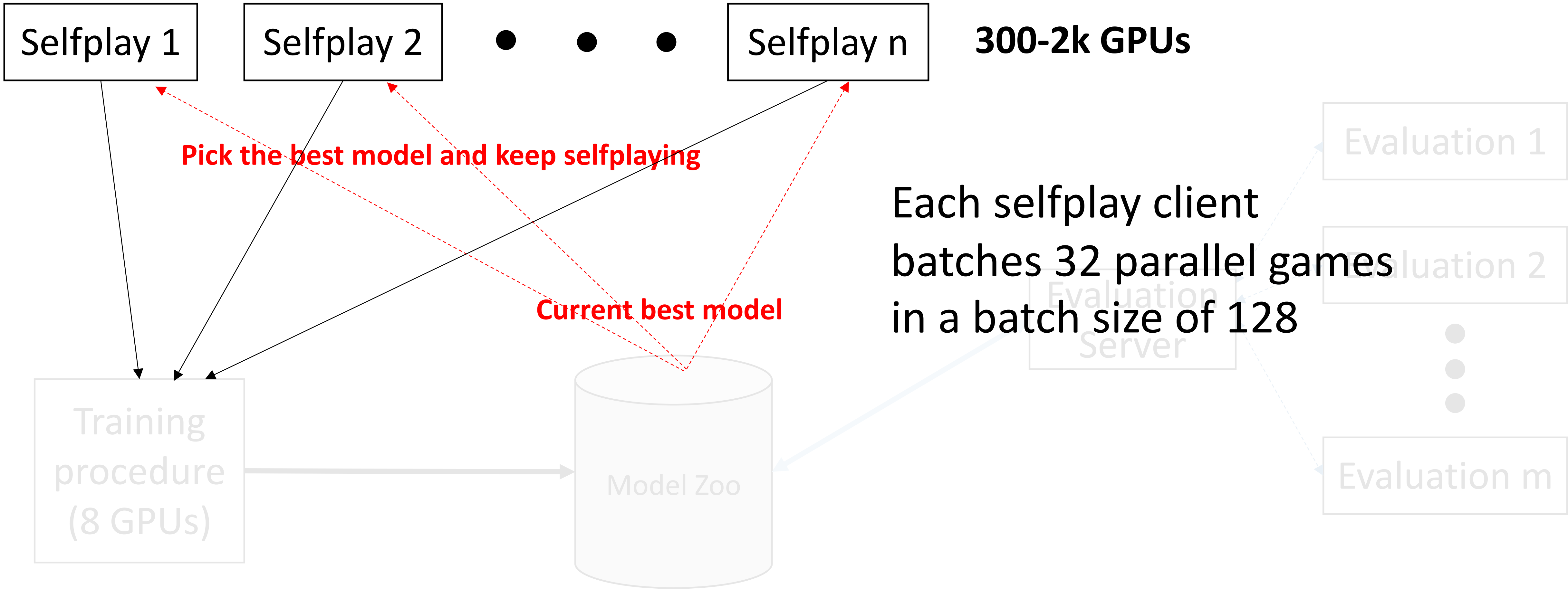
Distributed ELF (version 1, AlphaGoZero)



Distributed ELF (version 1)

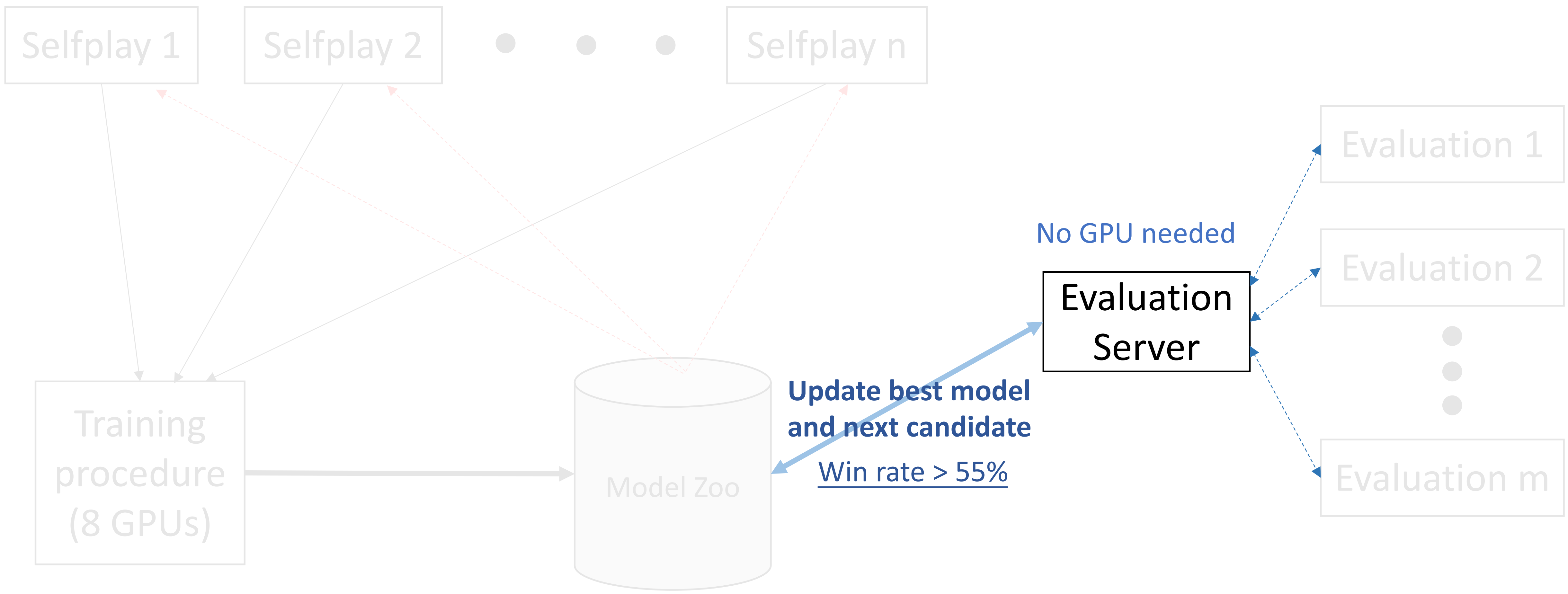


Distributed ELF (version 1)

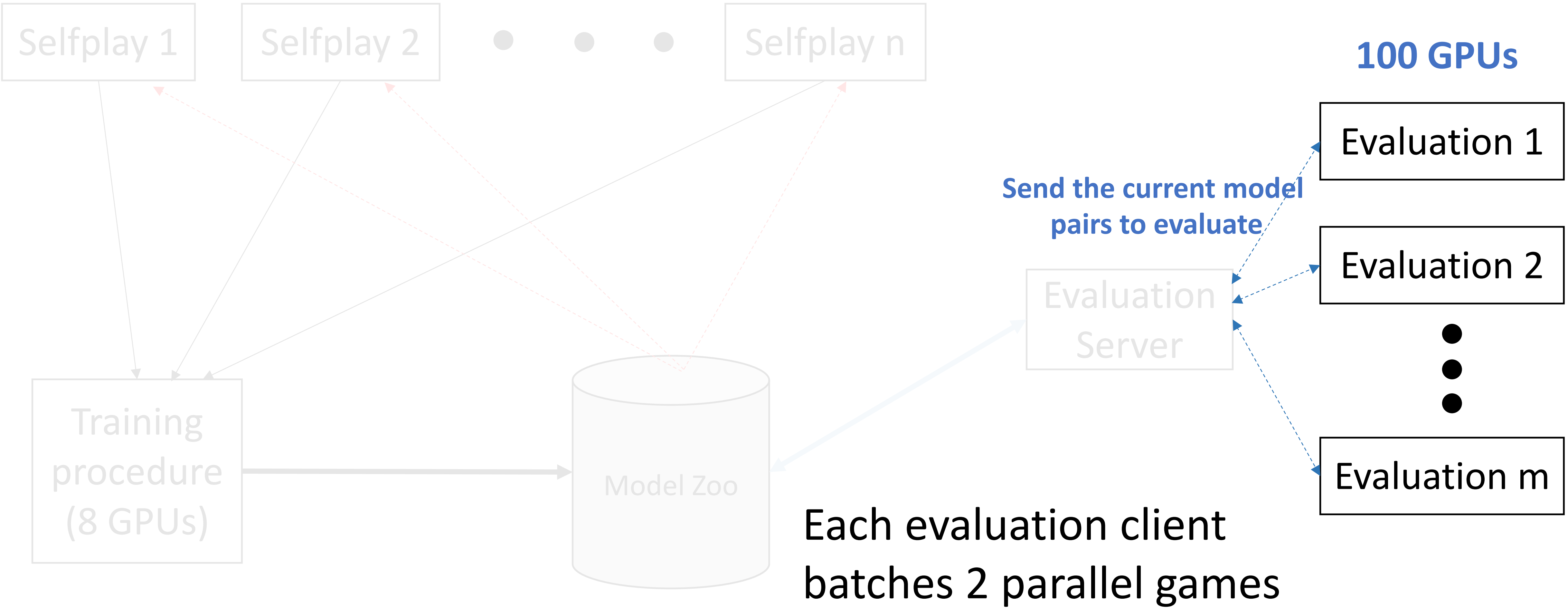


Each selfplay client batches 32 parallel games in a batch size of 128

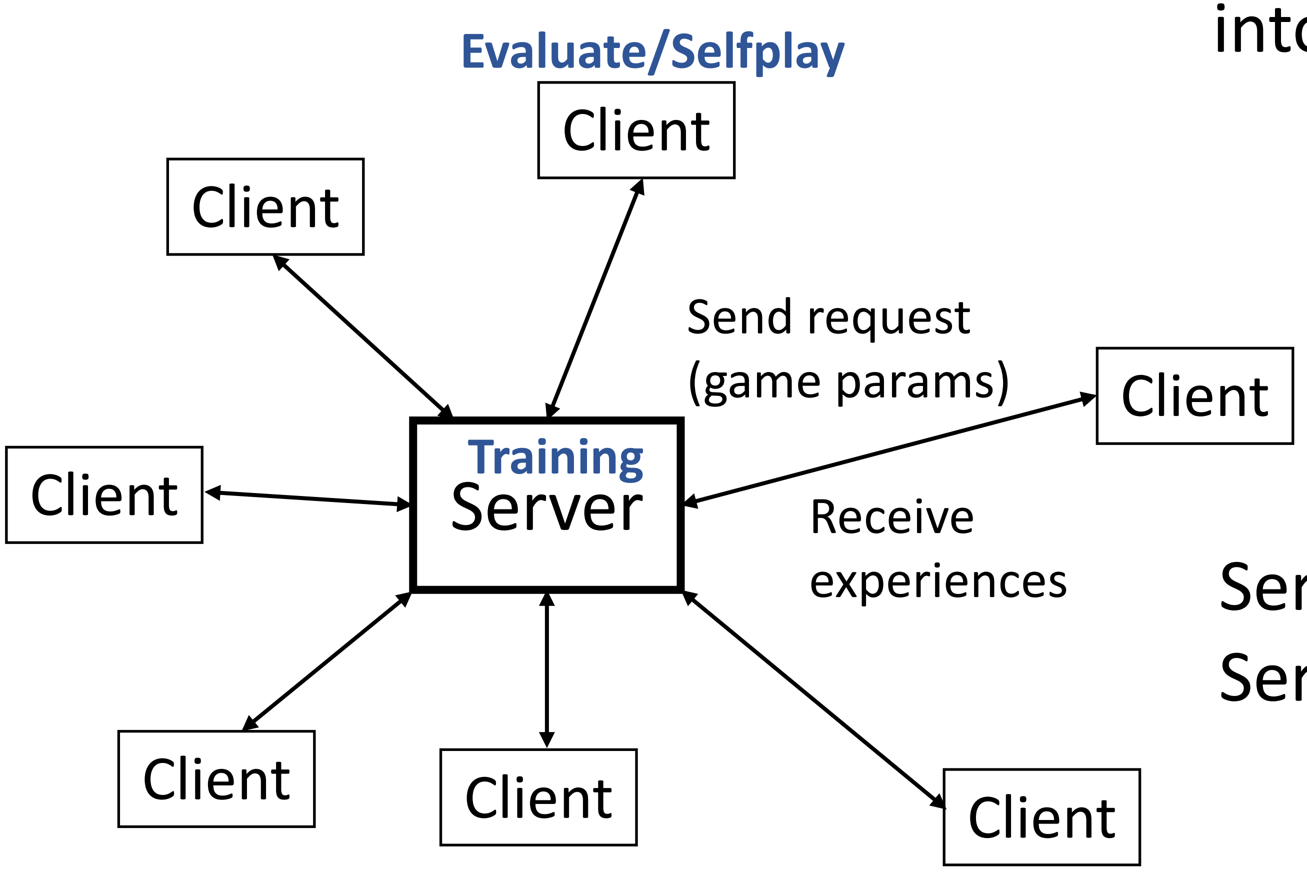
Distributed ELF (version 1)



Distributed ELF (version 1)



Distributed ELF (v2)



Putting AlphaGoZero and AlphaZero into the same framework

AlphaGoZero (more synchronization)
AlphaZero (less synchronization)

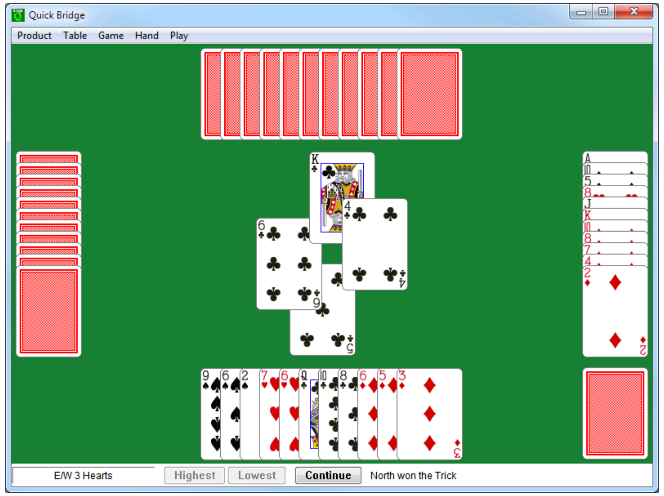
Server controls synchronization
Server also does training.

Next Step: RL Assembly

- Backbone infrastructure for ongoing projects (Hanabi, Bridge, etc)
- Reimplementation of SoTA off-policy RL methods like Ape-X and R2D2
- Incorporate OpenGo and SoTA implementation of MCTS.
- Efficient on single machine (SoTA training FPS so far)

Open source soon

Current Projects using ReLA



Contract Bridge

Hanabi

MiniRTSv2

More projects to come!

Thanks!