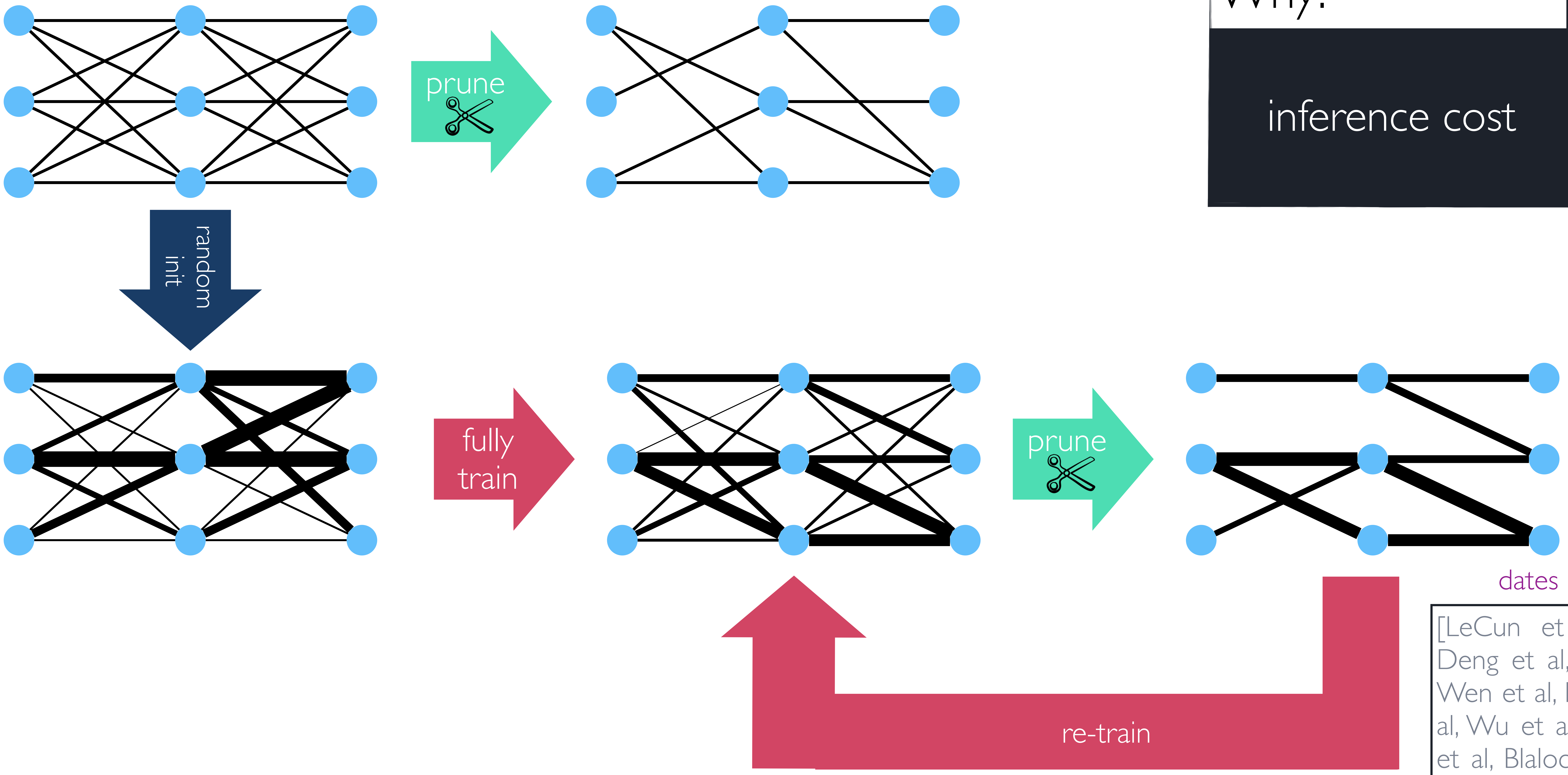
From Model Pruning to Sparse Updates and the Lottery Ticket Hypothesis

Network Pruning, 1980-2018



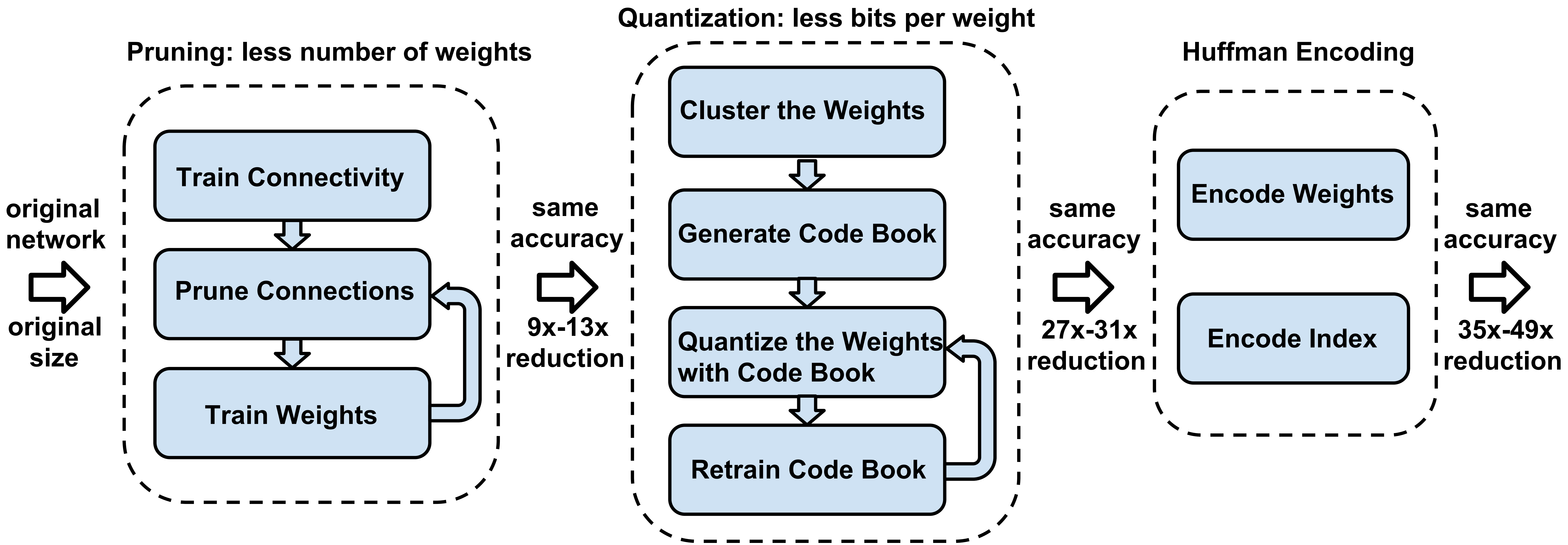
Why?

inference cost

dates back to 80s

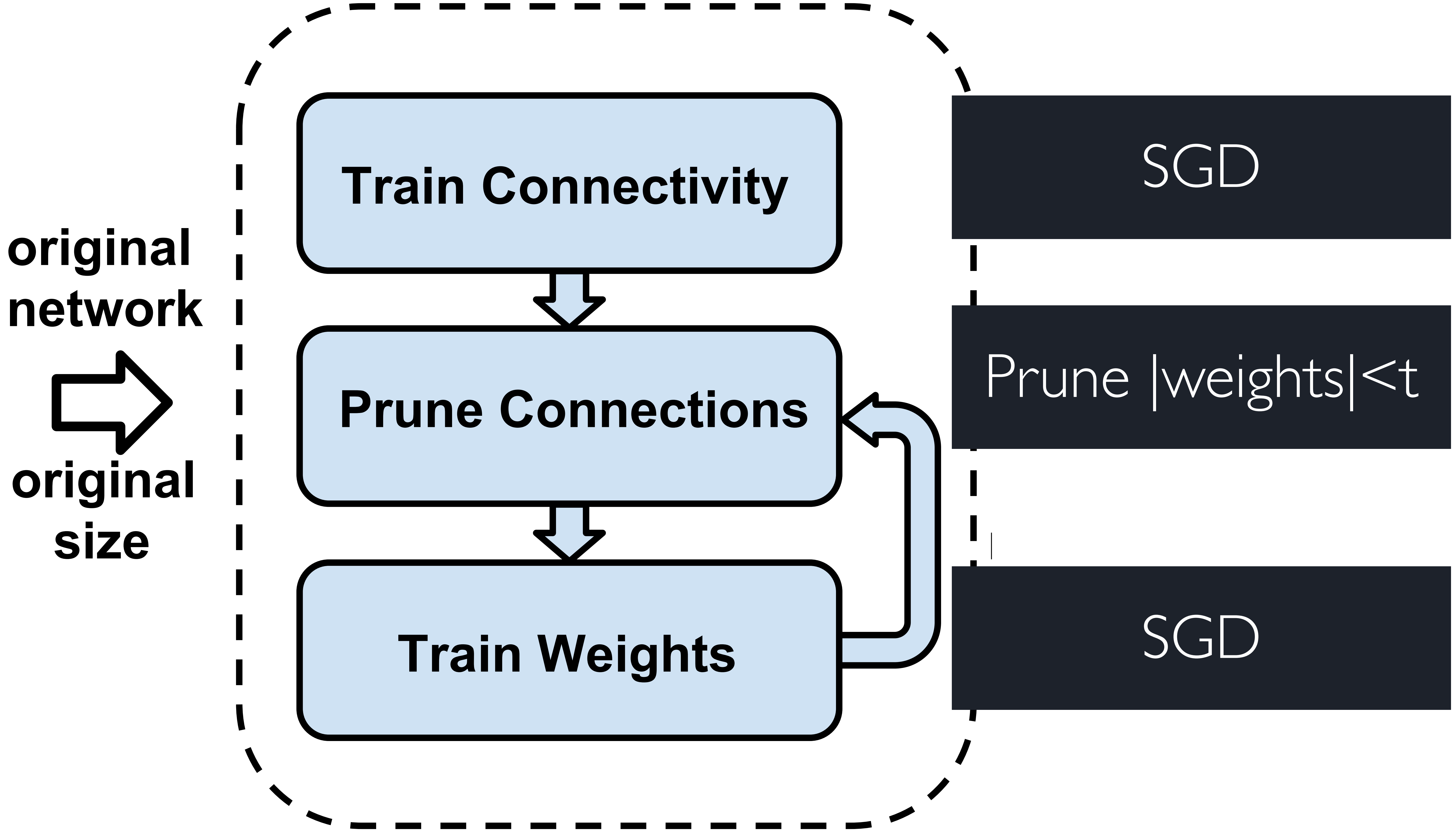
[LeCun et al, Hassibi et al, Deng et al, Han et al,Li et al, Wen et al, Hubara et al, He et al, Wu et al, Zhu et al, Cheng et al, Blalock et al, Levin et al, Mozer et al]

An example: Deep Compression [ICLR, 2016]



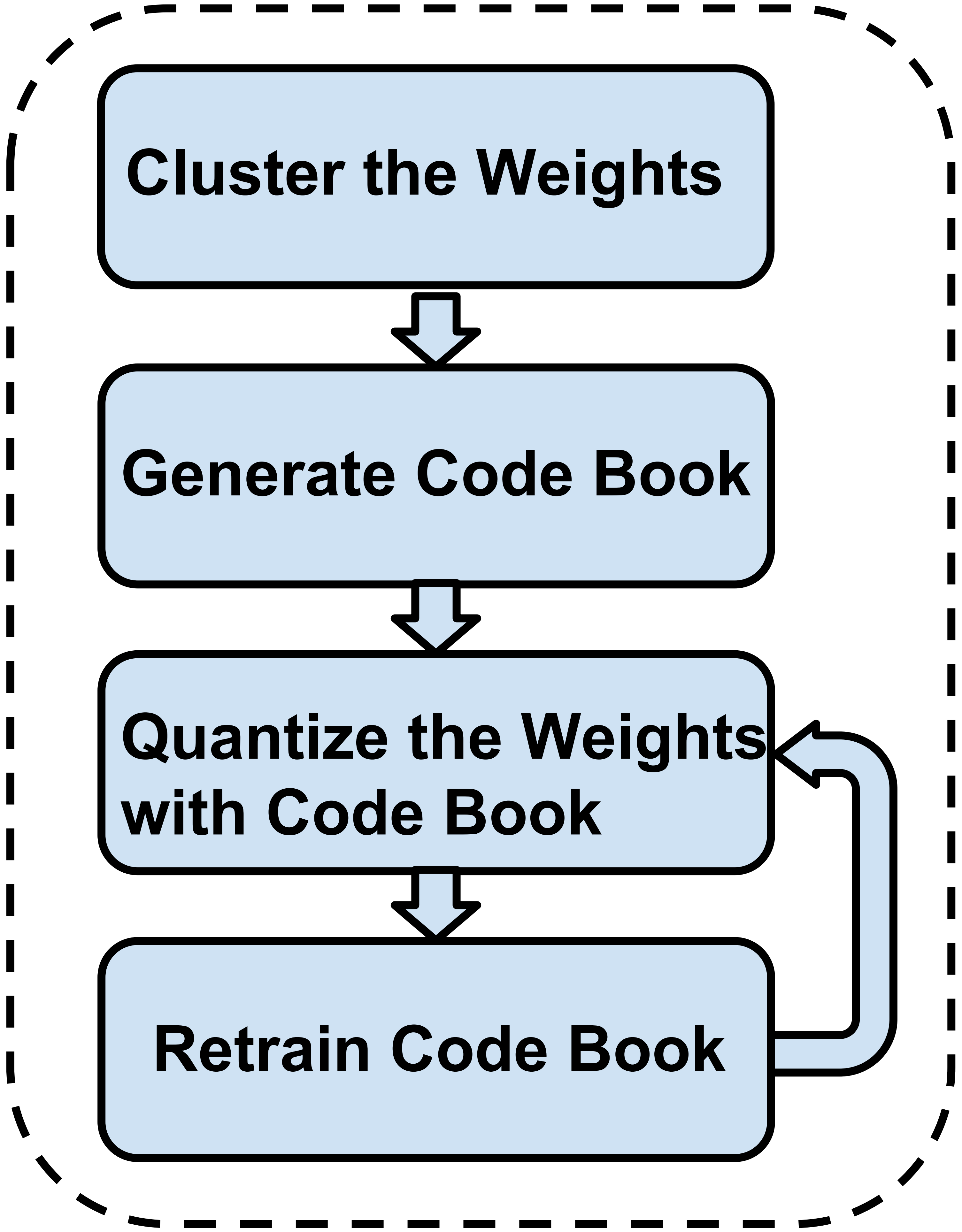
[Han, Mao, Dally, ICLR 2016]

Deep Compression: Step 1, prune



[Han, Mao, Dally, ICLR 2016]

Deep Compression: Step 2, quantize

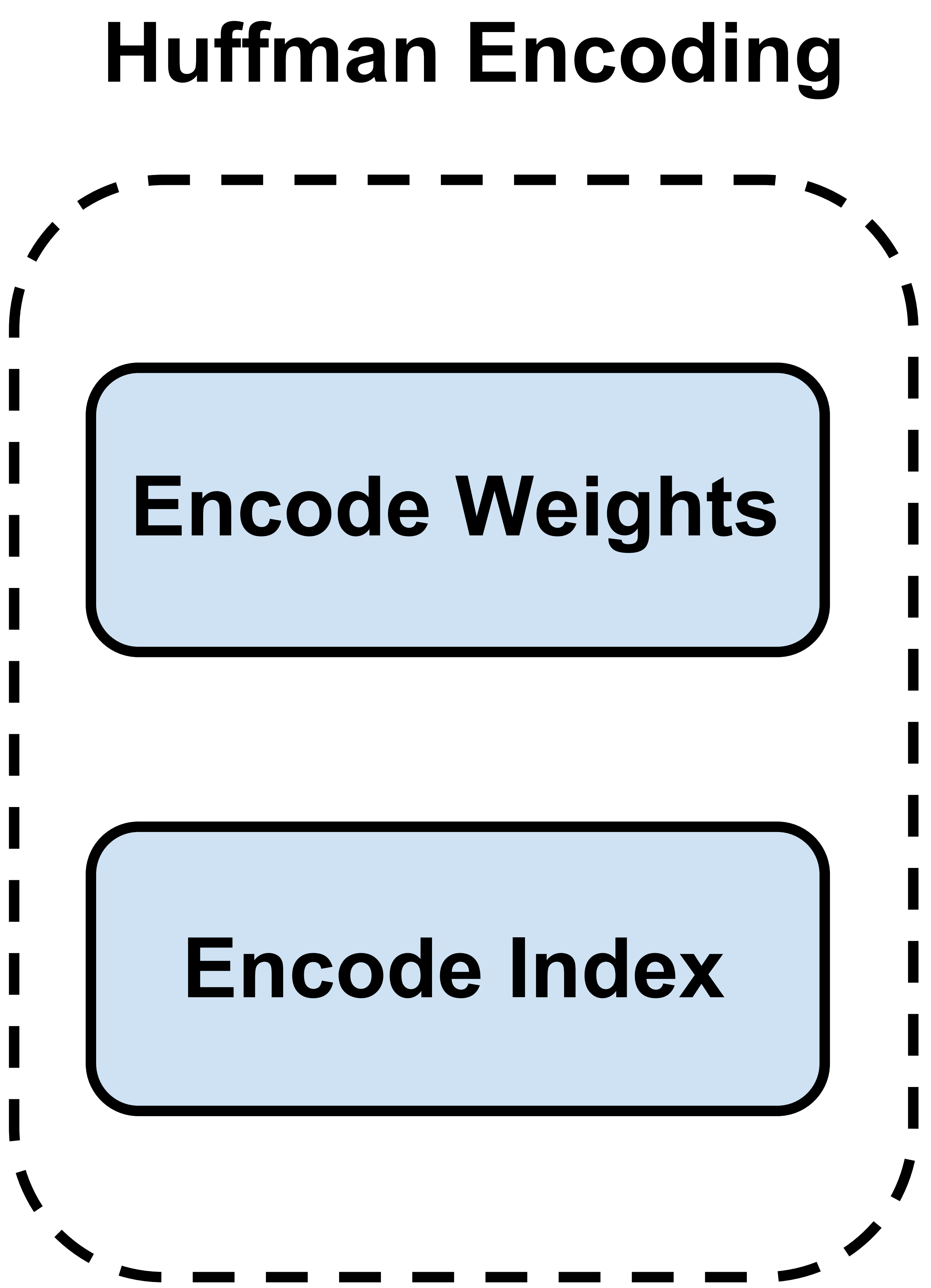


$$\min_{S_1,\ldots,S_k} \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu_i\|^2$$

K-means used
for clustering

[Han, Mao, Dally, ICLR 2016]

Deep Compression: Step 3, compress

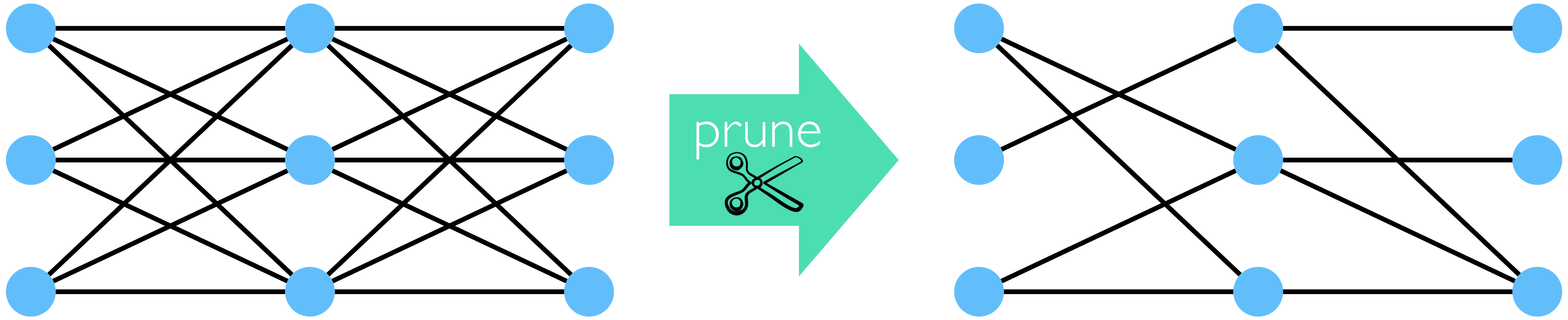


Deep Compression: Experiments

Network	Top-1 Error	Top-5 Error	Parameters	Compress Rate
LeNet-300-100 Ref	1.64%	-	1070 KB	
LeNet-300-100 Compressed	1.58%	-	**27 KB**	**40**×
LeNet-5 Ref	0.80%	-	1720 KB	
LeNet-5 Compressed	0.74%	-	**44 KB**	**39**×
AlexNet Ref	42.78%	19.73%	240 MB	
AlexNet Compressed	42.78%	19.70%	**6.9 MB**	**35**×
VGG-16 Ref	31.50%	11.32%	552 MB	
VGG-16 Compressed	31.17%	10.91%	**11.3 MB**	**49**×

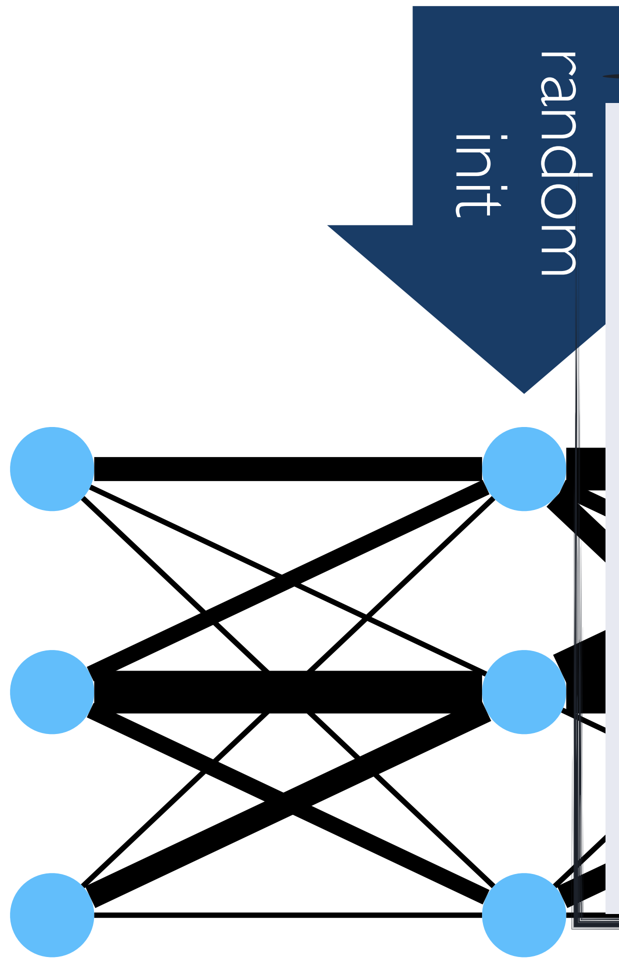
network pruning works!

Network Pruning, 1980-2018



Why?

inference cost



Issue:
training to full acc and THEN pruning
is expensive
Q: Can we avoid?

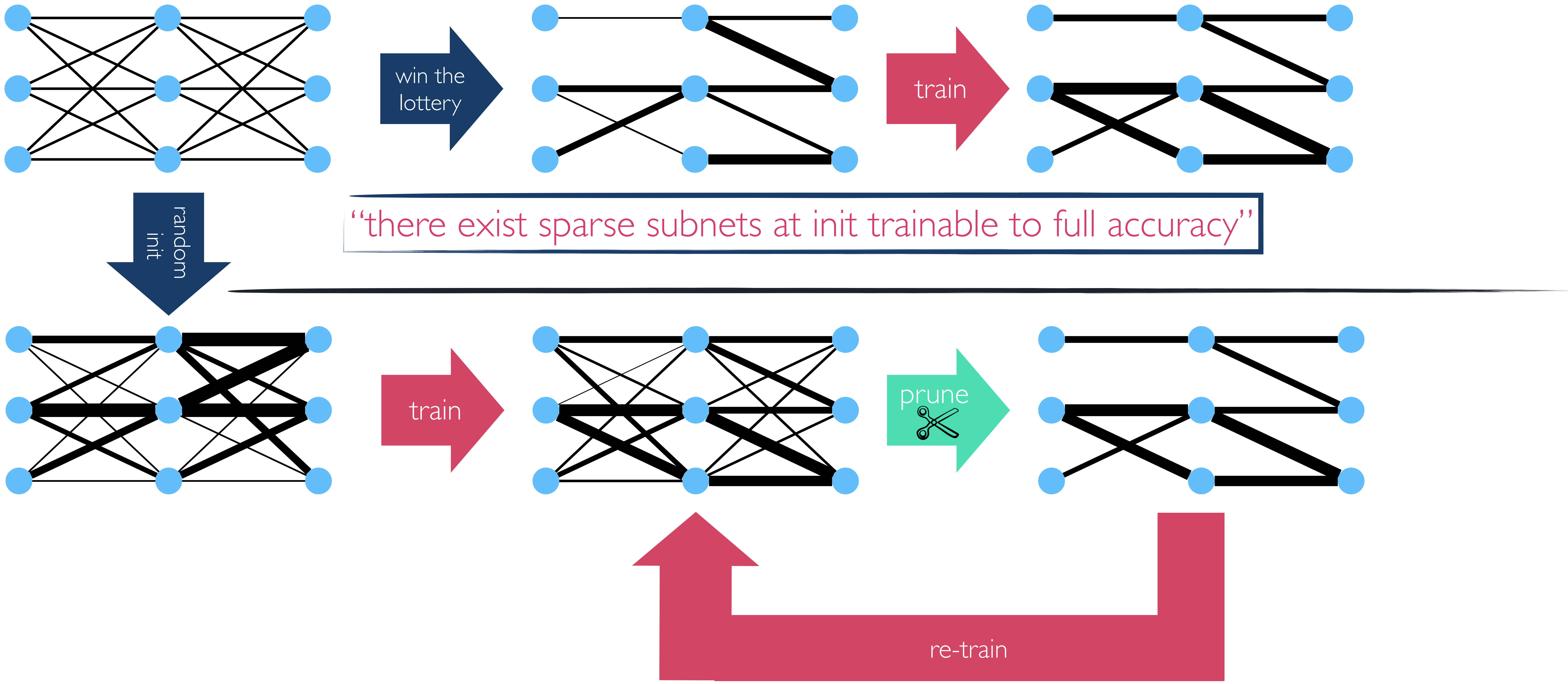
dates back to 80s

[LeCun et al, Hassibi et al, Deng et al, Han et al,Li et al, Wen et al, Hubara et al, He et al, Wu et al, Zhu et al, Cheng et al, Blalock et al, Levin et al, Mozer et al]

The Lottery Ticket Hypothesis

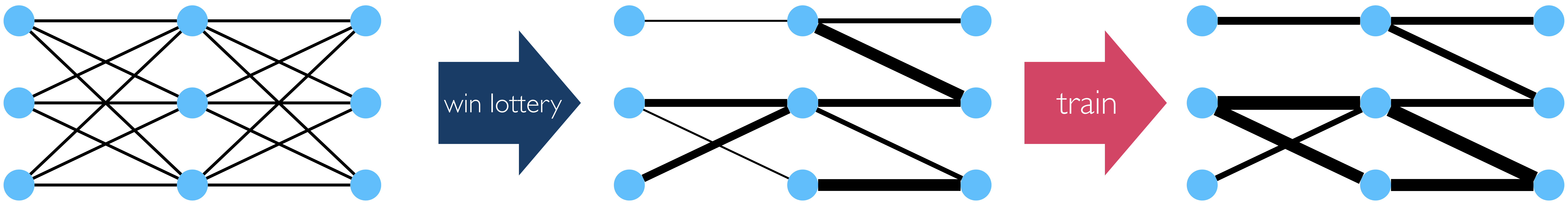
Lottery Ticket Hypothesis (LTH)

Frankle, Carbin, ICLR 2019



Lottery Ticket Hypothesis (LTH)

Frankle, Carbin, ICLR 2019



"there exist sparse subnetworks at init that can be trained to full accuracy"

Identify

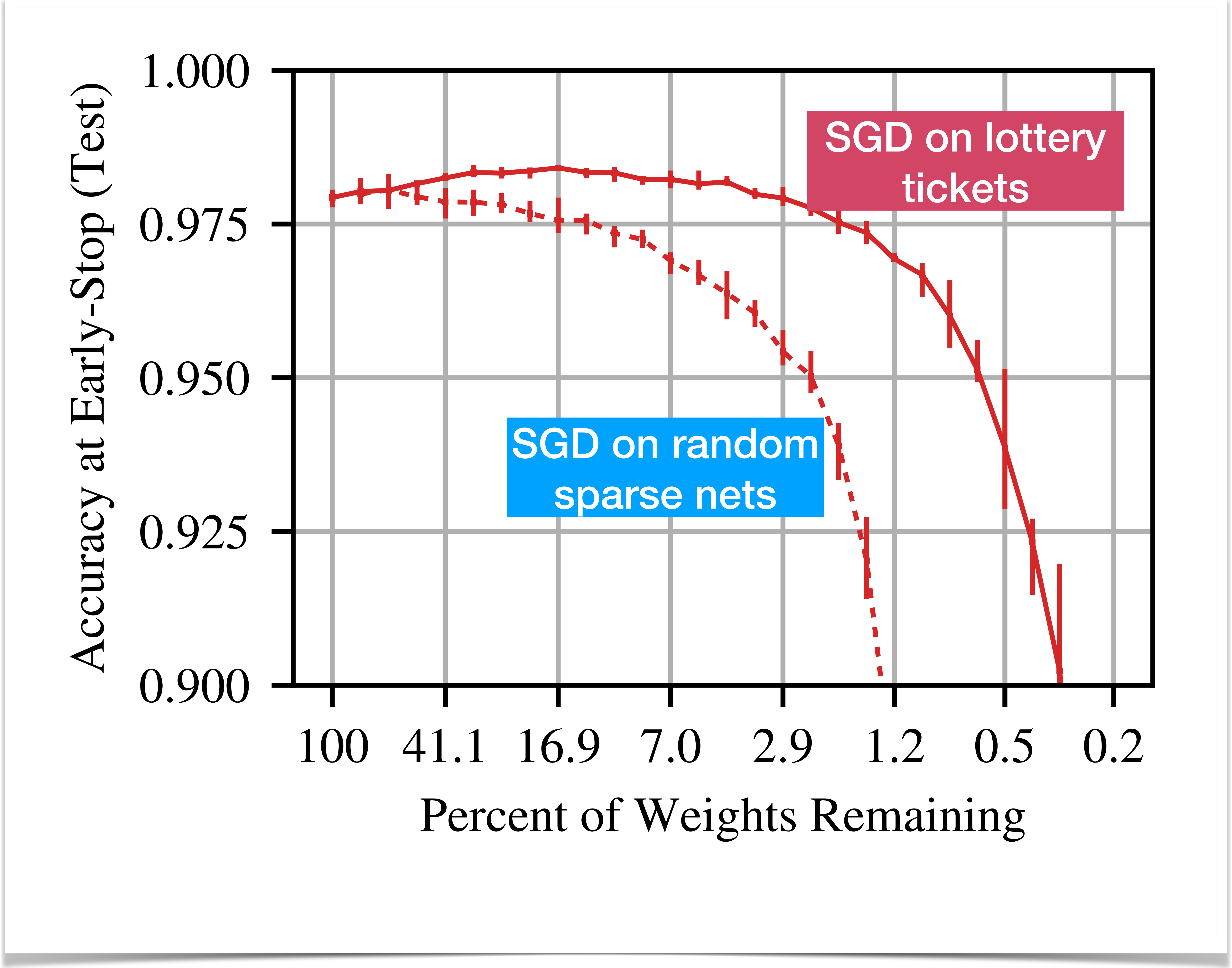
If true, kinda big deal!
We can avoid pruning/retraining cycle

Q: How do you win the lottery??

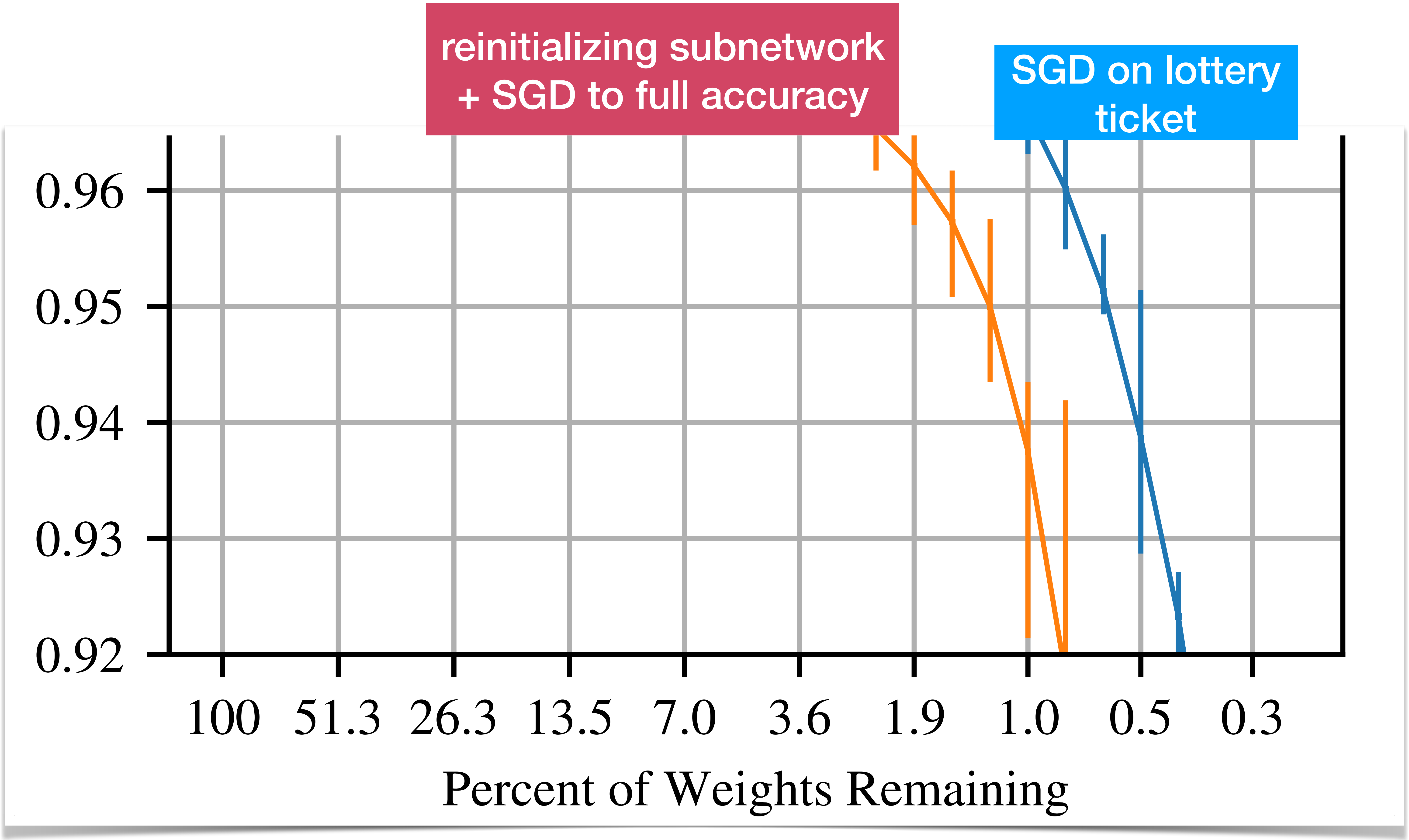
5. (Sometimes) GOTO 3

note: winning the lottery does not come for free

Lottery tickets >> random subnets



lottery ticket = subnetwork + original weights



<u>LTH research has been very active:</u>

[Zhou, Lan, Liu, Yosinski]
[Frankle, Dziugaite, Roy, Carbin]
[Cosentino, Zaiter, Pei, Zhu]
[Soelen, Sheppard]
[Sabatelli, Kestemont, Geurts]
[Ramanujan, Wortsman, Kembhavi, Farhadi, Rastegari]
[Wang, Zhang, Xie, Zhou, Su, Zhang, Hu]

Many many extensions

The Lottery Ticket Hypothesis for Pre-trained BERT Networks

Tianlong Chen[1], Jonathan Frankle[2], Shiyu Chang[3], Sijia Liu[3], Yang Zhang[3], Zhangyang Wang[1], Michael Carbin[2]
[1]University of Texas at Austin, [2]MIT CSAIL, [3]MIT-IBM Watson AI Lab, IBM Research
{tianlong.chen,atlaswang}@utexas.edu,{jfrankle,mcarbin}@csail.mit.edu,
{shiyu.chang,sijia.liu,yang.zhang2}@ibm.com

One ticket to win them all: generalizing lottery ticket initializations across datasets and optimizers

Ari S. Morcos*
Facebook AI Research
arimorcos@fb.com

Haonan Yu
Facebook AI Research
haonanu@gmail.com

Michela Paganini
Facebook AI Research
michela@fb.com

Yuandong Tian
Facebook AI Research
yuandong@fb.com

One ticket to win them all: generalizing lottery ticket initializations across datasets and optimizers

Ari S. Morcos*
Facebook AI Research
arimorcos@fb.com

Haonan Yu
Facebook AI Research
haonanu@gmail.com

Michela Paganini
Facebook AI Research
michela@fb.com

Yuandong Tian
Facebook AI Research
yuandong@fb.com

Published as a conference paper at ICLR 2020

DRAWING EARLY-BIRD TICKETS: TOWARDS MORE EFFICIENT TRAINING OF DEEP NETWORKS

Haoran You, Chaojian Li, Pengfei Xu, Yonggan Fu, Yue Wang, Richard G. Baraniuk & Yingyan Lin*
Department of Electrical and Computer Engineering
Rice University
Houston, TX 77005, USA
{hy34, cl114, px5, yf22, yw68, yingyan.lin, richb}@rice.edu

Xiaohan Chen & Zhangyang Wang*
Department of Computer Science and Engineering
Texas A&M University
College Station, TX 77843, USA
{chernxh, atlaswang}@tamu.edu

Rigging the Lottery: Making All Tickets Winners

Utku Evci[1] Trevor Gale[1] Jacob Menick[2] Pablo Samuel Castro[1] Erich Elsen[2]

PUFFERFISH: COMMUNICATION-EFFICIENT MODELS AT NO EXTRA COST

Hongyi Wang,[1] Saurabh Agarwal,[1] Dimitris Papailiopoulos[2]

Challenges of Pruning At Initialization

Pruning at initialization does’t seem work

Published as a conference paper at ICLR 2019

RETHINKING THE VALUE OF NETWORK PRUNING

Zhuang Liu[1*], Mingjie Sun[2*†], Tinghui Zhou[1], Gao Huang[2], Trevor Darrell[1]
[1]University of California, Berkeley [2]Tsinghua University

Published as a conference paper at ICLR 2021

PRUNING NEURAL NETWORKS AT INITIALIZATION: WHY ARE WE MISSING THE MARK?

Jonathan Frankle, MIT CSAIL
Gintare Karolina Dziugaite, Element AI
Daniel M. Roy, University of Toronto, Vector Institute
Michael Carbin, MIT CSAIL

Random networks areas good as (or better than) claimed lottery tickets..

Sanity-Checking Pruning Methods: Random Tickets can Win the Jackpot

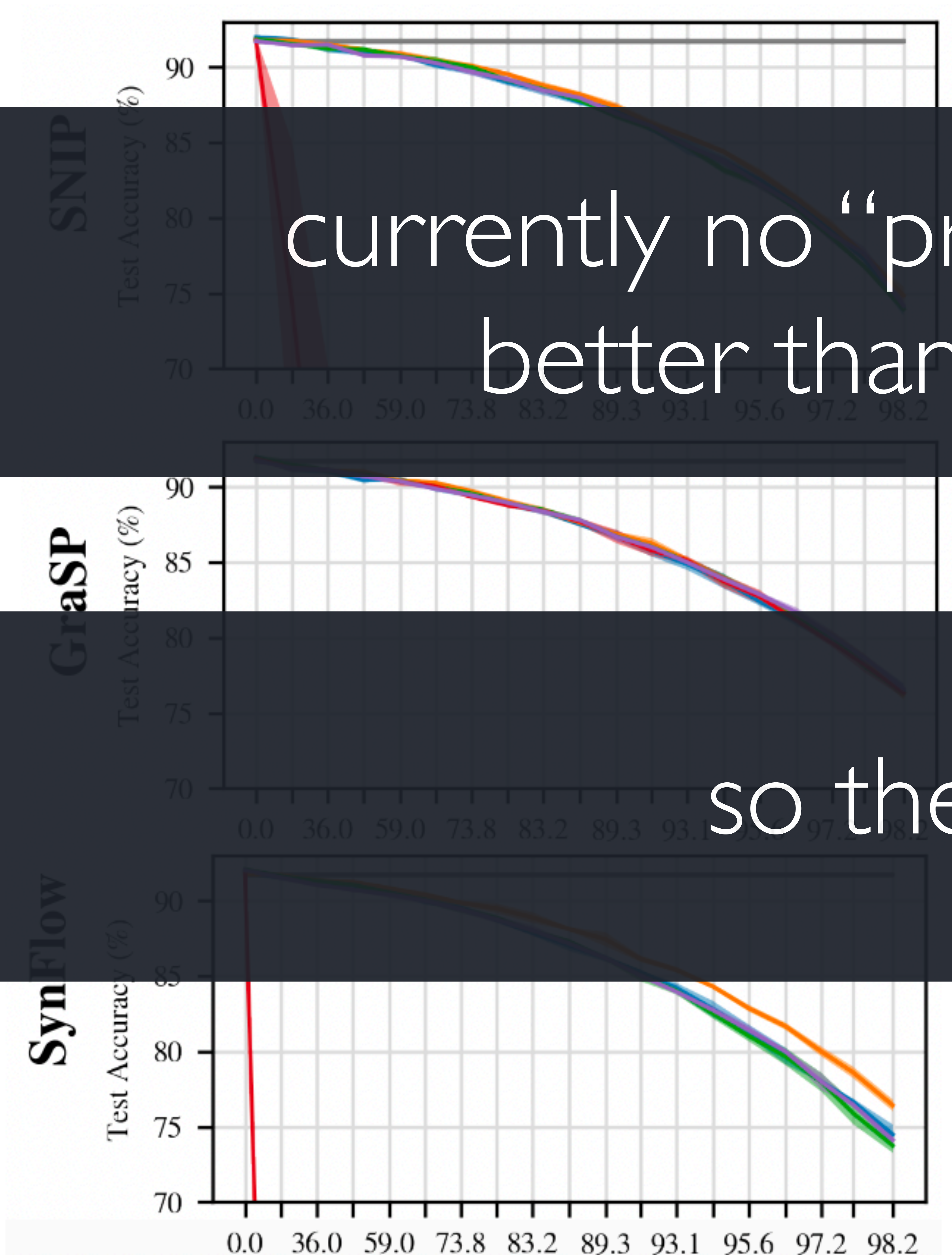
Jingtong Su[1*] Yihang Chen[2*] Tianle Cai[3,4*]
Tianhao Wu[2] Ruiqi Gao[3,4] Liwei Wang[5] Jason D. Lee[3]

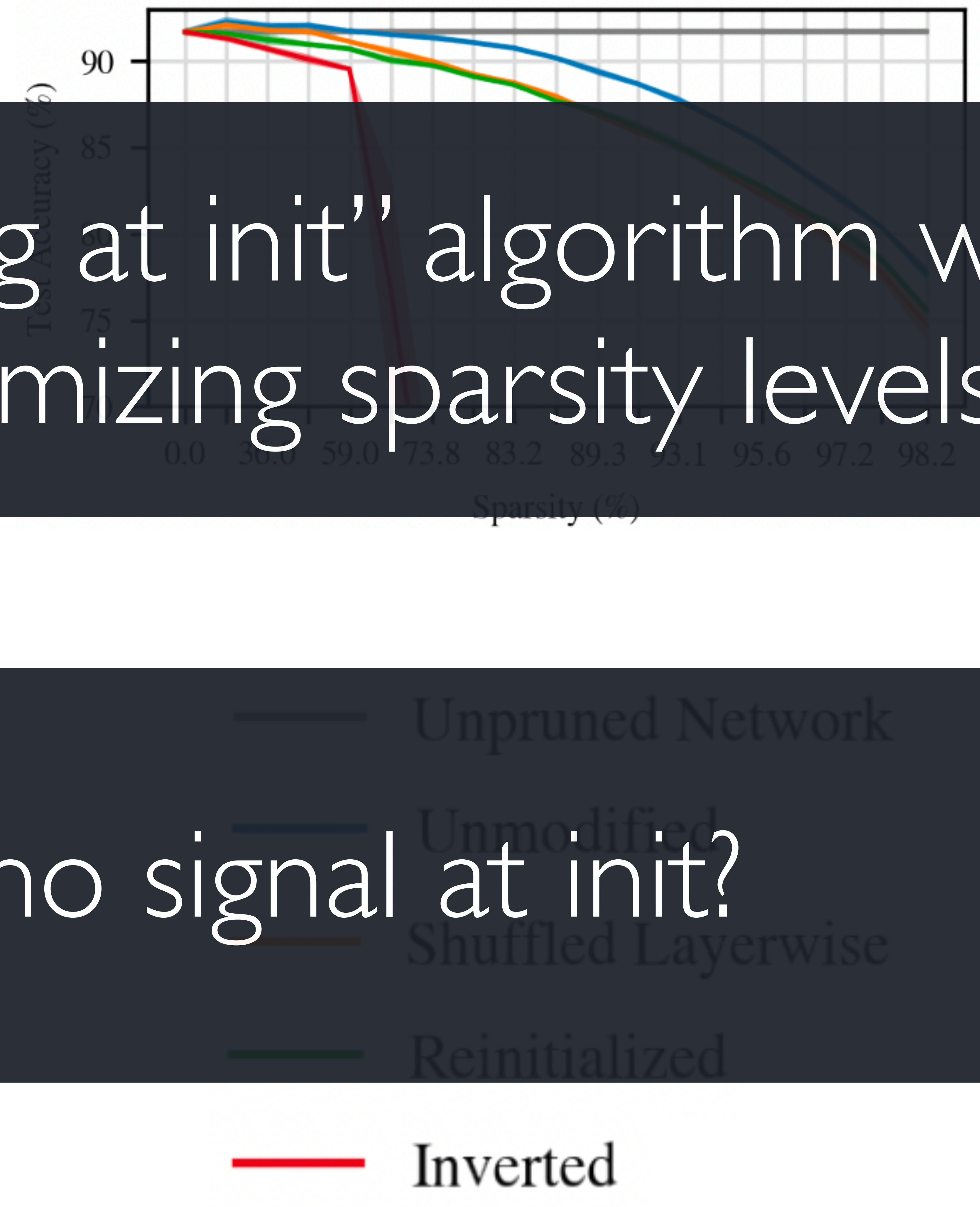
THE EARLY PHASE OF NEURAL NETWORK TRAINING

Jonathan Frankle, MIT CSAIL
David J. Schwab, CUNY ITS, Facebook AI Research
Ari S. Morcos, Facebook AI Research

Finding good tickets at init seems hard.
Current fix = wait for a few epochs

Sanity Checks [Frankle'21]





currently no "pruning at init" algorithm works better than optimizing sparsity levels

so there's no signal at init?

(ResNet-20, CIFAR-10)

Fixing IMP

fix = rewind shortly after init

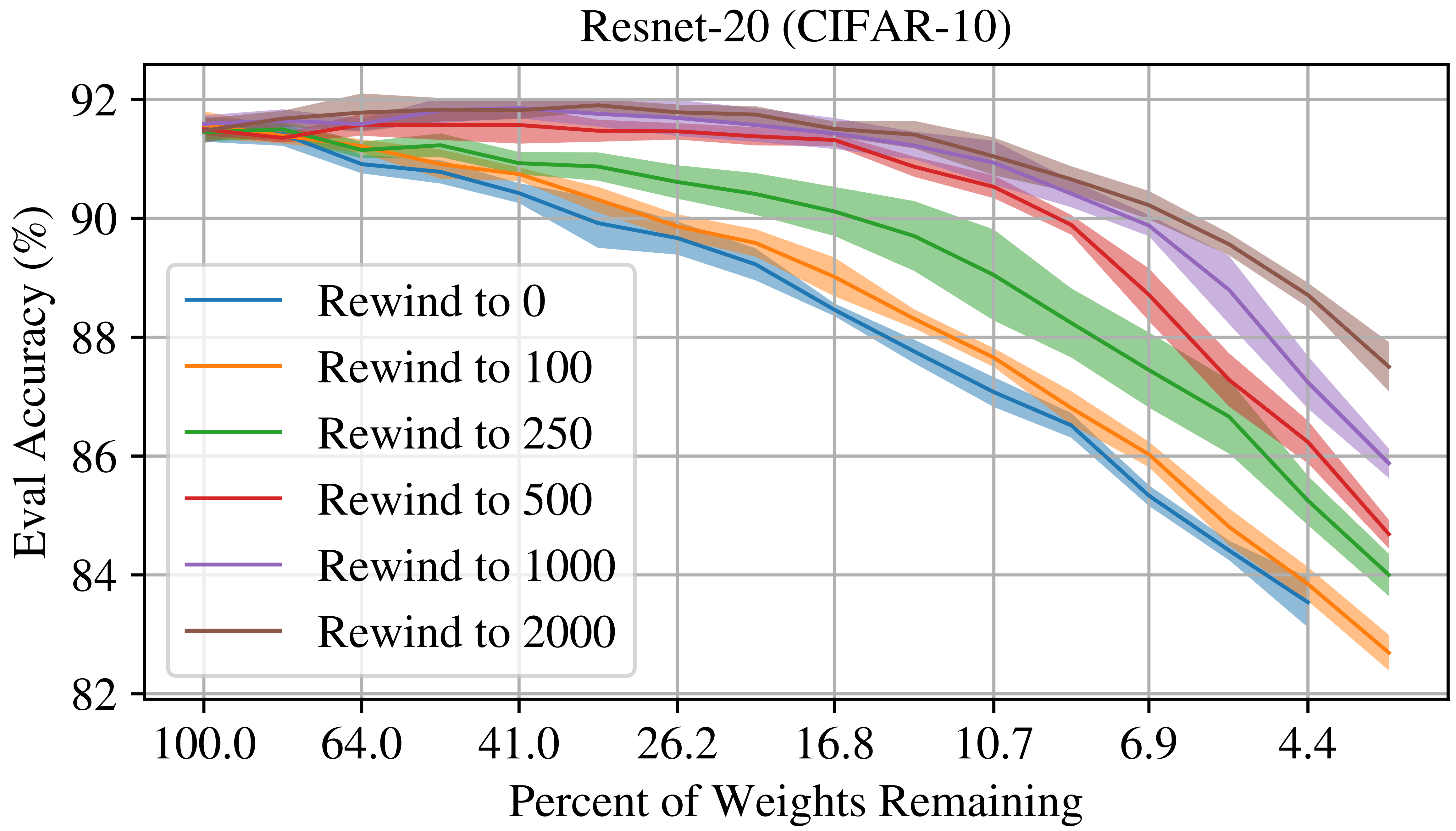
THE EARLY PHASE OF NEURAL NETWORK TRAINING

Jonathan Frankle†
MIT CSAIL

David J. Schwab
CUNY ITS
Facebook AI Research

Ari S. Morcos
Facebook AI Research

- “vanilla” LTH only true for MNIST/small nets
- Rewinding to init doesn’t work for Resnet/Cifar10
- One needs to rewind later (i.e., train a bit)



Lottery Tickets are hard to get at Init

Linear Mode Connectivity and the Lottery Ticket Hypothesis

Jonathan Frankle[1] Gintare Karolina Dziugaite[2] Daniel M. Roy[3 4] Michael Carbin[1]

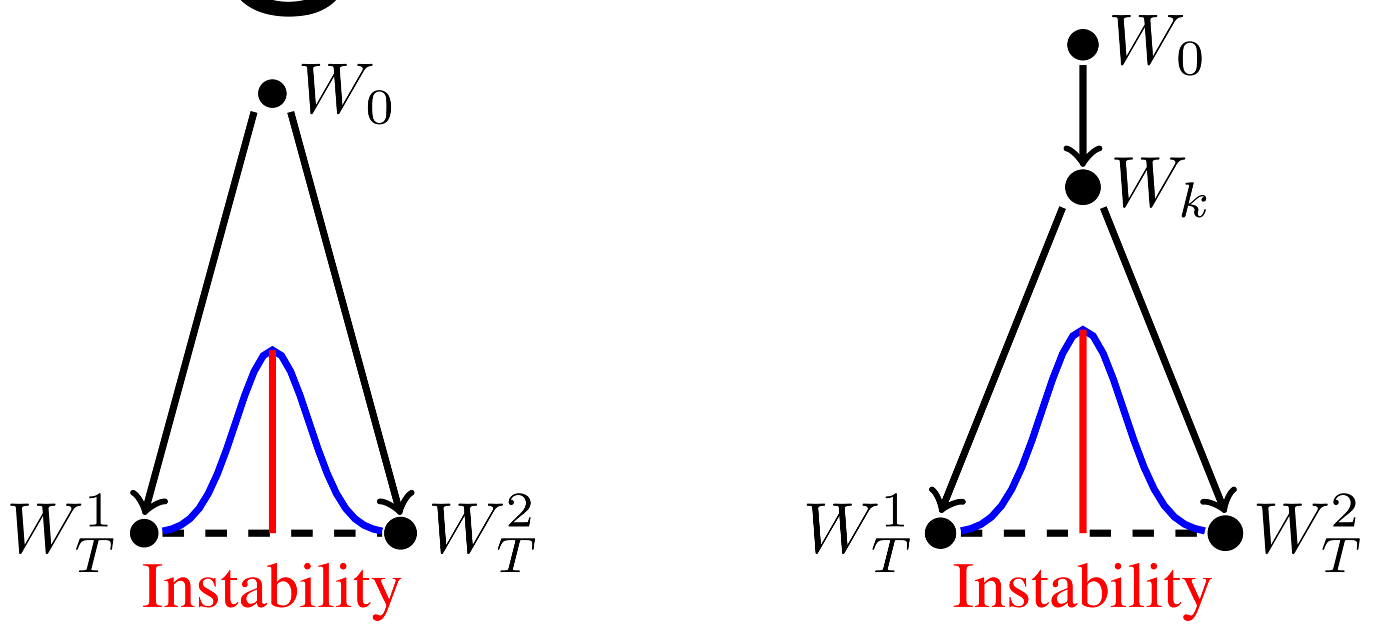
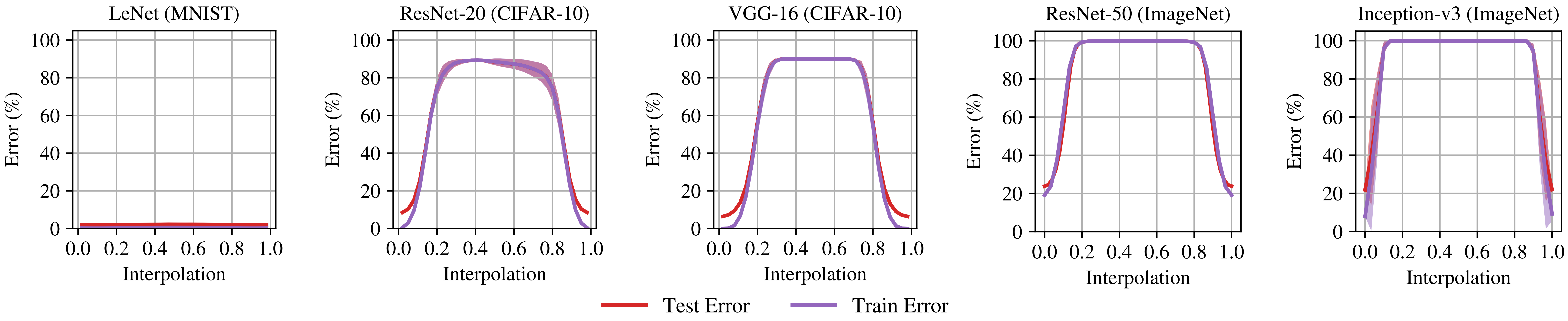


Figure 1. A diagram of instability analysis from step 0 (left) and step k (right) when comparing networks using linear interpolation.

- Rewinding to iteration K, rather than init works much better
- Experimental analysis through the existence of linear connectivity



Lottery Tickets are hard to get at Init

Linear Mode Connectivity and the Lottery Ticket Hypothesis

Jonathan Frankle [1] Gintare Karolina Dziugaite [2] Daniel M. Roy [3 4] Michael Carbin [1]

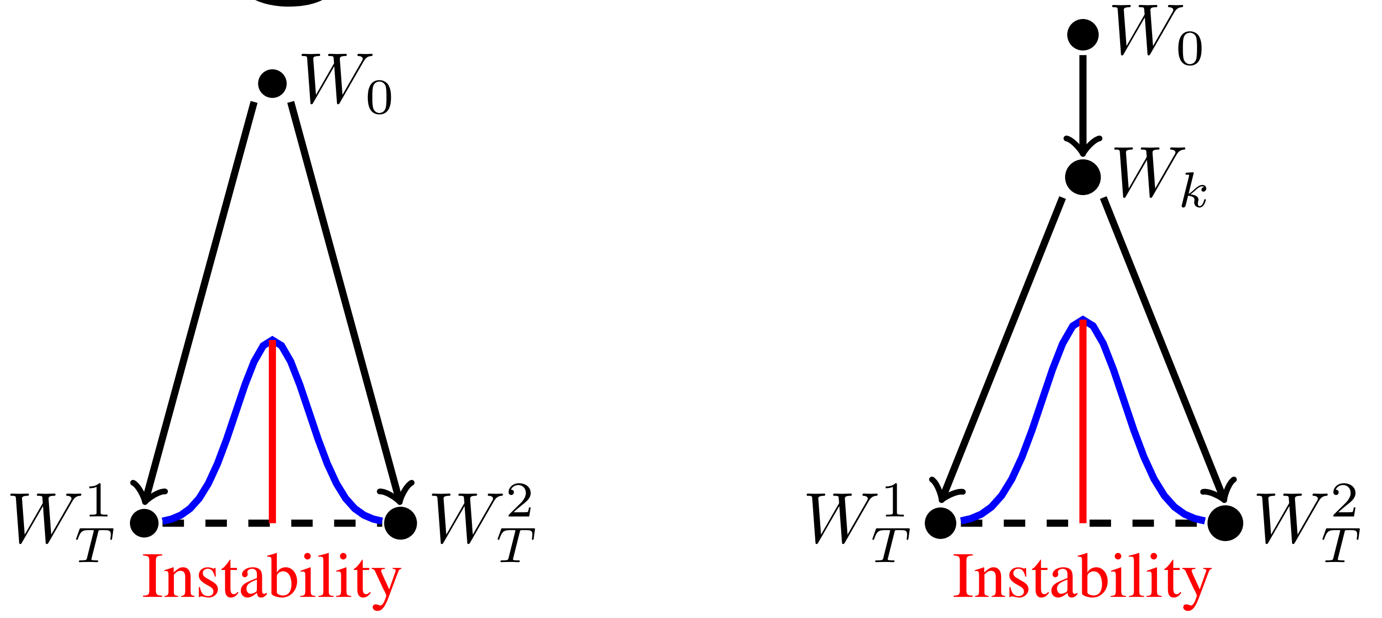
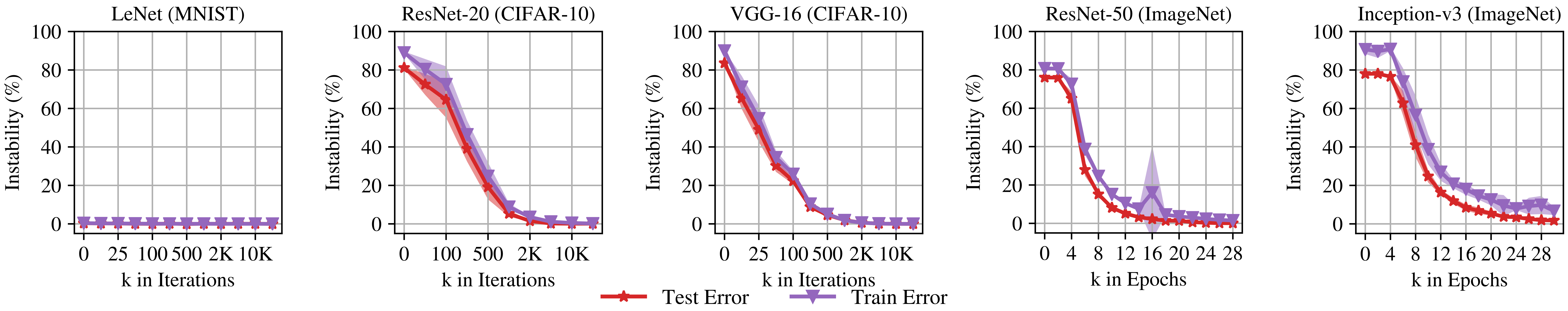


Figure 1. A diagram of instability analysis from step 0 (left) and step k (right) when comparing networks using linear interpolation.

- Rewinding to iteration K, rather than init works much better
- Experimental analysis through the existence of linear connectivity
- Connectivity emerges early in training, but not at init (hard to find models that exhibit it)



An interesting finding

The value of values

Deconstructing Lottery Tickets: Zeros, Signs, and the Supermask

Hattie Zhou
Uber
hattie@uber.com

Janice Lan
Uber AI
janlan@uber.com

Rosanne Liu
Uber AI
rosanne@uber.com

Jason Yosinski
Uber AI
yosinski@uber.com

Why do lottery tickets perform well?

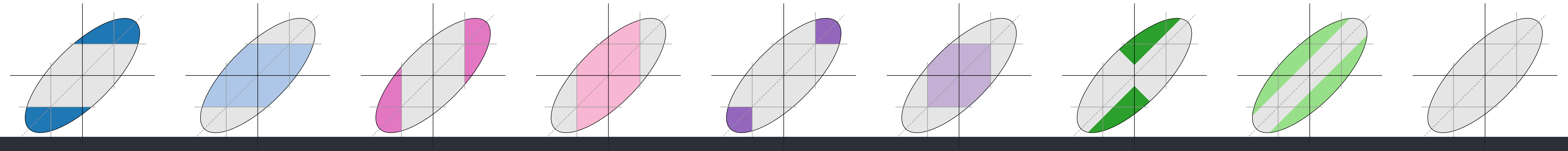
- *A Study on what allows LTs to be good*

Revisiting the IMP algorithm

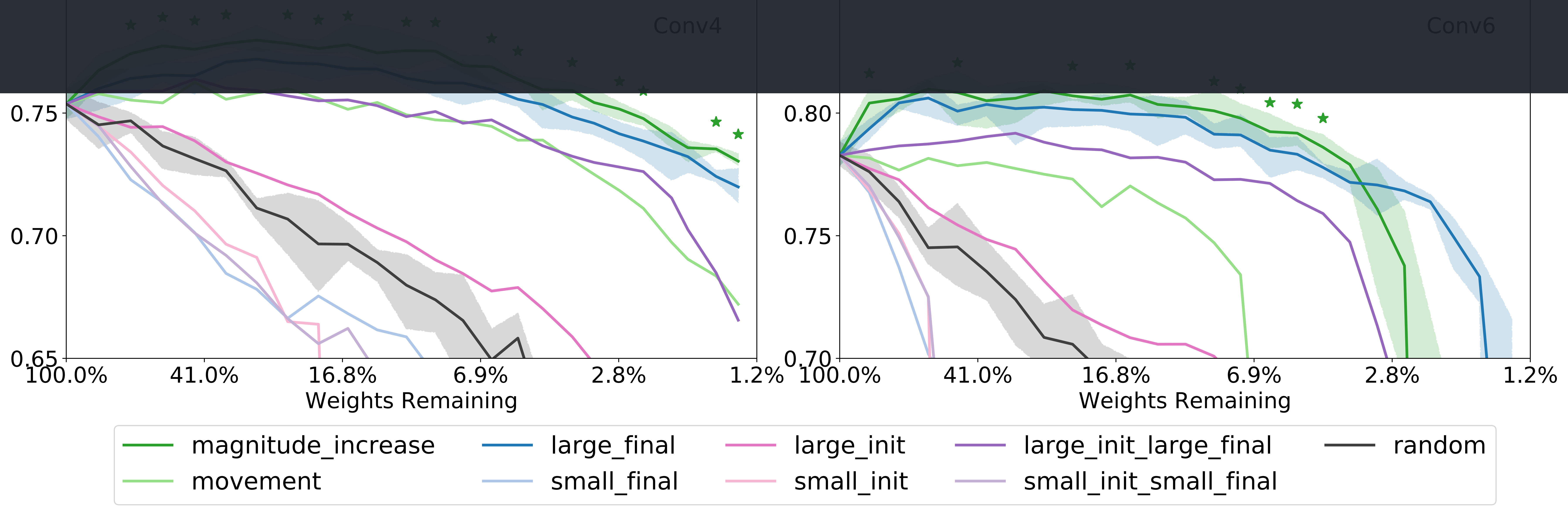
1. Randomly initialize weights
2. pick "importance" metric M — Mask criterion
3. Train for small number of iterations
4. Prune bottom p% of according to M — Mask-1 action
5. Rollback top 100-p% non-zero weights — Mask- 0 action
6. re-train to full accuracy
7. (Sometimes) GOTO 3

Mask Criteria

large final	small final	large init	small init	large init large final	small init small final	magnitude increase	movement	random
$\lvert w_f \rvert$	$-\lvert w_f \rvert$	$\lvert w_i \rvert$	$-\lvert w_i \rvert$	$min(\alpha \lvert w_f \rvert, \lvert w_i \rvert)$	$-max(\alpha \lvert w_f \rvert, \lvert w_i \rvert)$	$\lvert w_f \rvert - \lvert w_i \rvert$	$\lvert w_f - w_i \rvert$	0

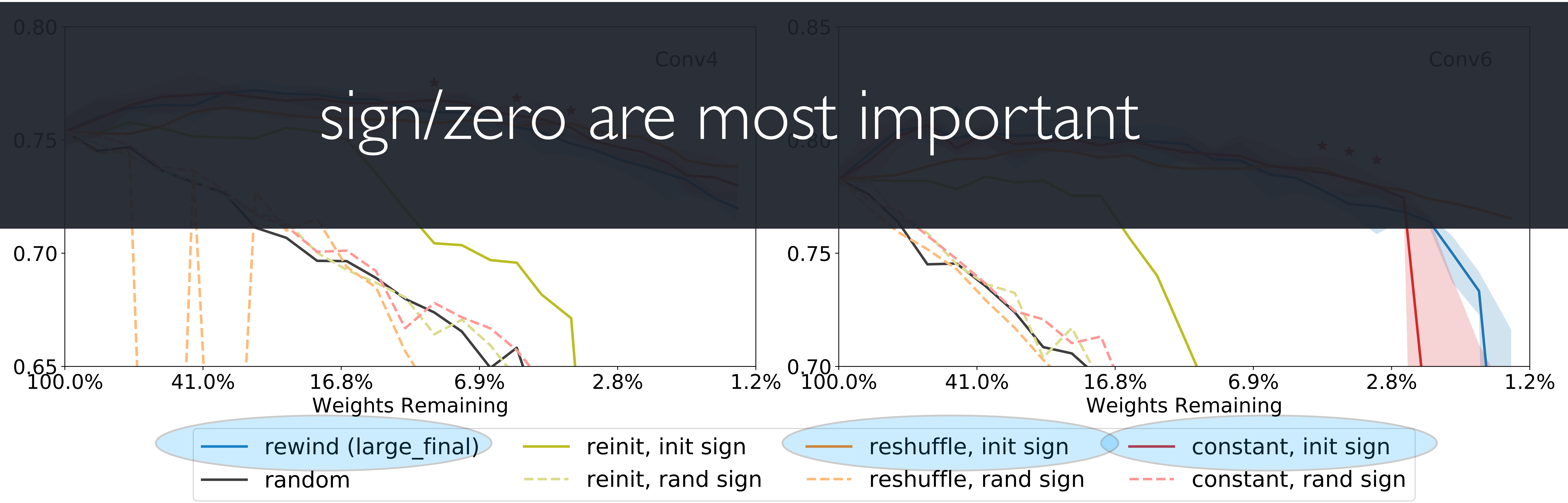


original LTH criterion pretty good



What to do with surviving weights?

- *Reinitialize*
- *Shuffle original values*
- *Replace with sign(w)*std(init. values)*

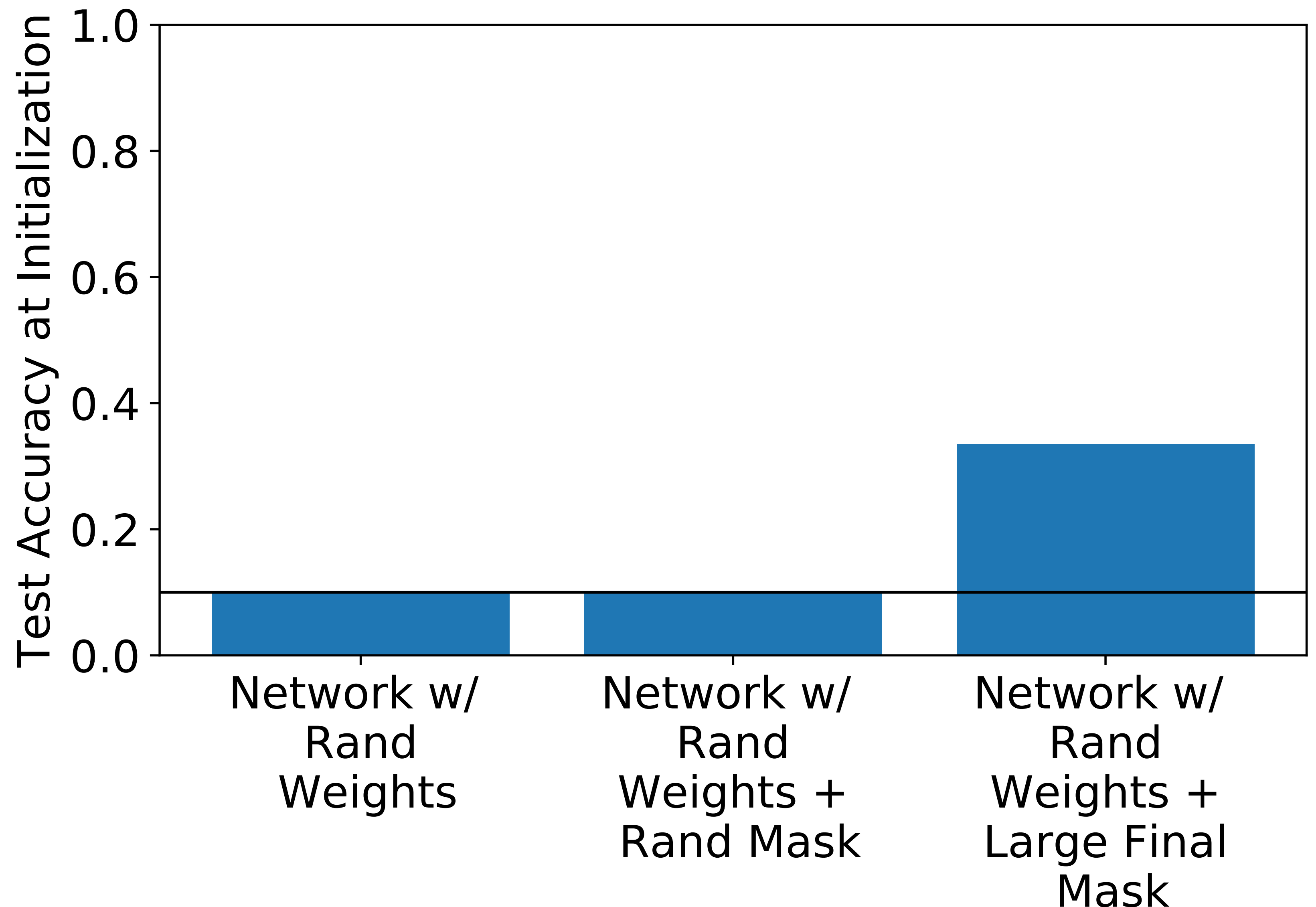


What to do with pruned weights?

- *freeze to init value*
- *set to zero*

zero is special:
learning the supermask similar to training

- *indication that pruning (without training) attains non-trivial test error*



Pruning is all you need??

What's Hidden in a Randomly Weighted Neural Network?

Vivek Ramanujan *† Mitchell Wortsman *‡ Aniruddha Kembhavi †‡
Ali Farhadi ‡ Mohammad Rastegari ‡

Abstract

Training a neural network is synonymous with learning the values of the weights. In contrast, we demonstrate that randomly weighted neural networks contain subnetworks which achieve impressive performance without ever modifying the weight values. Hidden in a randomly weighted Wide ResNet-50 [32] we find a subnetwork (with random weights) that is smaller than, but matches the performance of a ResNet-34 [9] trained on ImageNet [4]. Not only do these "untrained subnetworks" exist, but we provide an algorithm to effectively find them. We empirically show that as randomly weighted neural networks with fixed weights grow wider and deeper, an "untrained subnetwork" approaches a network with learned weights in accuracy. Our code and pretrained models are available at: https://github.com/allenai/hidden-networks.

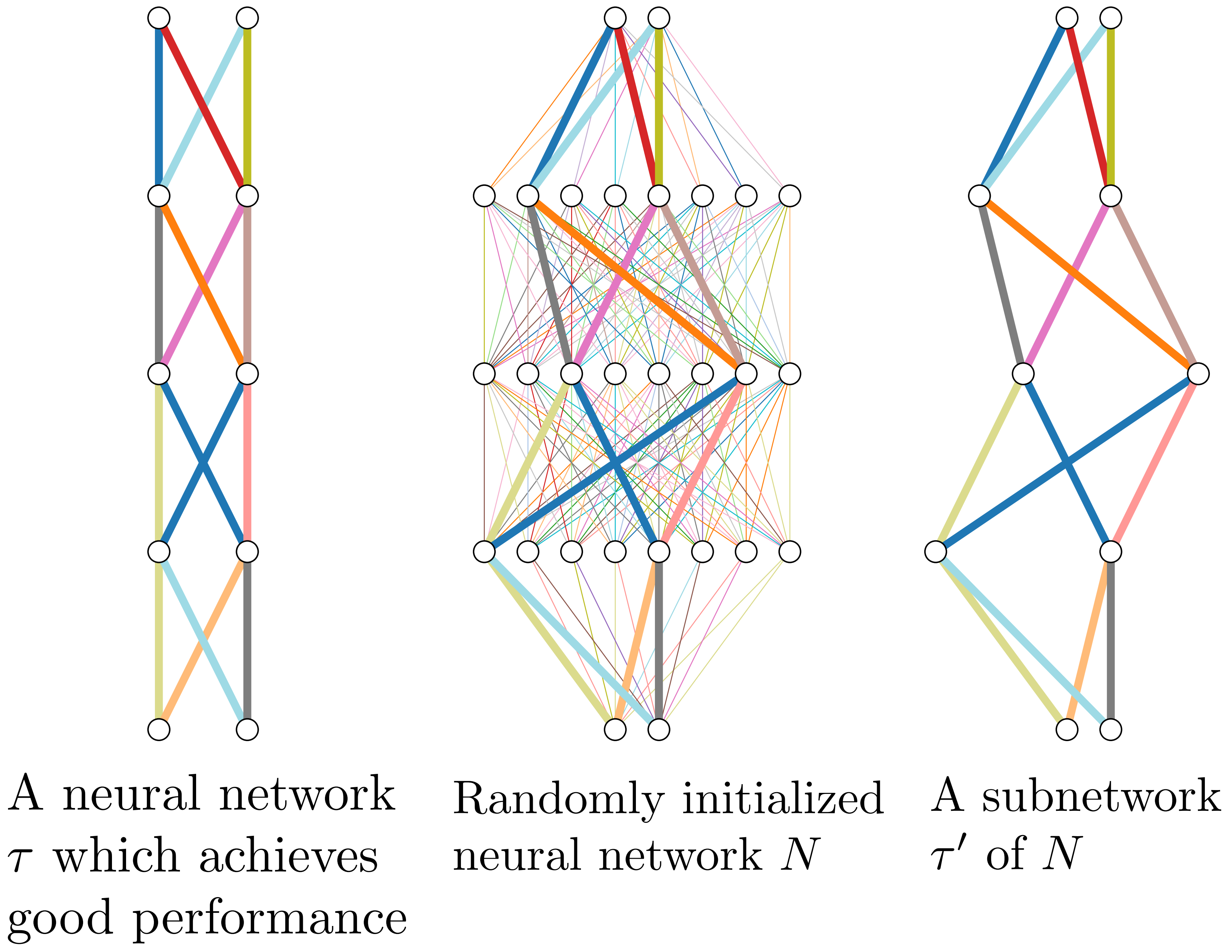
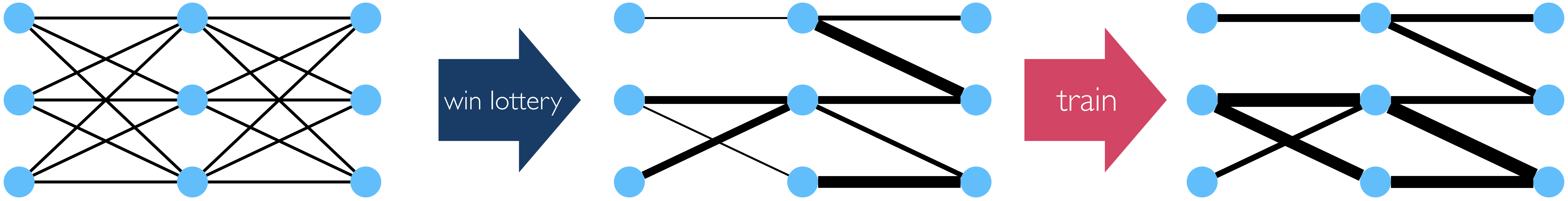


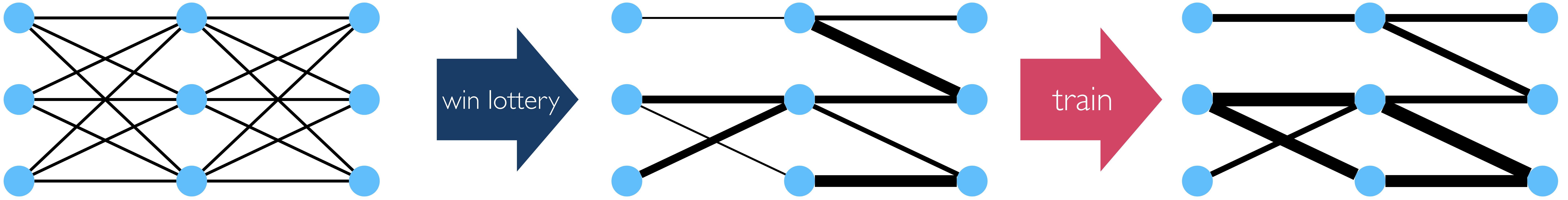
Figure 1. If a neural network with random weights (center) is sufficiently overparameterized, it will contain a subnetwork (right) that perform as well as a trained neural network (left) with the same number of parameters.

1. Introduction

LTH

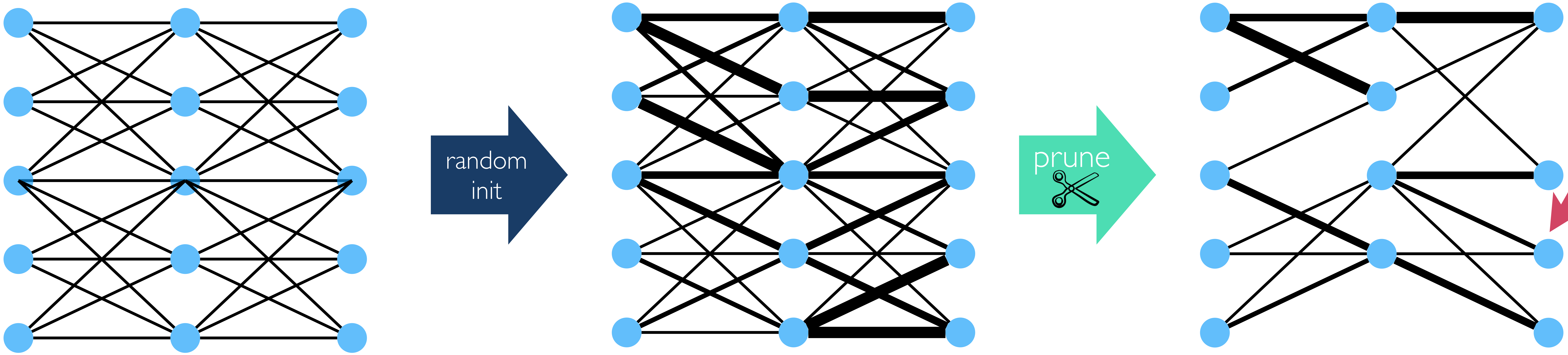


LTH



Strong LTH: **pruning is all you need**

SOTA accuracy tickets simply reside <u>within</u> random NNs



Woah, hold on…

You can get a high accuracy model…

WITHOUT SGD??

Q: Is the strong LTH universally true?

I.e., Can we always approximate a target NN by pruning a larger random network?

well, if the larger net contains all possible weights..

Conclusions & Open Problems

- “Trainable” sparse nets are desirable
- "early” *LTH = winning tickets exist at initialization*
- *later stage LTH = well, not quite, you have to train a bit first*
- *Finding LTs at init seems hard. Is it impossible though?*
- *Many extensions to BERT, Low-rank models, structured pruning*
- *Pruning is learning? WTFeta?*

Open Questions:

- Sparsity vs overparameterization
- Can we prune at initialization
- Where’s the math??

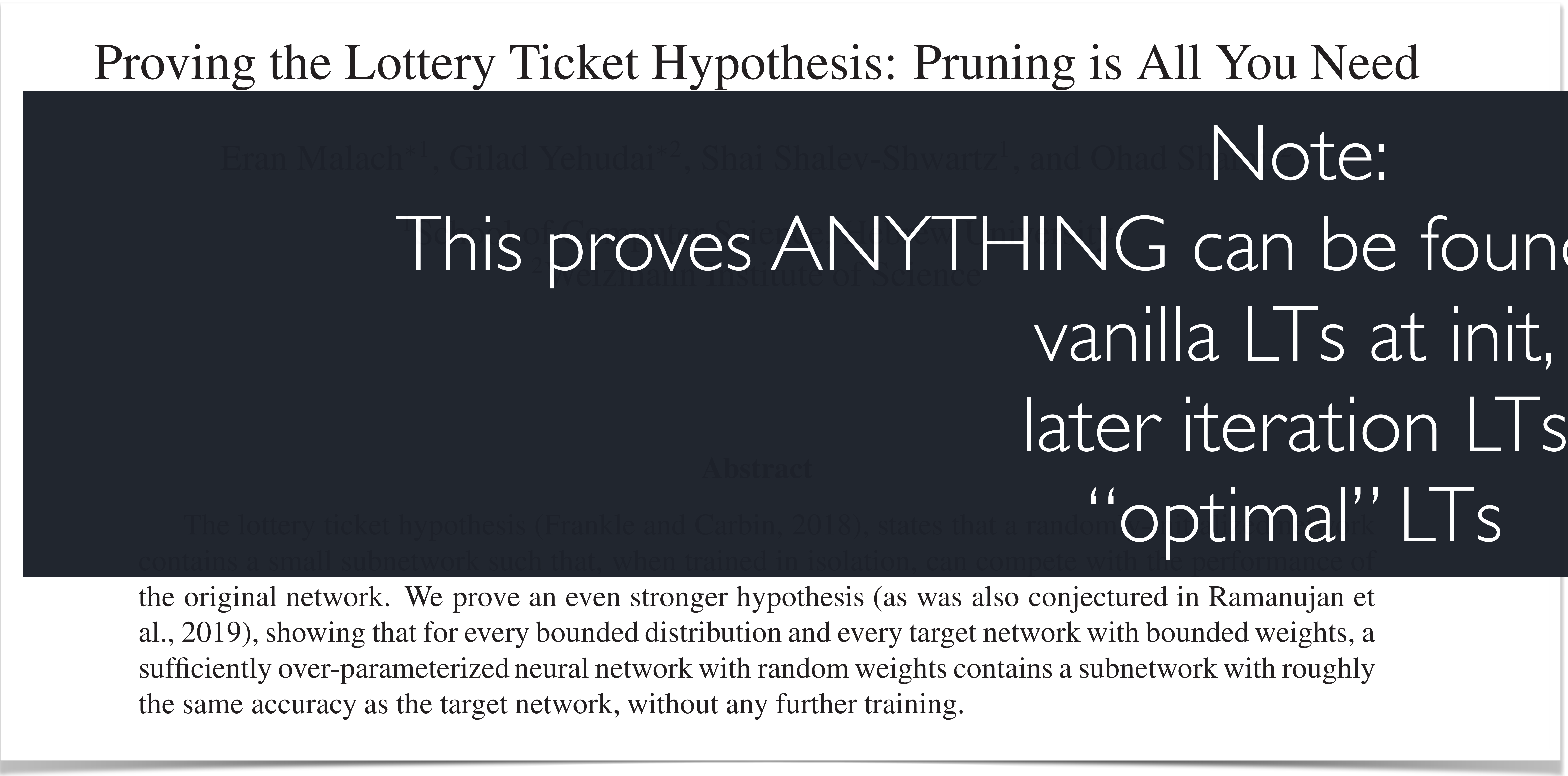
Part II: Theory

(mostly existential results)

Do lottery tickets exist?

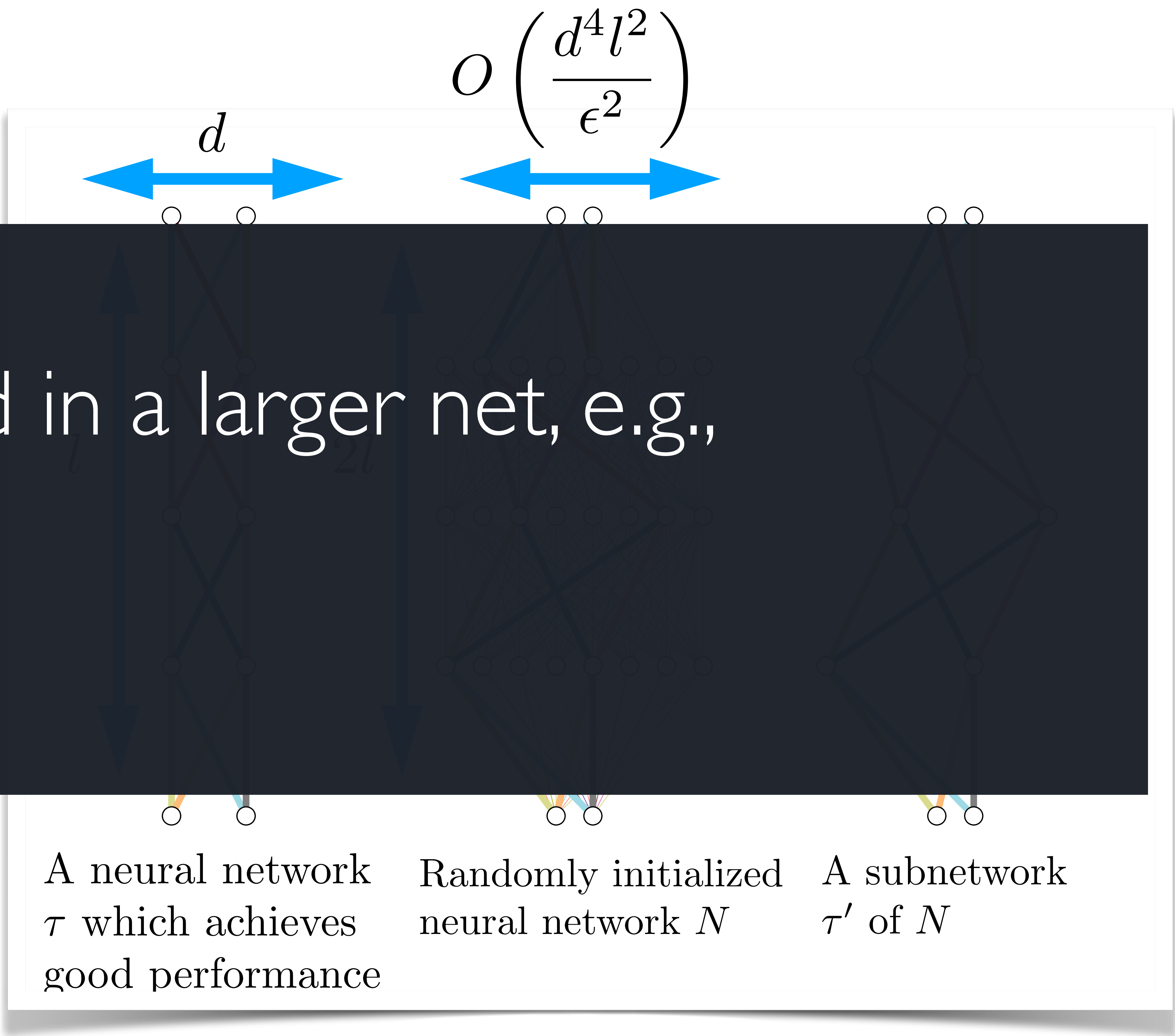
- Even in the absence of computational concerns, do LTs exist?
- If so, under what conditions?
- Provable poly-time algorithms?

Malach et al. Proving the Strong LTH



Proving the Lottery Ticket Hypothesis: Pruning is All You Need

the original network. We prove an even stronger hypothesis (as was also conjectured in Ramanujan et al., 2019), showing that for every bounded distribution and every target network with bounded weights, a sufficiently over-parameterized neural network with random weights contains a subnetwork with roughly the same accuracy as the target network, without any further training.



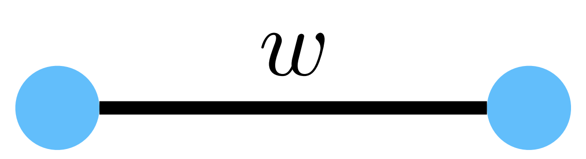
Note:
This proves ANYTHING can be found in a larger net, e.g.,
vanilla LTs at init,
later iteration LTs
"optimal" LTs

BUT… Ramanujan et al., prune a random WideResnet50
to approximate a Resnet 34

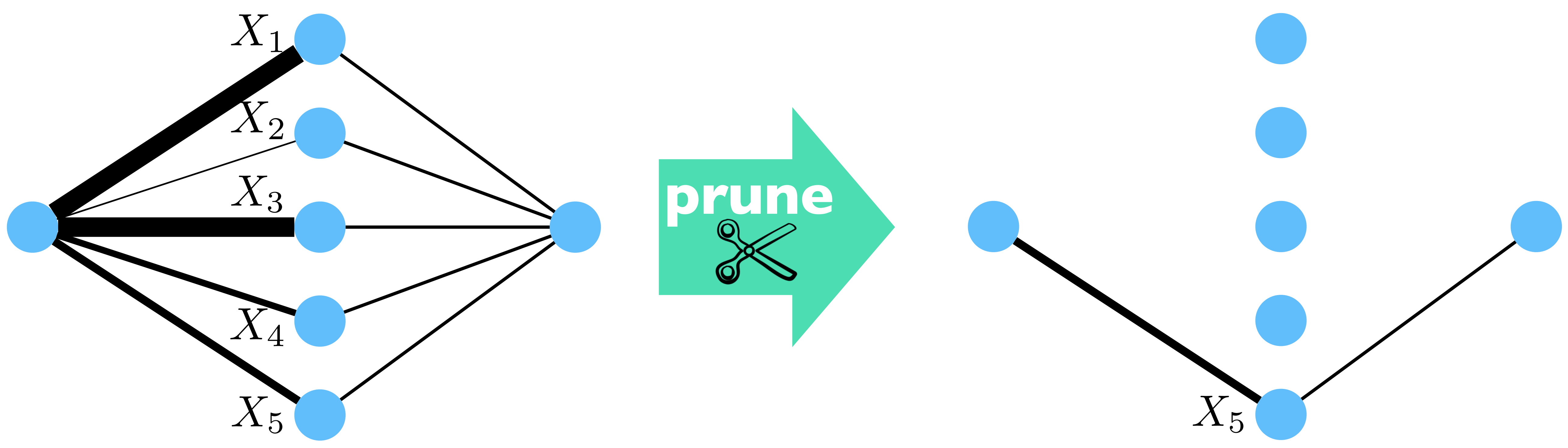
Sketch of Malach et al.

Main idea:
If there are enough weights
one can approximately find the target NN

target weight



pruning a highly over-parameterized network



$X_i \sim \text{Uniform R.V}$

Lemma: if we draw 1/ε random weights,
one will be ε-close to target with constant. probability

The general theorem is a more involved extension of this idea

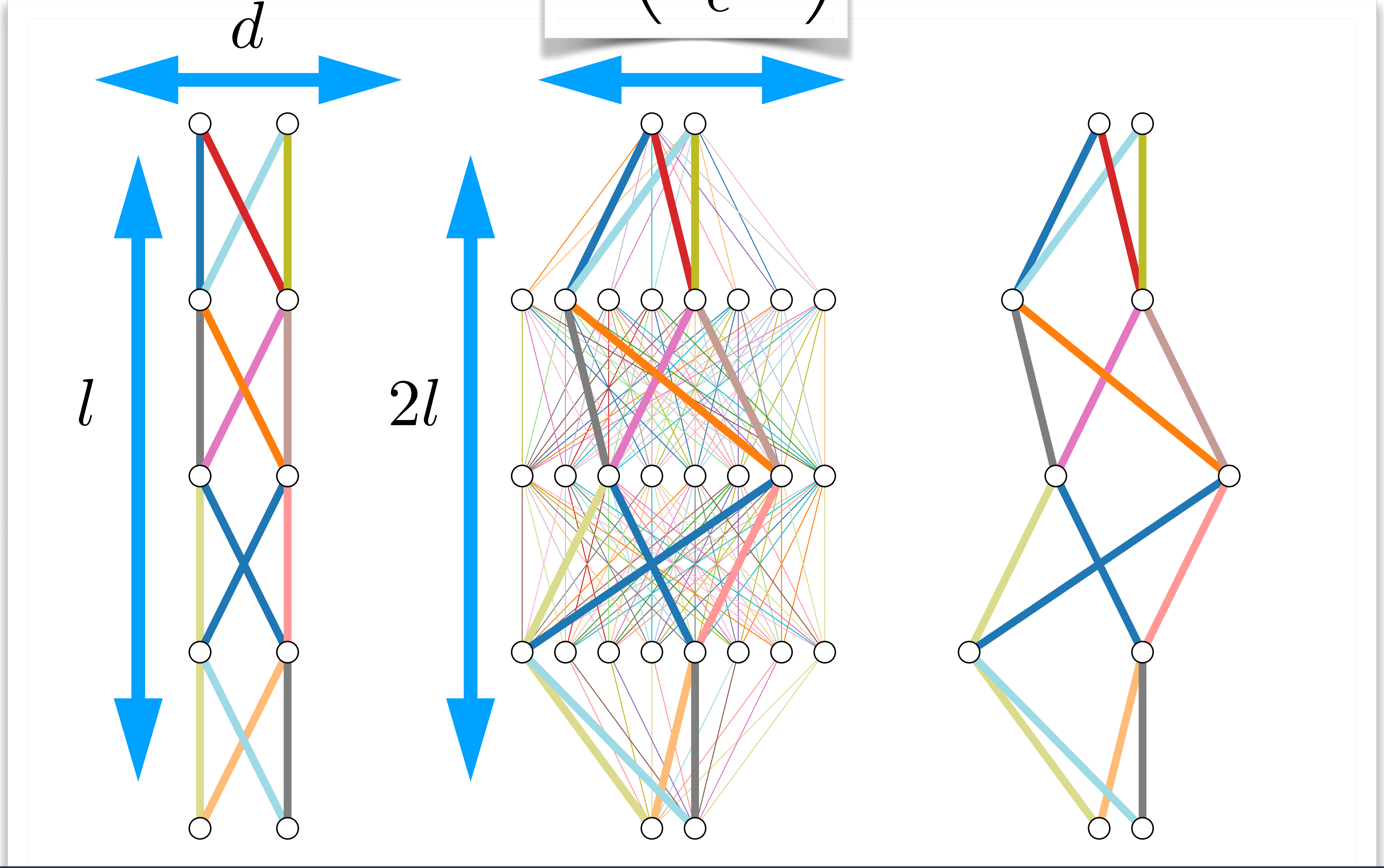
So we can't improve on this?

poly(1/ε) dependence unavoidable, if you prune down to one weight

Our work: Exponentially tighter Strong LTH

Malach et al.

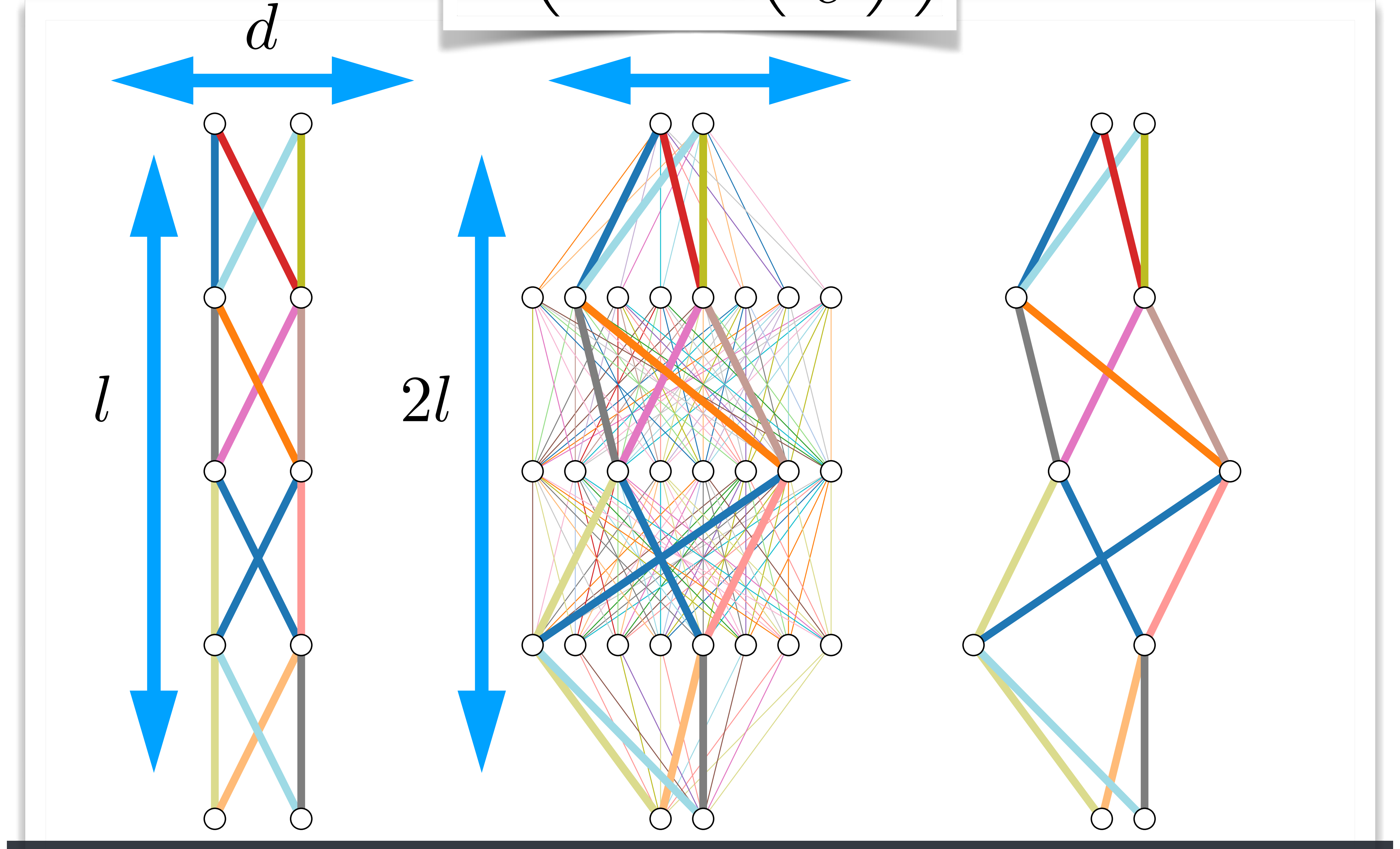
$$O\left(\frac{d^4 l^2}{\epsilon^2}\right)$$



winning a single weight lottery

Our work

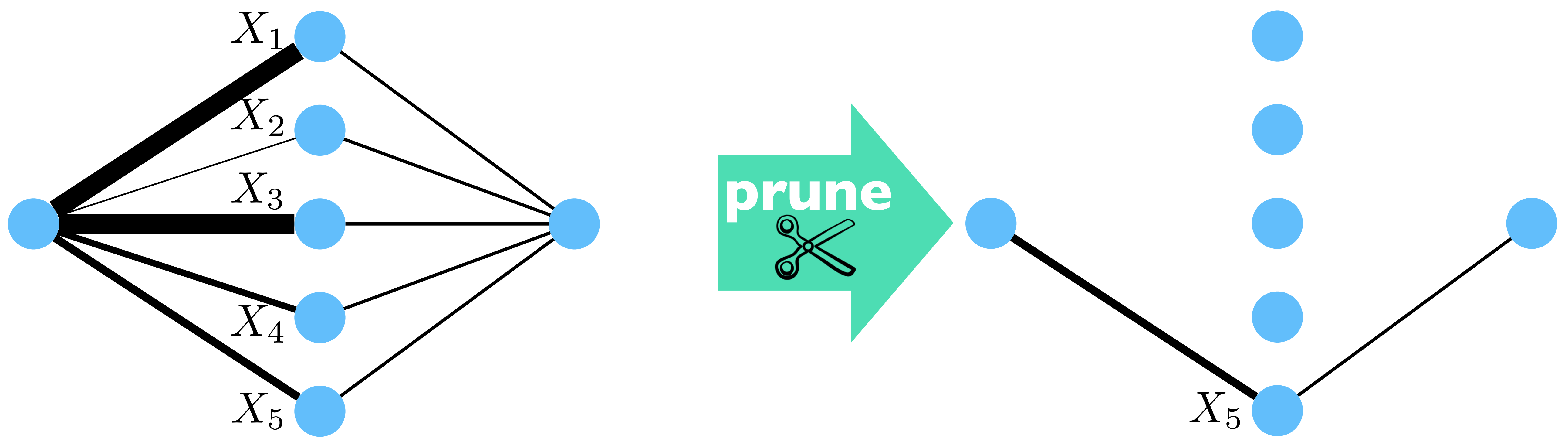
$$O\left(d \cdot \log\left(\frac{dl}{\epsilon}\right)\right)$$



winning a subset span lottery

The Subset Span approach

Malach theorem = pruning ε-nets



$X_i \sim \text{Uniform R.V}$

generating enough so that any number falls ε-close

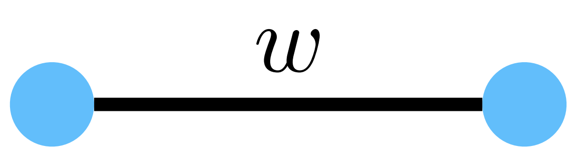
pruning = finding closest point to ε-net

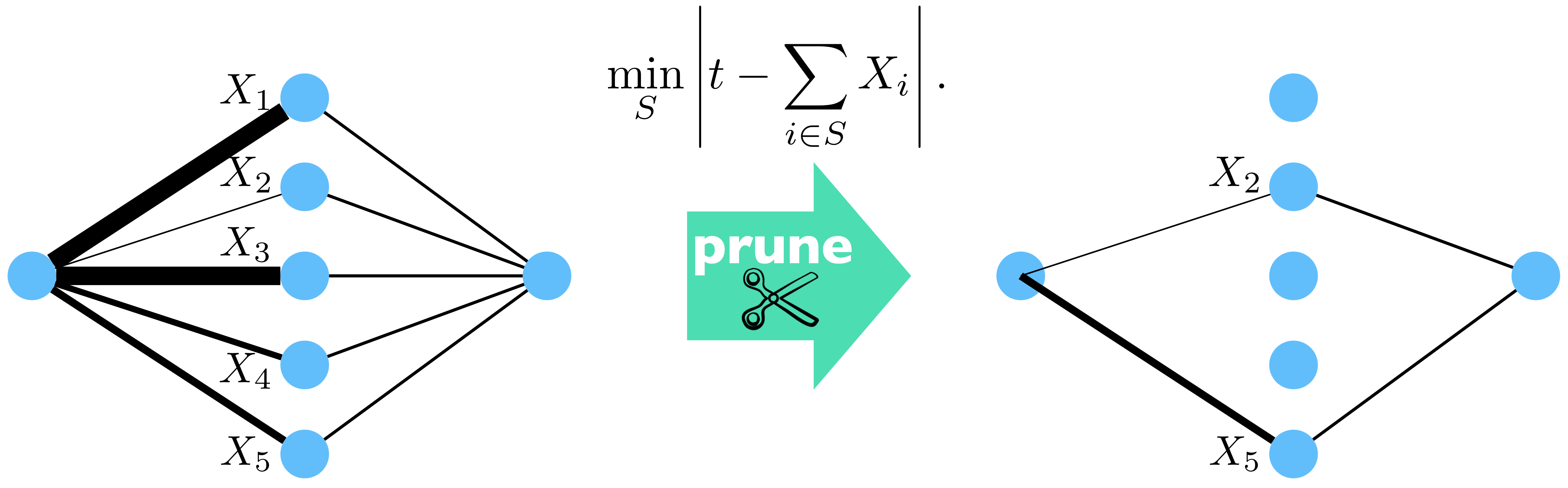
poly(1/ε) dependence unavoidable

what if we combine subsets of those random weights?

SUBSET Span

target weight





pruning = finding best subset sum to approximate target

Note: this is like a "batch" version of single parameter pruning

Q: how many RVs do I need for an ε-approximation?

[Lueker 1998]

Exponentially Small Bounds on the Expected Optimum of the Partition and Subset Sum Problems*

George S. Lueker
Department of Information and Computer Science
University of California, Irvine
Irvine, CA 92697-3425

Theorem 2.4. *Let $X_1, X_2, \ldots, X_n$ be i.i.d. uniform over $[-1, 1]$, and let $0 < \eta < \frac{1}{2}$. Suppose that $n/2 \geq C \ln \eta^{-1}$. Then, except with probability bounded by*

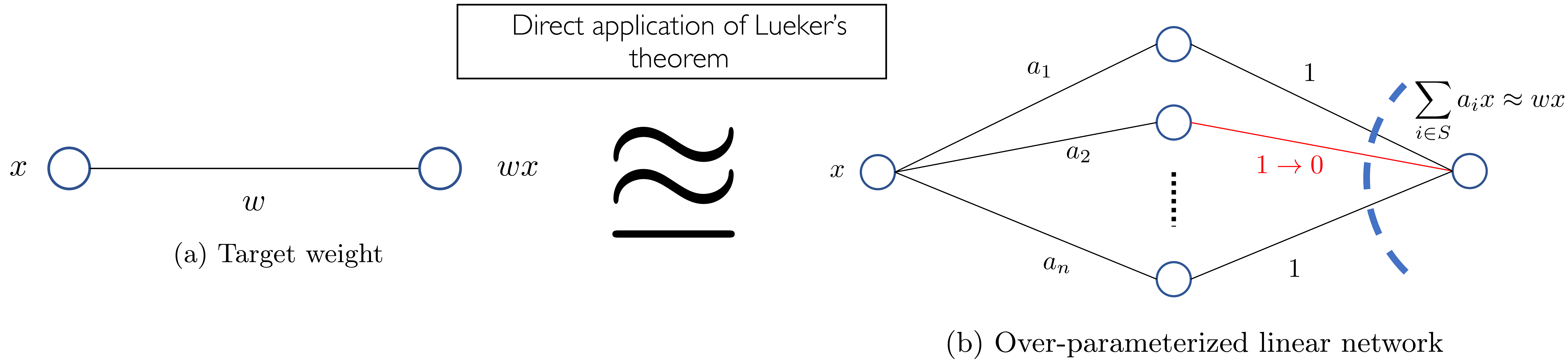
$$\exp\left(-\frac{\left(n/2 - C \ln \eta^{-1}\right)^2}{2n}\right),$$

all values in $[-\frac{1}{2}, \frac{1}{2}]$ have admissible 2η-approximations.

The Subset span is very expressive:

Every number in [-1,1], can be approximated by taking a subset of *log*(1/ε) RVs

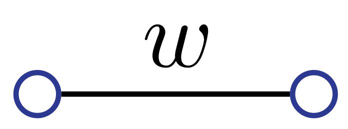
How to approximate a single weight



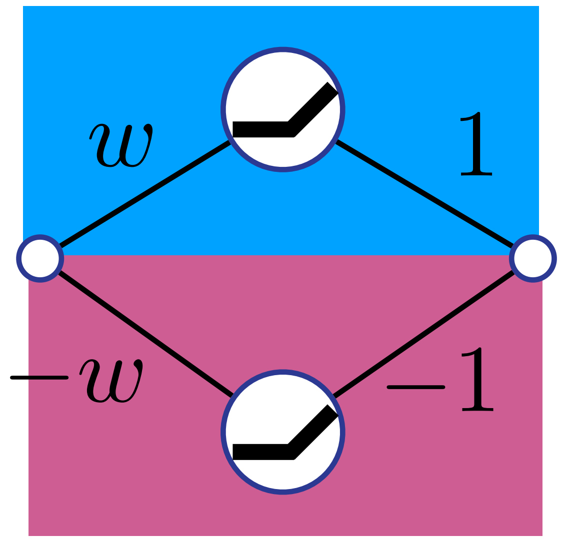
How do we transform the linear net to a ReLu?
<u>Constraint</u>: Weights have to be uniform and iid.

How to approximate a single weight

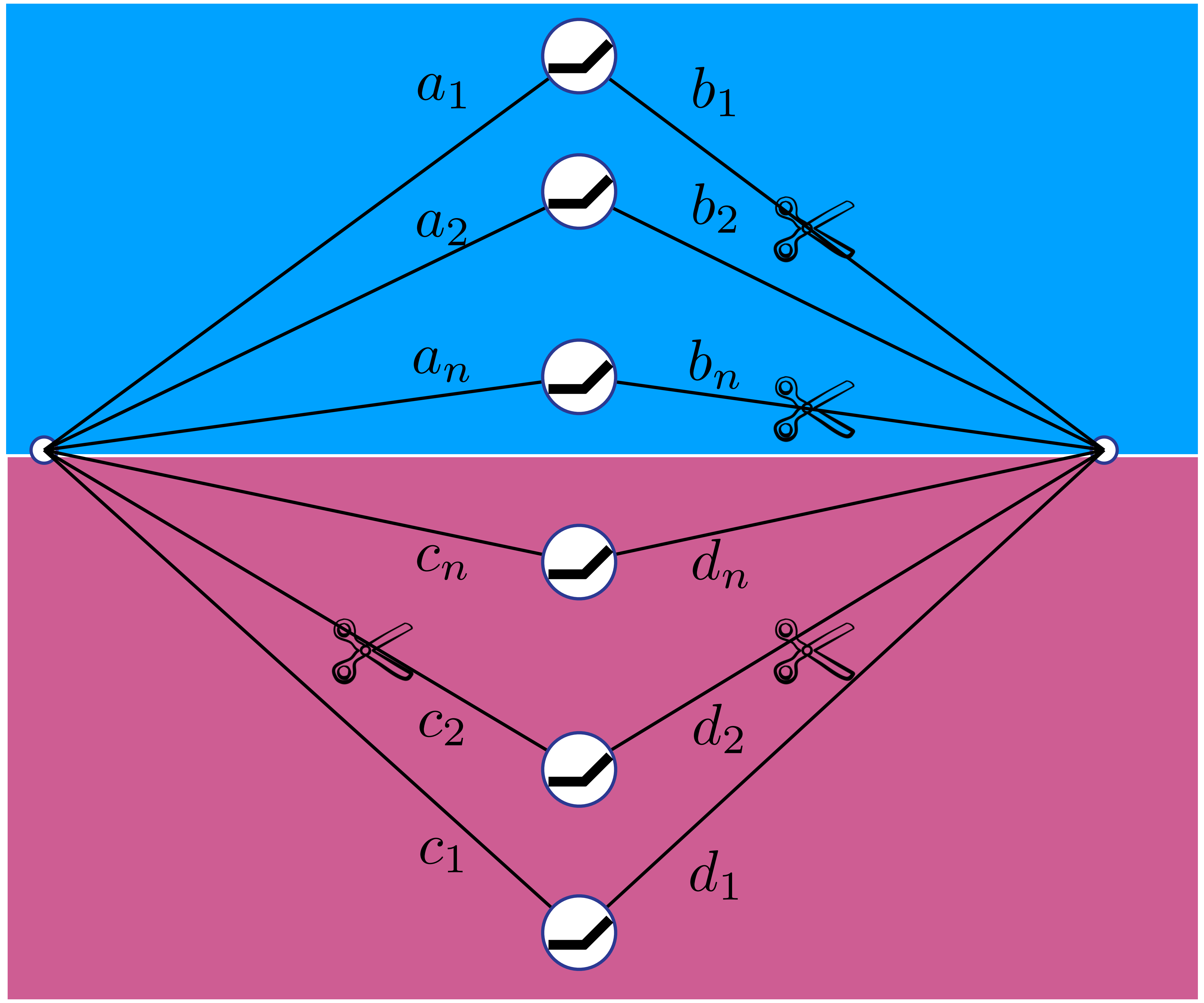
$: \sigma(x) = x \cdot \mathbf{1}_{x \geq 0}$



$=$



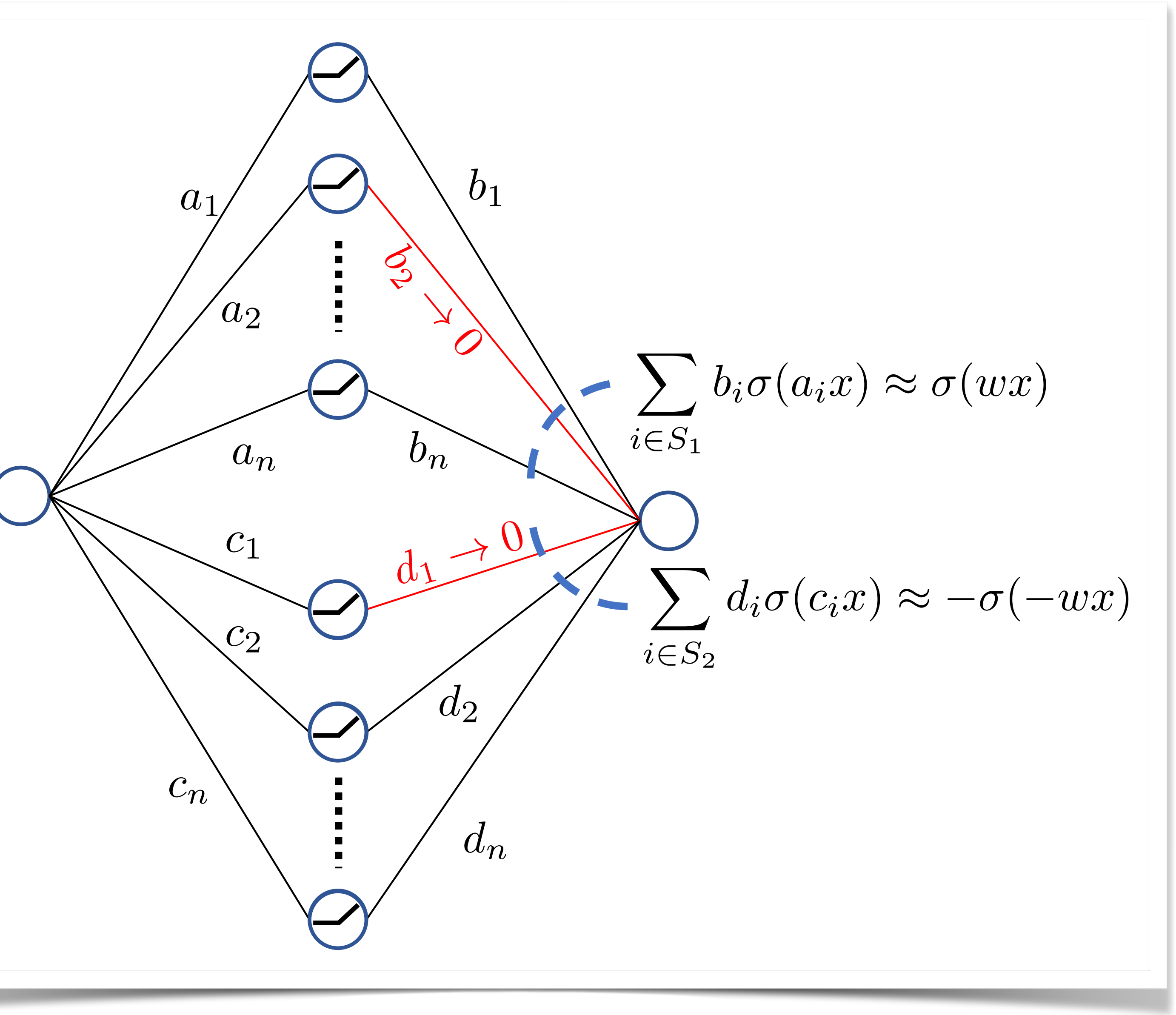
$\approx$

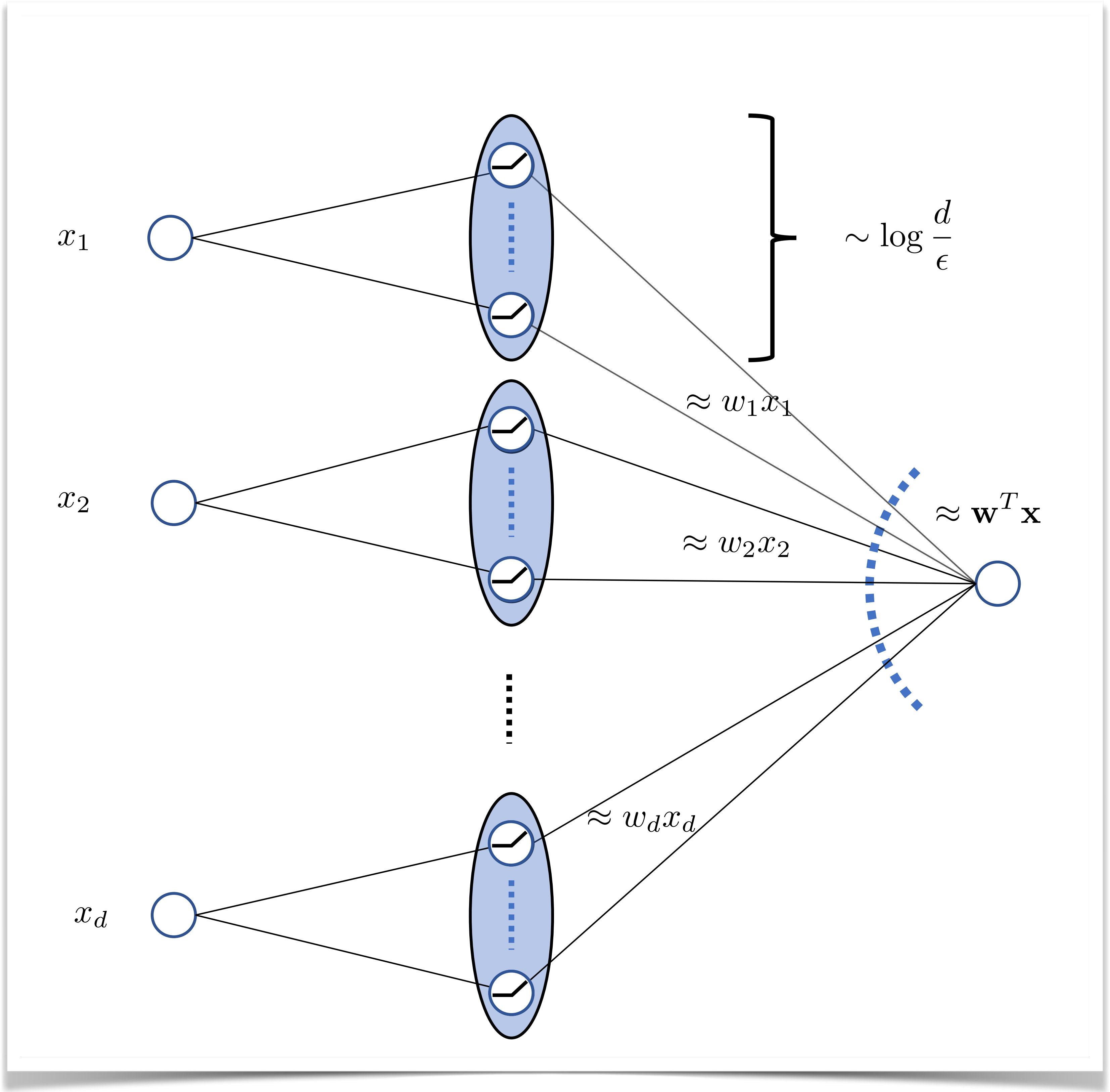


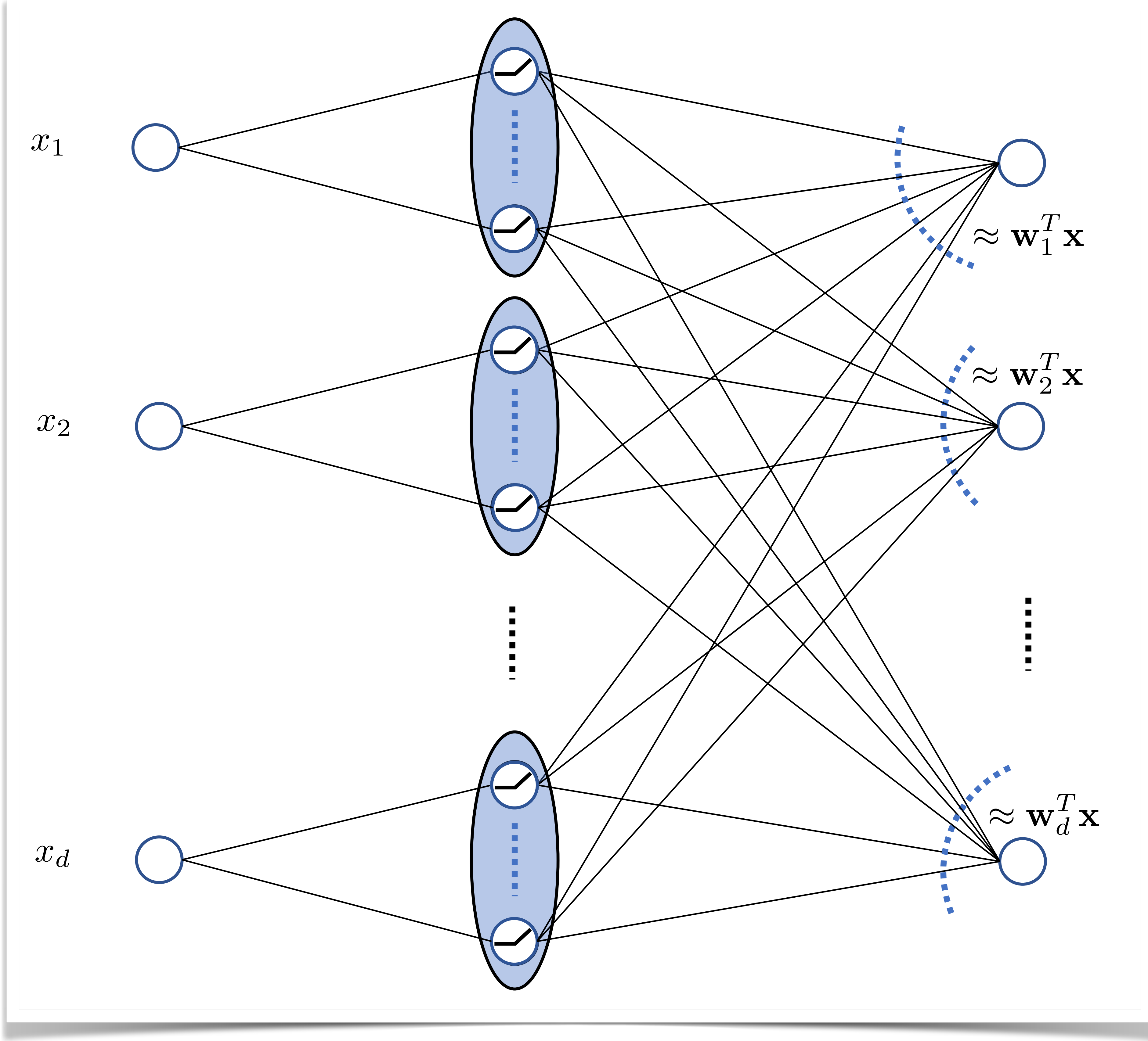
Lueker's theorem still holds if the distribution of a*b contain Uniform[-1,1]

from weights to neurons to networks

$w \cdot x$	$\mathbf{w}^T\mathbf{x}$	$\mathbf{W}^T\mathbf{x}$





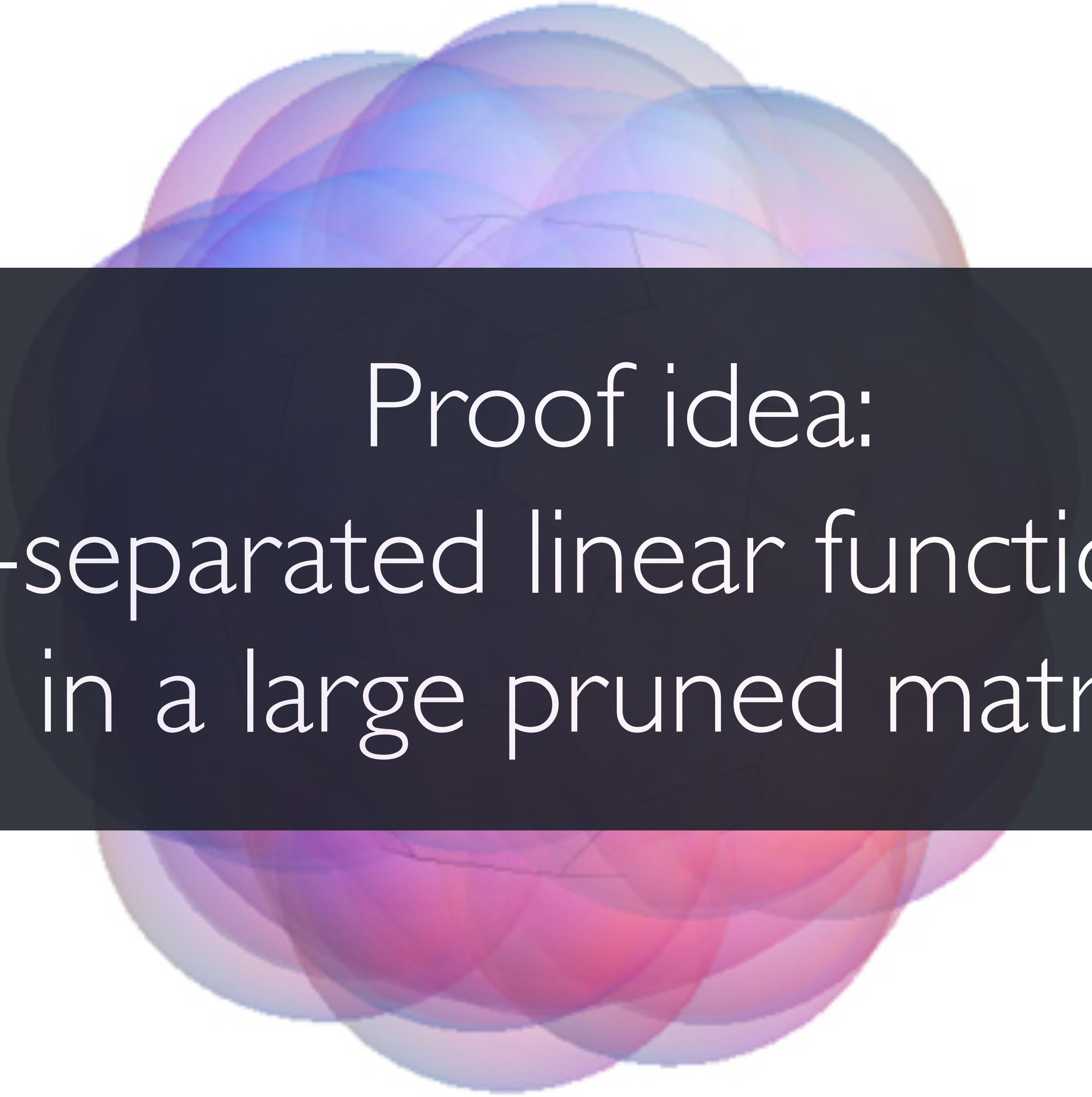


All that is left: operator norm bounds for each approximation layer so that

$$\min_{\mathbf{S}_i \in \{0,1\}^{d_i \times d_{i-1}}} \sup_{\|\mathbf{x}\| \le 1} \|f(\mathbf{x}) - (\mathbf{S}_{2l} \odot \mathbf{M}_{2l})\sigma((\mathbf{S}_{2l-1} \odot \mathbf{M}_{2l-1}) \ldots \sigma((\mathbf{S}_1 \odot \mathbf{M}_1)\mathbf{x})\| < \epsilon.$$

Lower bound via Packing

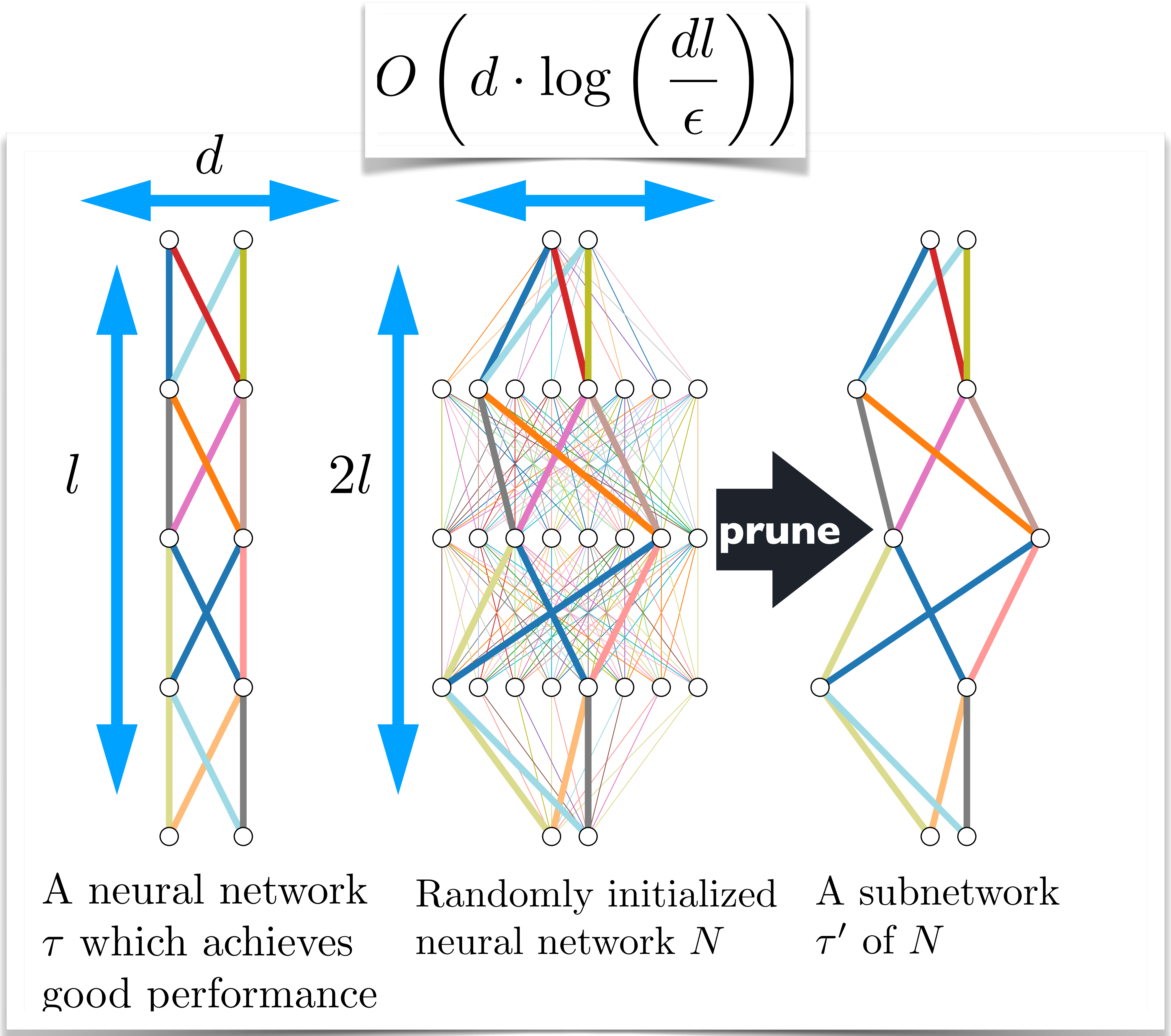
Theorem 2. *(informal) There exists a 2-layer neural network with width d which cannot be approximated to error within ϵ by pruning a randomly initialized 2-layer network, unless the random network has width at least $\Omega(d \log(1/\epsilon))$.*



Proof idea:
How many ε-separated linear functions can we pack
in a large pruned matrix?

Learning ⊂ Pruning

Any neural network be approximated by pruning
a logarithmically overparameterized network of random* weights

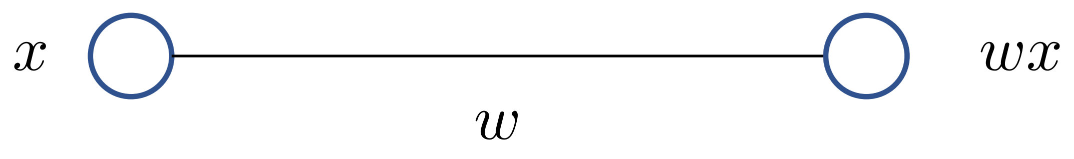


Note:
this is an existential result. Although our proof is algorithmic, we do not propose a new pruning algorithm

one experiment

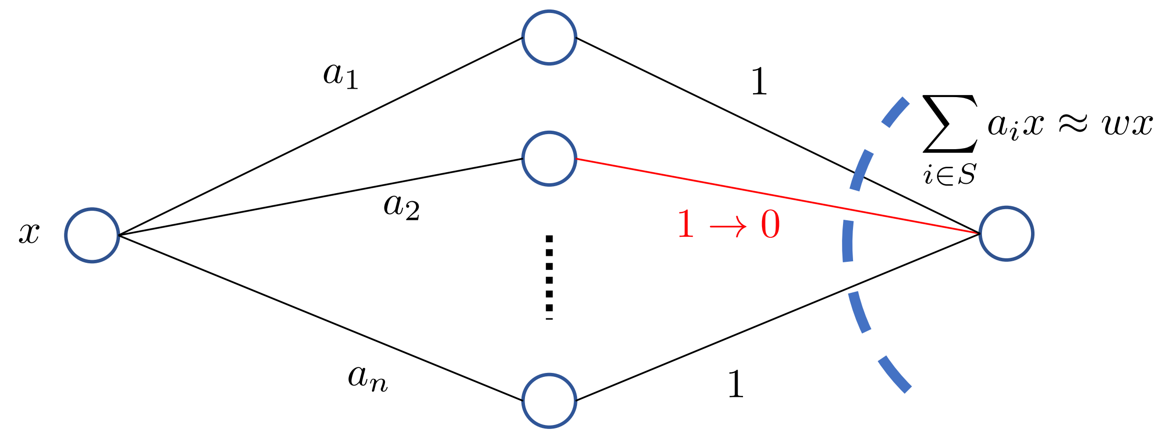
Type of Overparam Matters a lot!!

- Comparison for pruning wider nets
  - FC ReLU nets



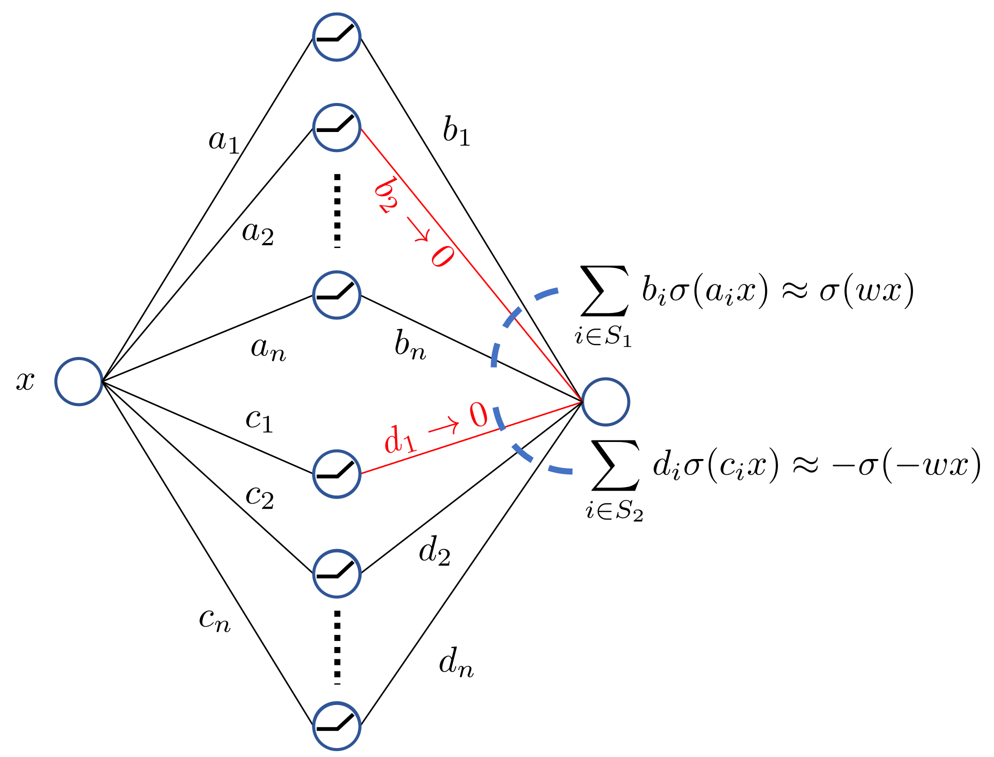
(a) Target weight

  - Nets with linear "diamond" structure



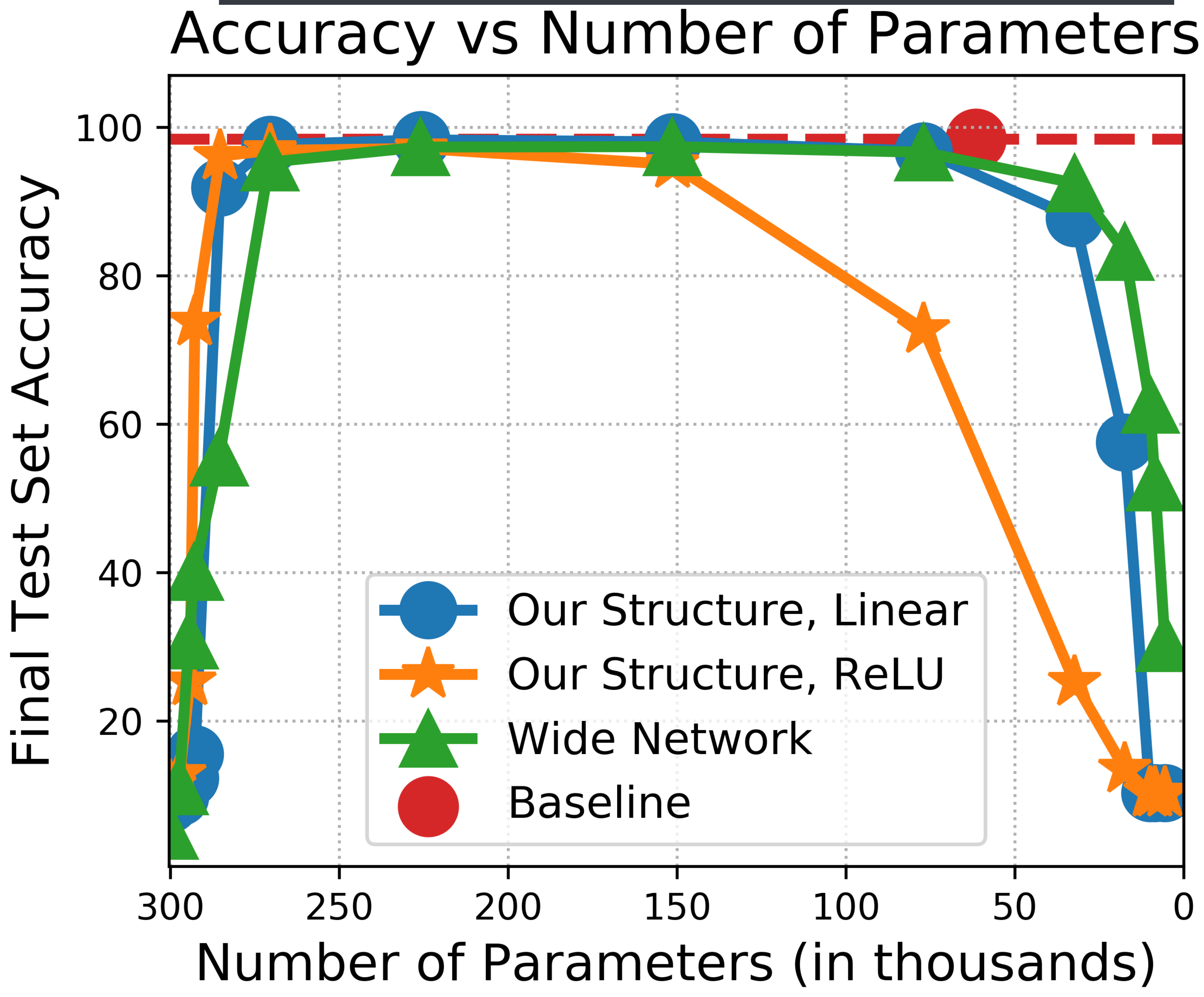
(b) Over-parameterized linear network

  - Nets with ReLU "diamond" structure



(c) Over-parameterized ReLU network

Pruning via the Ramanujan et al. algorithm



(c) LeNet5

Conclusions & Open Problems

- A DlogD random net contains ALL networks of size D!
- Vanilla LTs exist! So do Perfect LTs!
- The IMP's problem is not existence, but algorithmic.
- One can learn by pruning

Open Question:

- Can we fix IMP?
- Prune + train existential results?
- Can pruning be faster than training? (better for hardware?)
- Network architectures amenable to pruning
- Towards a "no-backprop" training framework

Reading List

Han, S., Mao, H. and Dally, W.J., 2015. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. arXiv preprint arXiv:1510.00149.

Frankle, J. and Carbin, M., 2018. The lottery ticket hypothesis: Finding sparse, trainable neural networks. ICLR 2019

Malach, E., Yehudai, G., Shalev-Schwartz, S. and Shamir, O., 2020, November. Proving the lottery ticket hypothesis: Pruning is all you need. In International Conference on Machine Learning (pp. 6682-6691). PMLR.

Frankle, J., Dziugaite, G.K., Roy, D. and Carbin, M., 2020, November. Linear mode connectivity and the lottery ticket hypothesis. In International Conference on Machine Learning (pp. 3259-3269). PMLR.

Zhou, H., Lan, J., Liu, R. and Yosinski, J., 2019. Deconstructing lottery tickets: Zeros, signs, and the supermask. Advances in neural information processing systems, 32.

Liu, Z., Sun, M., Zhou, T., Huang, G. and Darrell, T., 2018. Rethinking the value of network pruning. arXiv preprint arXiv:1810.05270.

Gale, T., Elsen, E. and Hooker, S., 2019. The state of sparsity in deep neural networks. arXiv preprint arXiv:1902.09574.

Ramanujan, V., Wortsman, M., Kembhavi, A., Farhadi, A. and Rastegari, M., 2020. What's hidden in a randomly weighted neural network?. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 11893-11902).

Pensia, A., Rajput, S., Nagle, A., Vishwakarma, H. and Papailiopoulos, D., 2020. Optimal lottery tickets via subset sum: Logarithmic over-parameterization is sufficient. Advances in Neural Information Processing Systems, 33, pp.2599-2610.