The Lottery Ticket Hypothesis

Shichang Zhang

04 19 2022

Roadmap

- The Lottery Ticket Hypothesis (LTH)
  - The pruning problem
  - The original LTH
  - Linear mode connectivity and instability analysis (helps the LTH to scale up)
  - Early-bird tickets (helps the LTH to become more practical)
- LTH and Graphs
  - A Unified Lottery Ticket Hypothesis for Graph Neural Networks (ICML 2021)

The Pruning Problem

The Pruning Problem

- Overparameterized NNs are less efficient and hard to deploy

The Pruning Problem

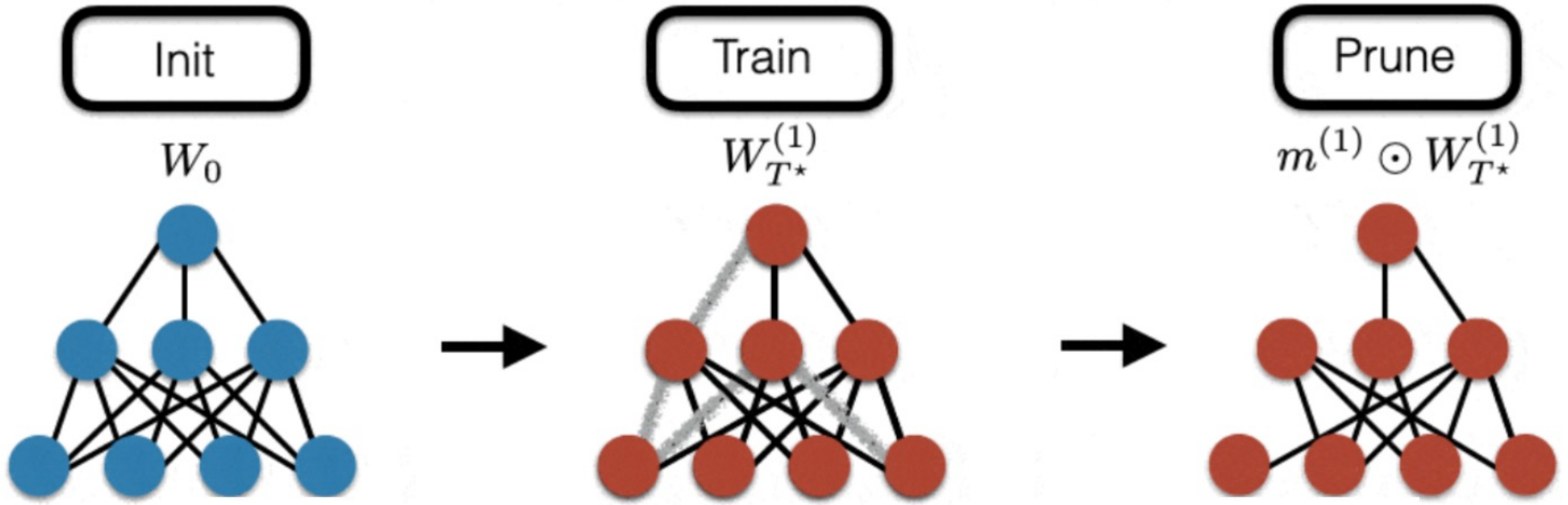
- Overparameterized NNs are less efficient and hard to deploy
- Research on pruning date back to 1990s. Han et al. (2015) prunes 90% NN weights and maintain performance matching the full NN

The Pruning Problem

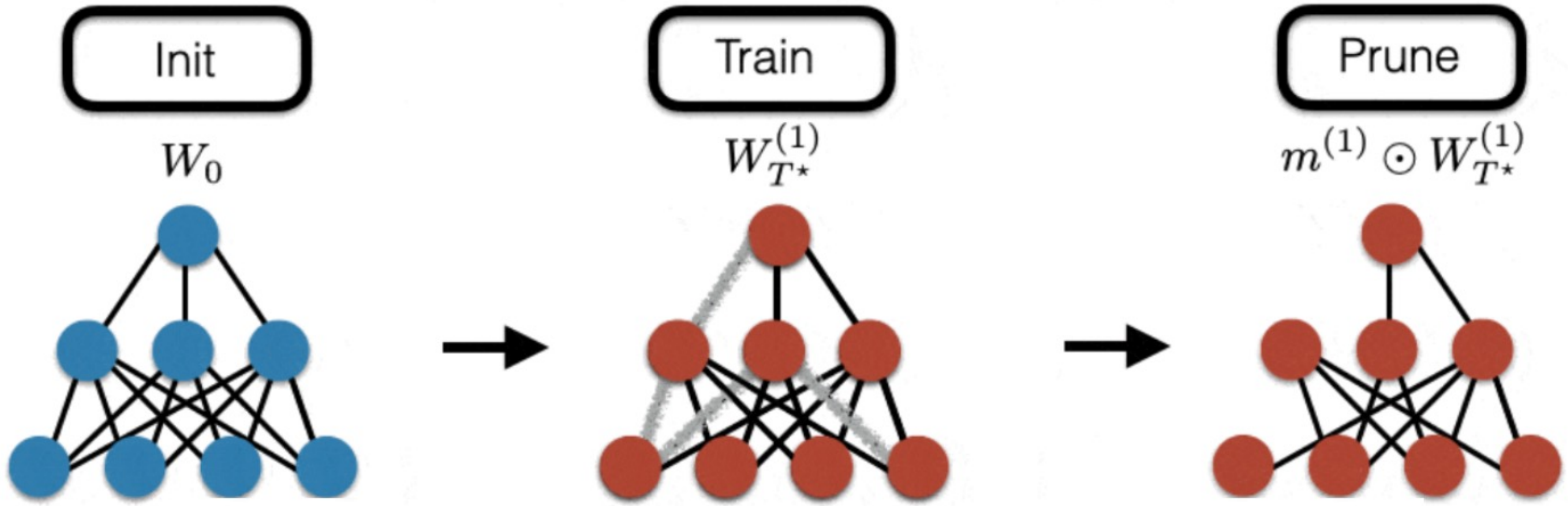
- Overparameterized NNs are less efficient and hard to deploy
- Research on pruning date back to 1990s. Han et al. (2015) prunes 90% NN weights and maintain performance matching the full NN
- Exists a small (sparse) NN with enough capacity

The Pruning Problem

- Overparameterized NNs are less efficient and hard to deploy
- Research on pruning date back to 1990s. Han et al. (2015) prunes 90% NN weights and maintain performance matching the full NN
- Exists a small (sparse) NN with enough capacity

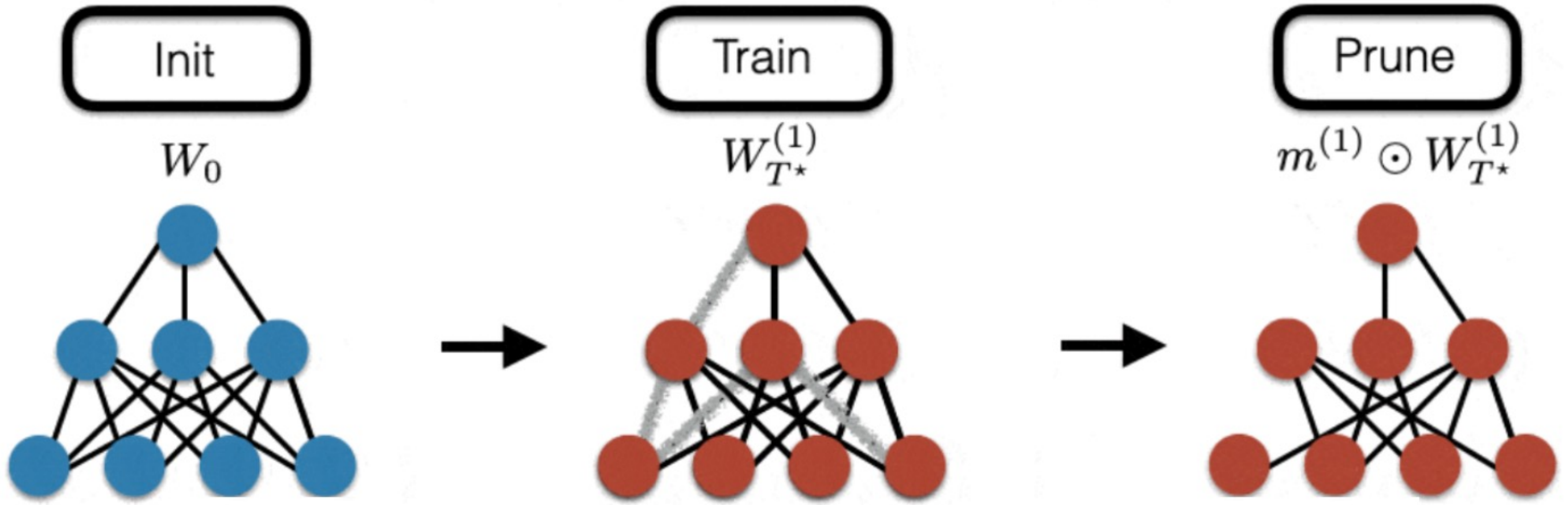


The Pruning Problem



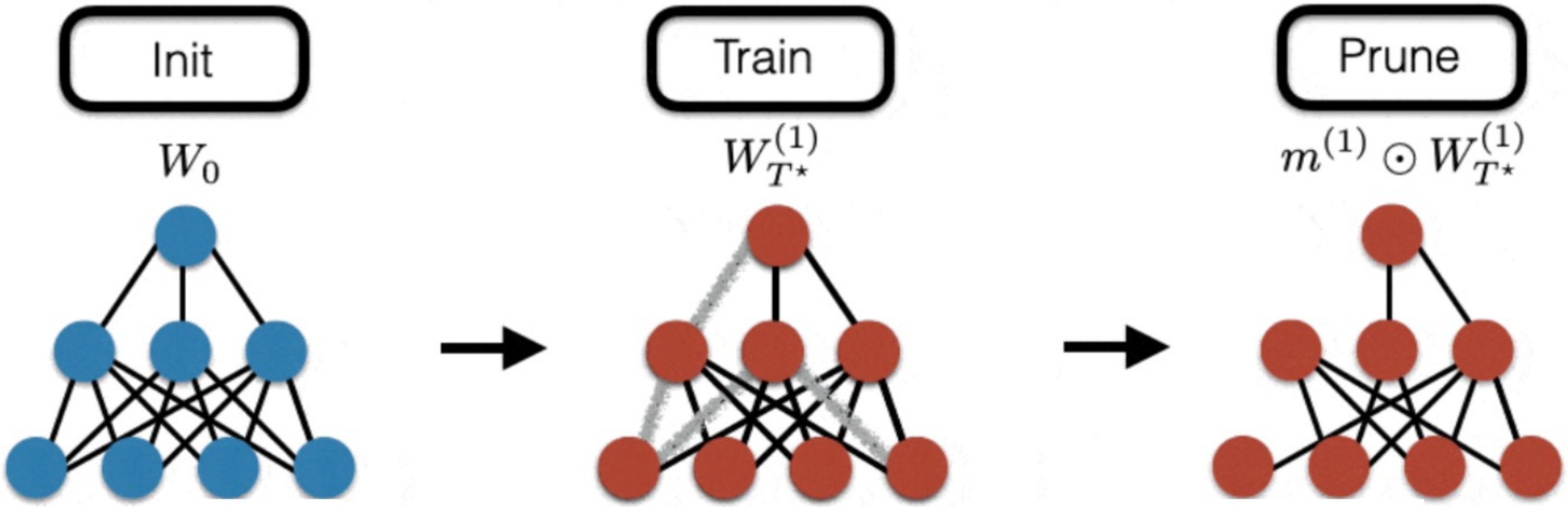
The Pruning Problem

- Such small (sparse) NNs can't be trained in isolation to achieve performance matching the full NN



The Pruning Problem

- Such small (sparse) NNs can't be trained in isolation to achieve performance matching the full NN
- Representing the learned knowledge doesn't require large capacity, but the learning process does



The Lottery Ticket Hypothesis (LTH)

THE LOTTERY TICKET HYPOTHESIS: FINDING SPARSE, TRAINABLE NEURAL NETWORKS

Jonathan Frankle
MIT CSAIL
jfrankle@csail.mit.edu

Michael Carbin
MIT CSAIL
mcarbin@csail.mit.edu

ICLR 2019 (Best Paper)

The Lottery Ticket Hypothesis (LTH)

THE LOTTERY TICKET HYPOTHESIS:
FINDING SPARSE, TRAINABLE NEURAL NETWORKS

Jonathan Frankle
MIT CSAIL
jfrankle@csail.mit.edu

Michael Carbin
MIT CSAIL
mcarbin@csail.mit.edu

ICLR 2019 (Best Paper)

Learning can also be done in isolation for model with a small capacity

The Lottery Ticket Hypothesis (LTH)

THE LOTTERY TICKET HYPOTHESIS:
FINDING SPARSE, TRAINABLE NEURAL NETWORKS

Jonathan Frankle
MIT CSAIL
jfrankle@csail.mit.edu

Michael Carbin
MIT CSAIL
mcarbin@csail.mit.edu

ICLR 2019 (Best Paper)

Learning can also be done in isolation for model with a small capacity

The Lottery Ticket Hypothesis. *A randomly-initialized, dense neural network contains a subnetwork that is initialized such that—when trained in isolation—it can match the test accuracy of the original network after training for at most the same number of iterations.*

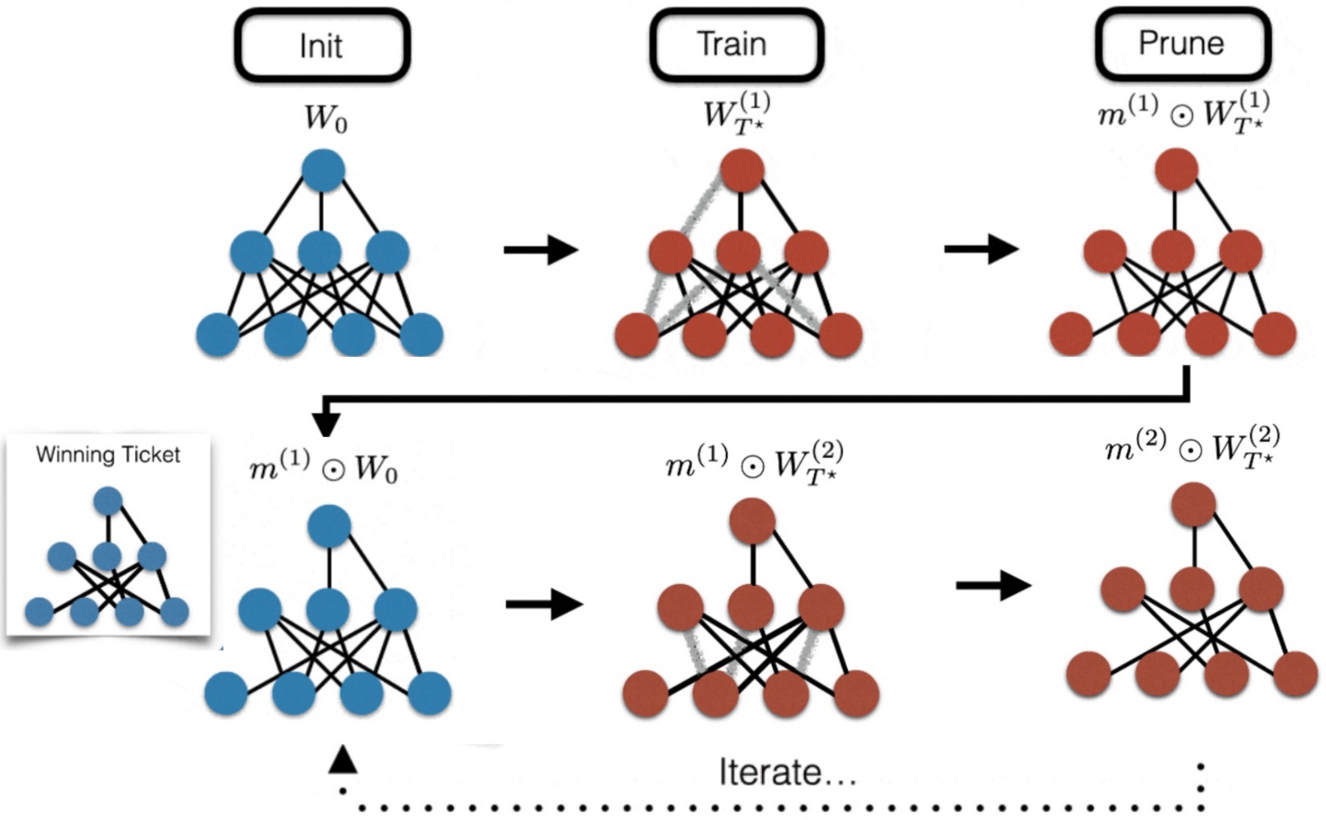
Finding The Winning Ticket

Finding The Winning Ticket

- Iterative magnitude pruning (IMP)

Finding The Winning Ticket

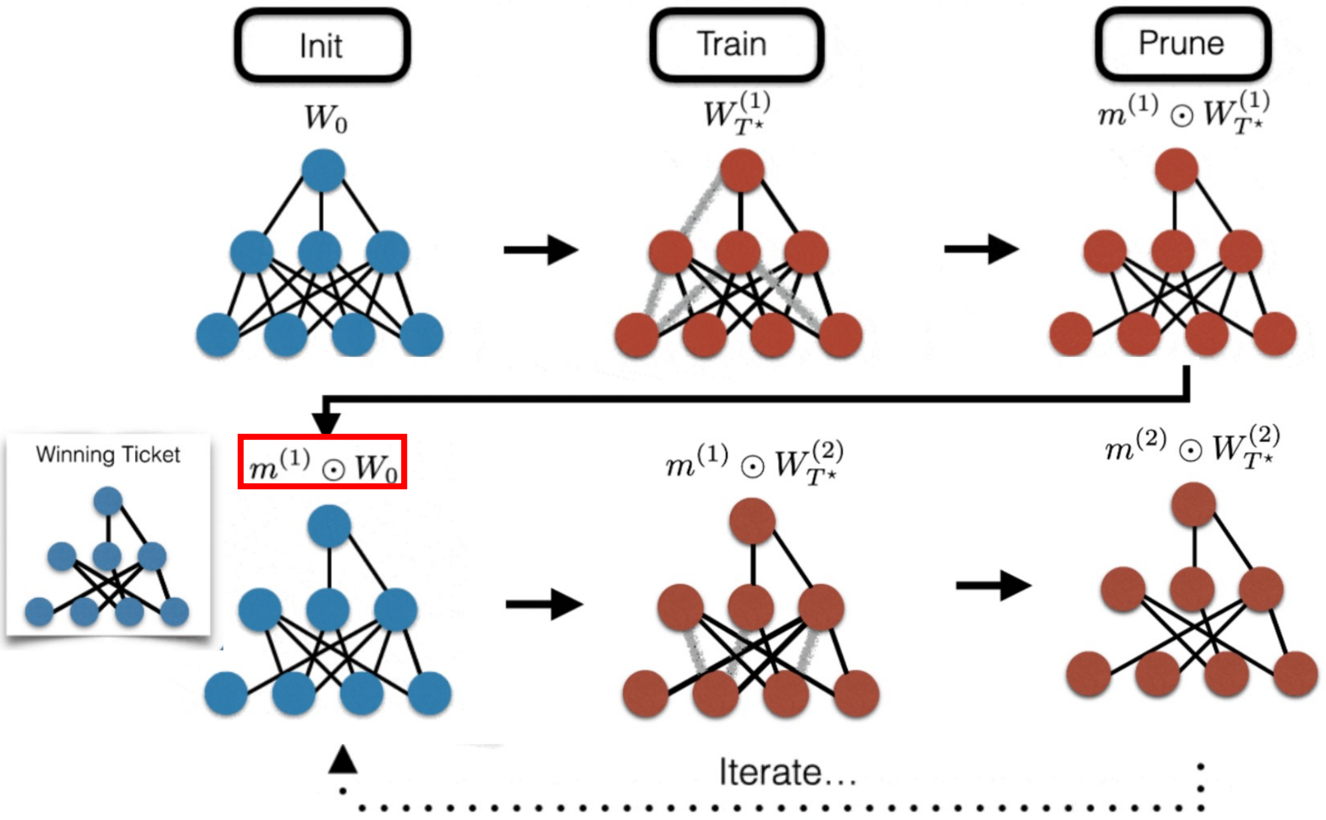
- Iterative magnitude pruning (IMP)



Lottery Ticket Hypothesis - IMP procedure [(Frankle & Carbin, 2019)](#)

Finding The Winning Ticket

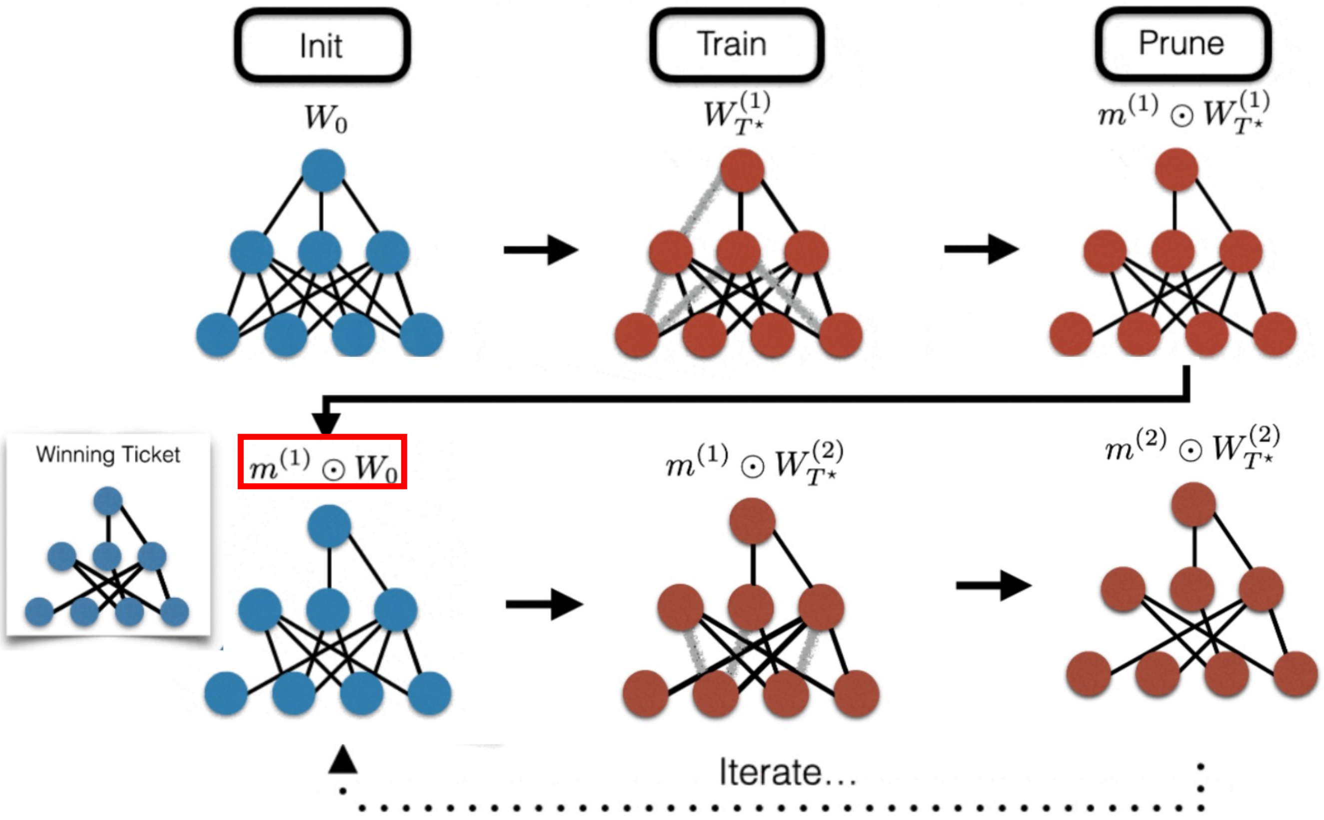
- Iterative magnitude pruning (IMP)



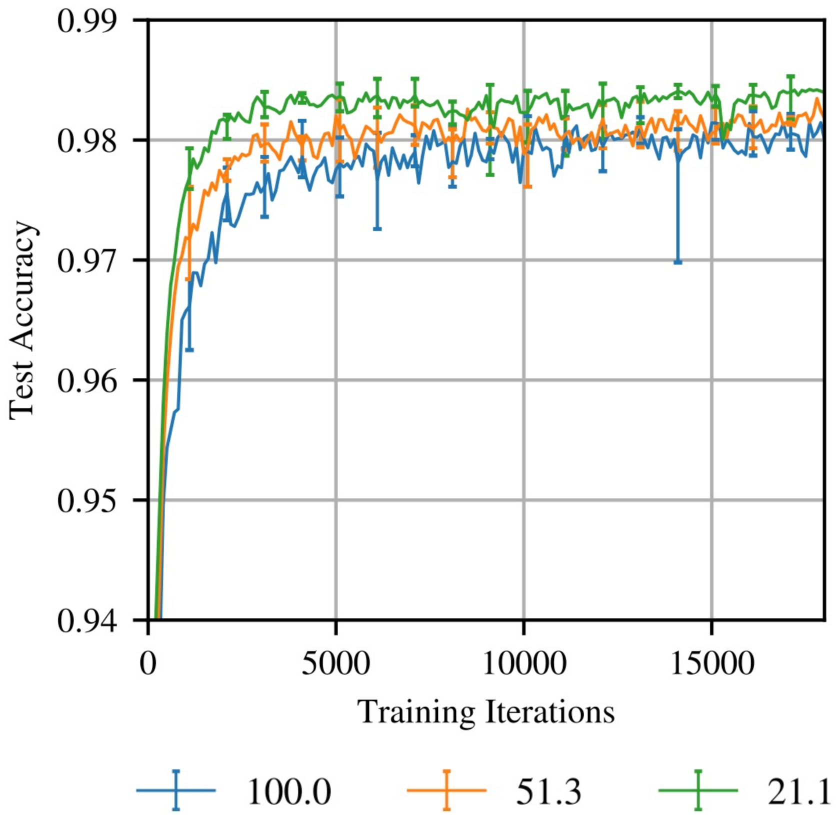
Lottery Ticket Hypothesis - IMP procedure [(Frankle & Carbin, 2019)]

Finding The Winning Ticket

- Iterative magnitude pruning (IMP)

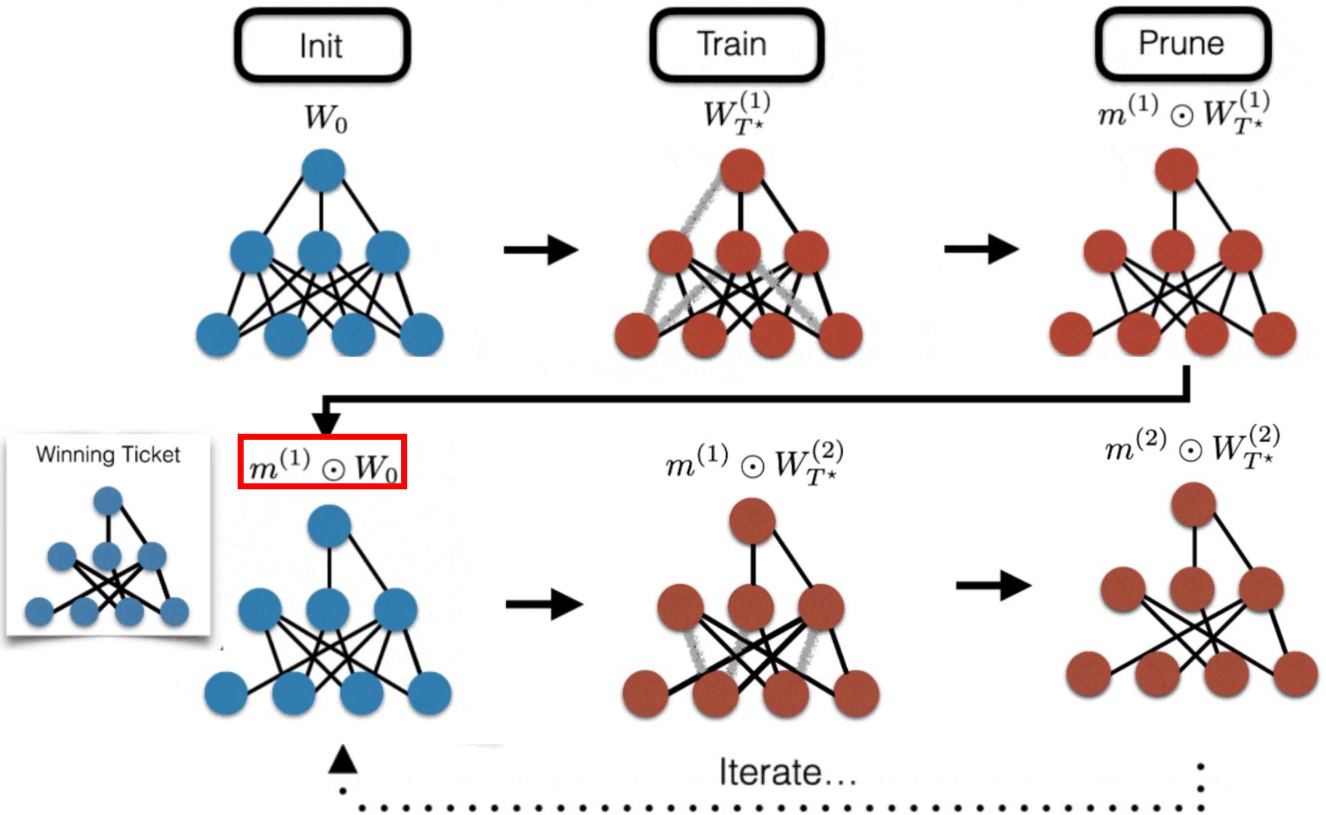


Lottery Ticket Hypothesis - IMP procedure [(Frankle & Carbin, 2019)](#)

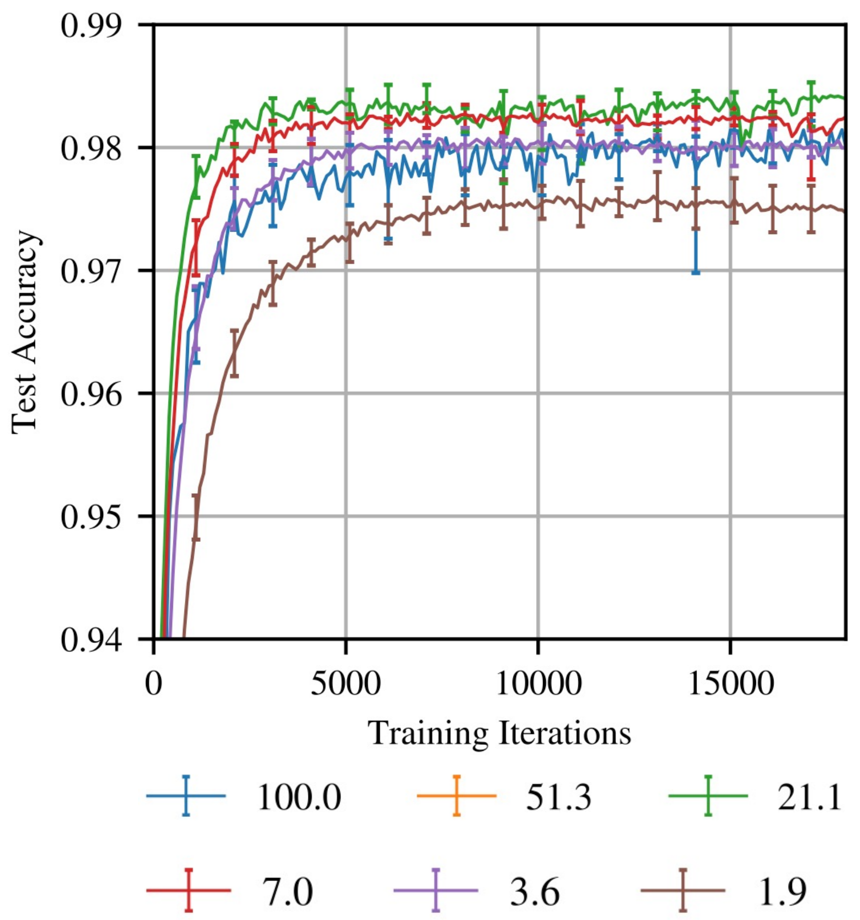


Finding The Winning Ticket

- Iterative magnitude pruning (IMP)

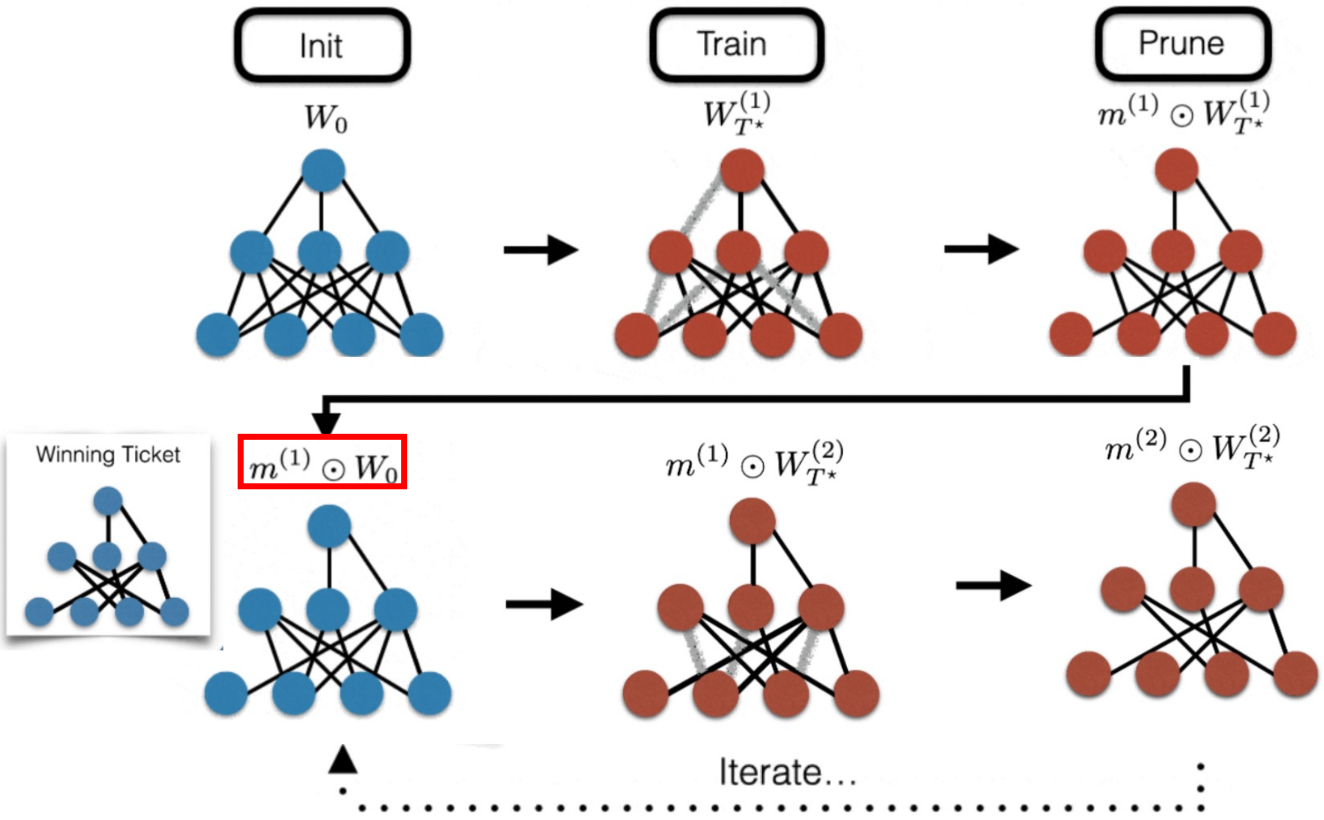


Lottery Ticket Hypothesis - IMP procedure [(Frankle & Carbin, 2019)](#)



Finding The Winning Tickets

- Iterative magnitude pruning (IMP)

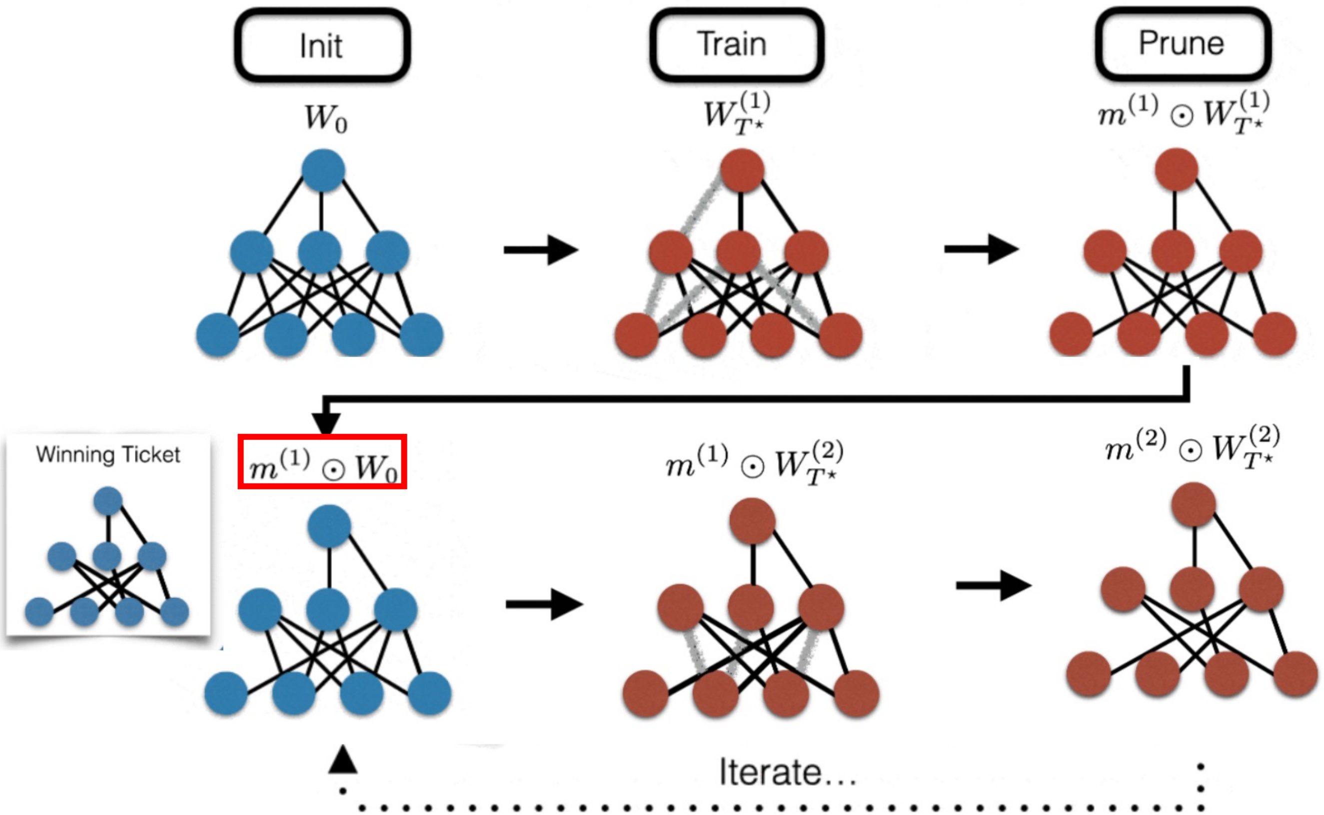


Lottery Ticket Hypothesis - IMP procedure [(Frankle & Carbin, 2019)]

- Hypothesis

Finding The Winning Tickets

- Iterative magnitude pruning (IMP)

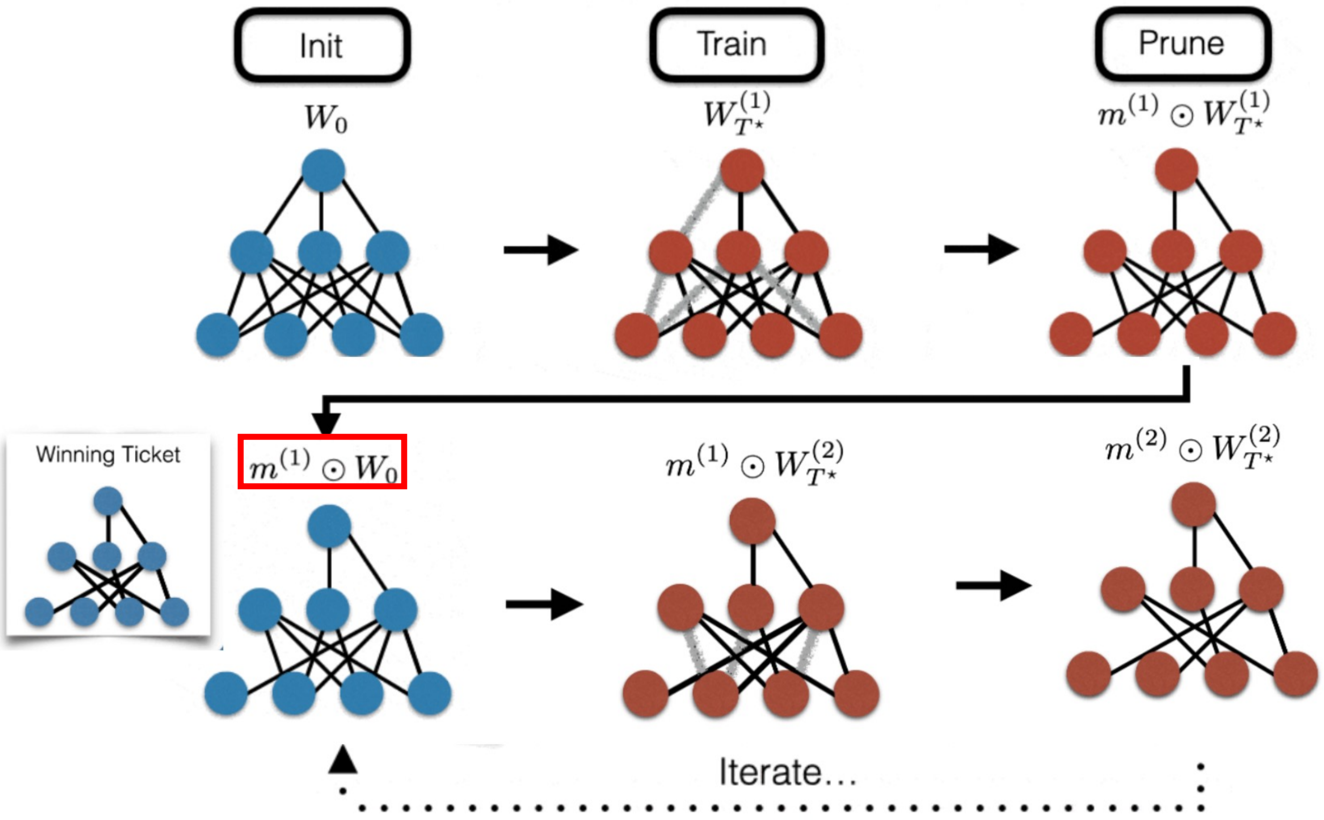


Lottery Ticket Hypothesis - IMP procedure [(Frankle & Carbin, 2019)]

- Hypothesis
  - Training includes initialization and optimization

Finding The Winning Tickets

- Iterative magnitude pruning (IMP)

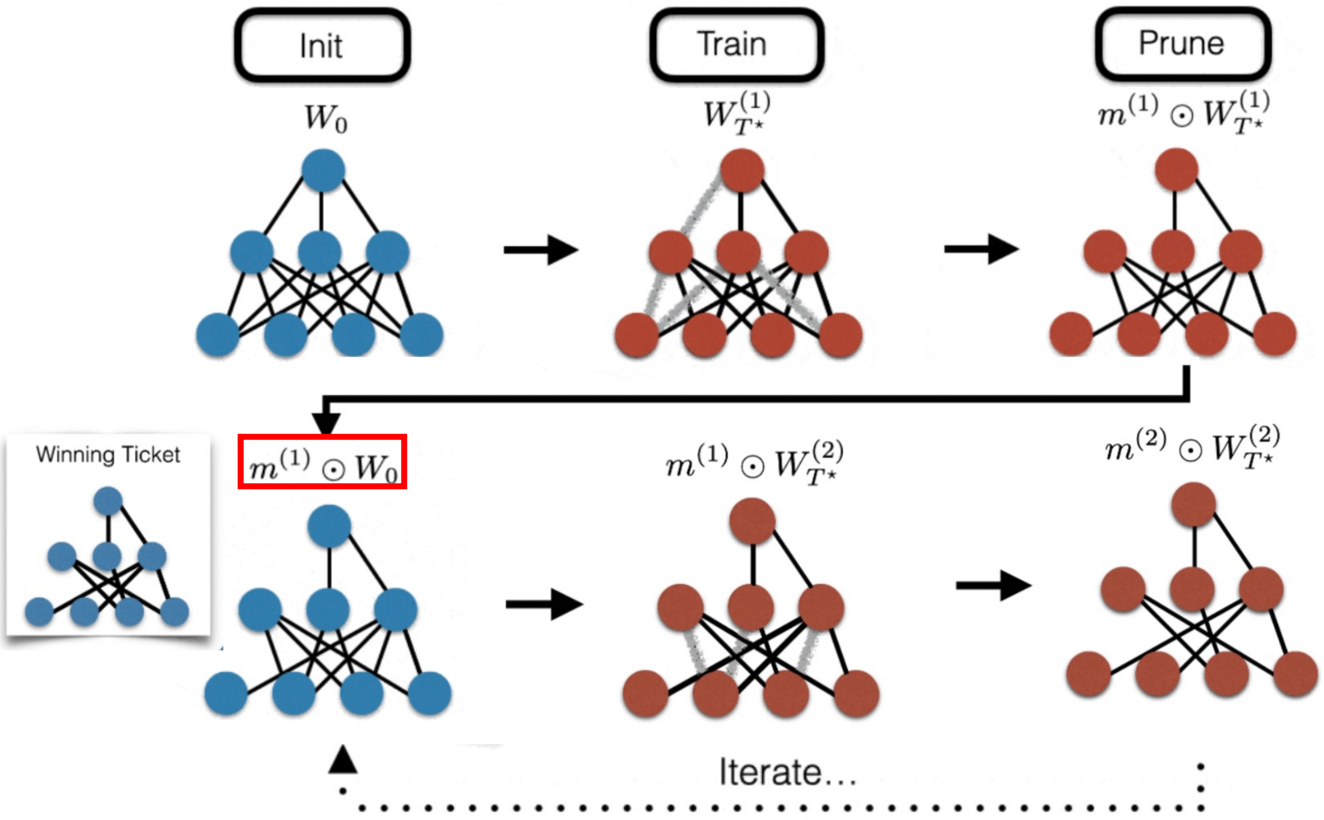


Lottery Ticket Hypothesis - IMP procedure (Frankle & Carbin, 2019)

- Hypothesis
  - Training includes initialization and optimization
  - The initialization step is a lottery

Finding The Winning Tickets

- Iterative magnitude pruning (IMP)

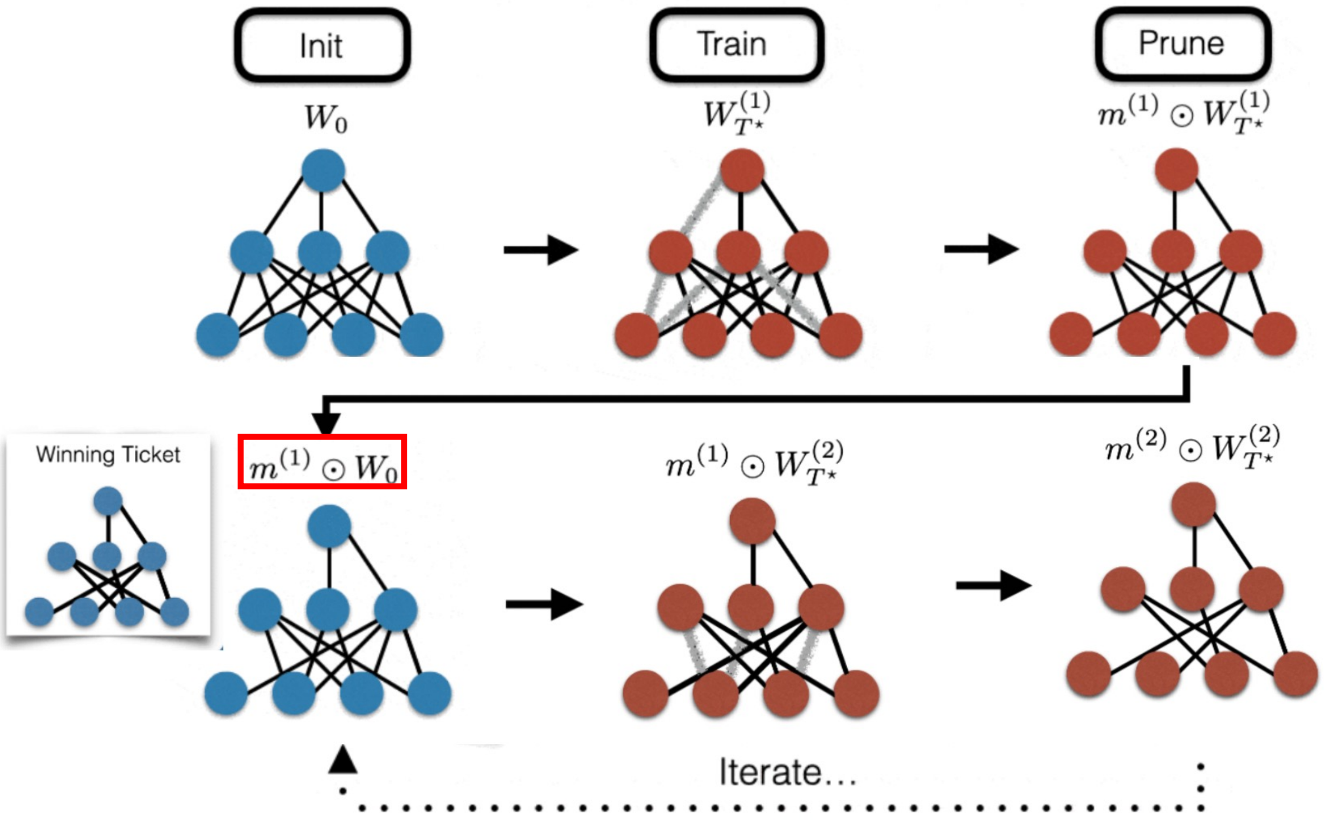


Lottery Ticket Hypothesis - IMP procedure [(Frankle & Carbin, 2019)]

- Hypothesis
  - Training includes initialization and optimization
  - The initialization step is a lottery
  - A lucky subnetwork will be selected and driving the optimization process, which can be trained in isolation to achieve performance matching the full NN

Finding The Winning Tickets

- Iterative magnitude pruning (IMP)



Lottery Ticket Hypothesis - IMP procedure [(Frankle & Carbin, 2019)]

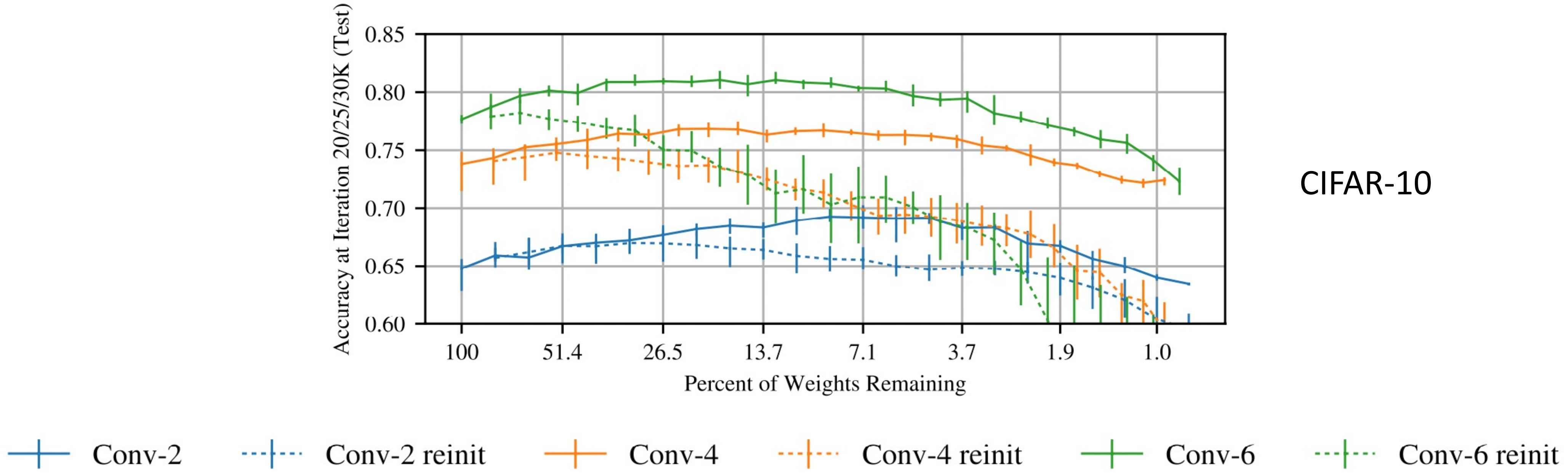
- Hypothesis
  - Training includes initialization and optimization
  - The initialization step is a lottery
  - A lucky subnetwork will be selected and driving the optimization process, which can be trained in isolation to achieve performance matching the full NN
  - One initialization maps to one set of winning tickets

Extensive Experiments to Support the LTH

- Different architectures
  - LeNet, ResNet-18, and VGG
- Different datasets
  - MNIST, CIFAR-10

Extensive Experiments to Support the LTH

- Different architectures
  - LeNet, ResNet-18, and VGG
- Different datasets
  - MNIST, CIFAR-10



CIFAR-10

Summary

Summary

- Contributions
  - Propose the Lottery Ticket **Hypothesis**, which shows that retraining the pruned NN is possible
  - Verified the hypothesis with extensive experiments

Summary

- Contributions
  - Propose the Lottery Ticket **Hypothesis**, which shows that retraining the pruned NN is possible
  - Verified the hypothesis with extensive experiments
- Limitations
  - Experiment scale is still relatively small
    - Later on, people found that the IMP strategy fails on ImageNet with large NNs

Summary

- Contributions
  - Propose the Lottery Ticket **Hypothesis**, which shows that retraining the pruned NN is possible
  - Verified the hypothesis with extensive experiments
- Limitations
  - Experiment scale is still relatively small
    - Later on, people found that the IMP strategy fails on ImageNet with large NNs
  - IMP requires expensive iterative training
    - Making LTH more of an analysis tool rather than a practical pruning method

Can LTH Scales Up?

Can LTH Scales Up?

Linear Mode Connectivity and the Lottery Ticket Hypothesis

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

ICML 2020

Can LTH Scales Up?

Linear Mode Connectivity and the Lottery Ticket Hypothesis

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

ICML 2020

Yes. If we go back to somewhere close to the beginning, but not the initial state for retraining.

Research Question

- How does the stochastic gradient descent (SGD) training affect the optimization trajectories of NNs?

Research Question

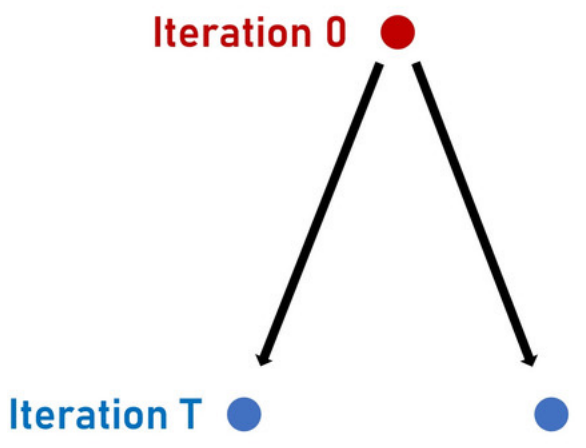
- How does the $\boxed{\text{stochastic}}$ gradient descent (SGD) training affect the optimization trajectories of NNs?

data order

Research Question

- How does the stochastic gradient descent (SGD) training affect the optimization trajectories of NNs?

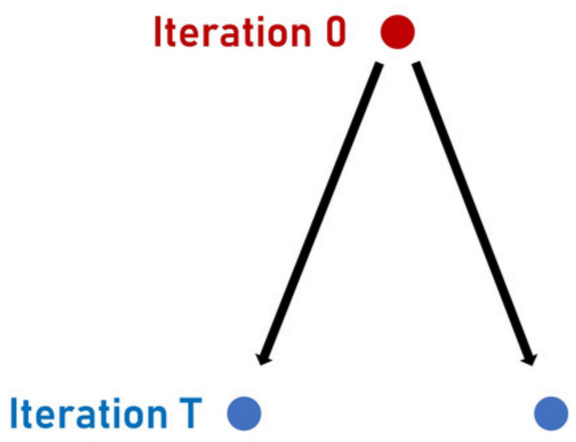
data order

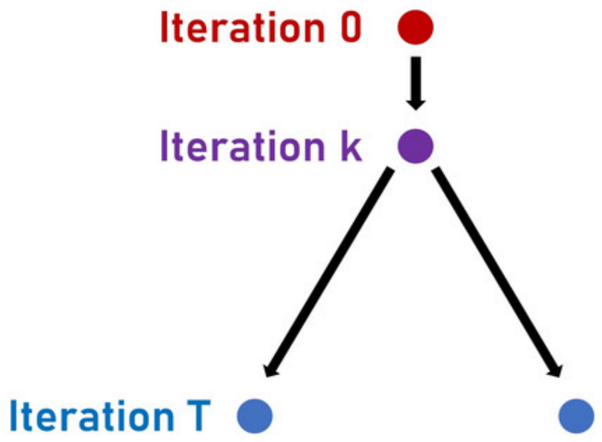


Research Question

- How does the stochastic gradient descent (SGD) training affect the optimization trajectories of NNs?

data order





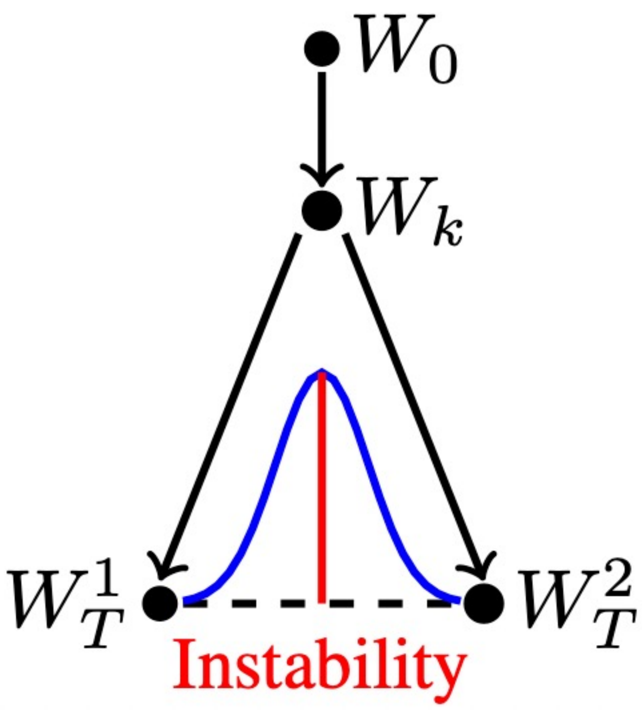
Instability Analysis

Instability Analysis

- *linear interpolation instability* (instability) of N to SGD noise as the maximum increase in error along this path (red)

Instability Analysis

- *linear interpolation instability* (instability) of N to SGD noise as the maximum increase in error along this path (red)



Instability Analysis

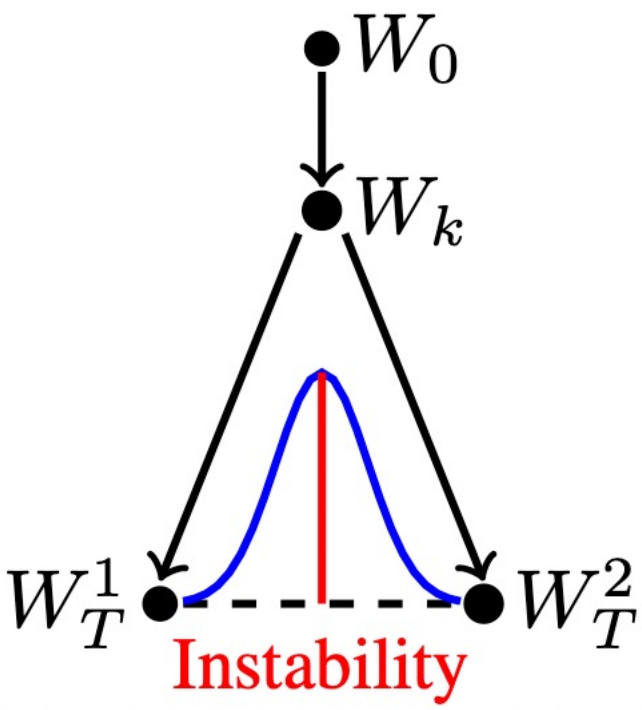
- *linear interpolation instability* (instability) of N to SGD noise as the maximum increase in error along this path (red)

$$\mathcal{E}_\alpha(W_1, W_2) = \mathcal{E}(\alpha W_1 + (1-\alpha) W_2)$$

$$\mathcal{E}_{\mathsf{sup}}(W_1, W_2) \;=\; \sup_\alpha \mathcal{E}_\alpha(W_1, W_2)$$

$$\bar{\mathcal{E}}(W_1, W_2) = \mathsf{mean}(\mathcal{E}(W_1), \mathcal{E}(W_2))$$

$$\mathcal{E}_{\mathsf{sup}}(W_1, W_2) - \bar{\mathcal{E}}(W_1, W_2)$$



Instability Analysis

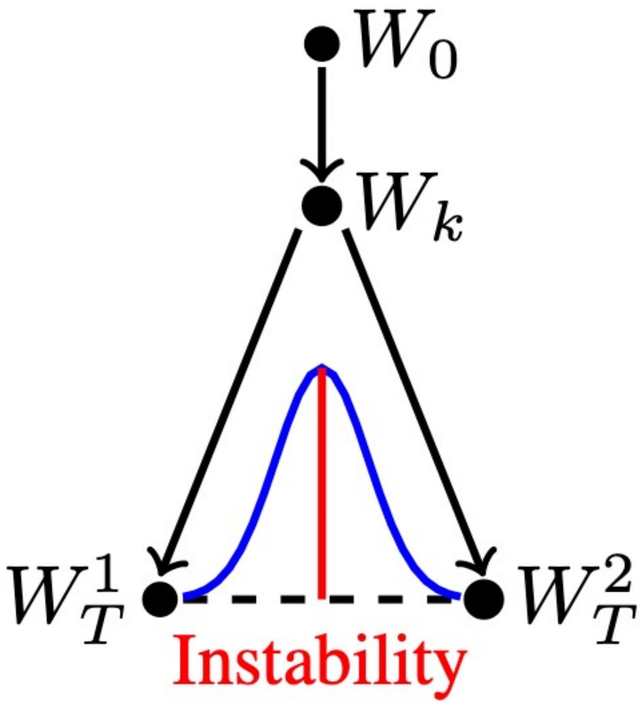
- *linear interpolation instability* (instability) of N to SGD noise as the maximum increase in error along this path (red)

$$\mathcal{E}_\alpha(W_1, W_2) = \mathcal{E}(\alpha W_1 + (1-\alpha)W_2)$$

$$\mathcal{E}_{\textsf{sup}}(W_1, W_2) \; = \; \sup_\alpha \mathcal{E}_\alpha(W_1, W_2)$$

$$\bar{\mathcal{E}}(W_1, W_2) = \textsf{mean}(\mathcal{E}(W_1), \mathcal{E}(W_2))$$

$$\boxed{\mathcal{E}_{\textsf{sup}}(W_1, W_2) - \bar{\mathcal{E}}(W_1, W_2)}$$



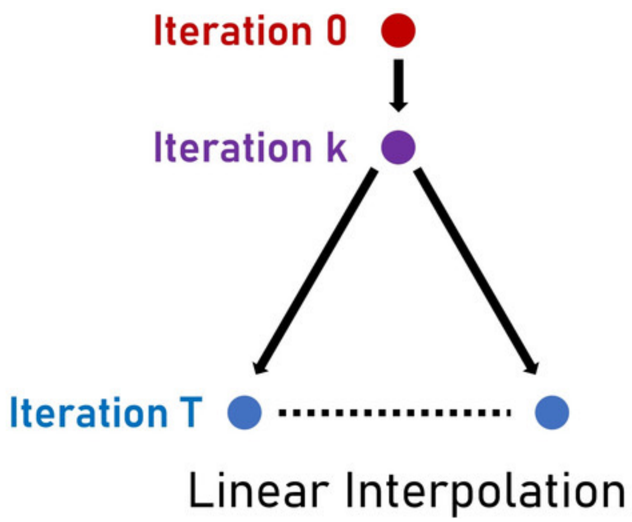
Linear Mode Connectivity

Linear Mode Connectivity

- Instability < threshold

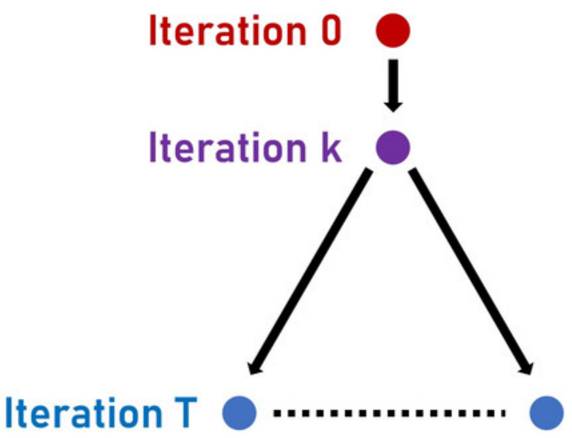
Linear Mode Connectivity

- Instability < threshold

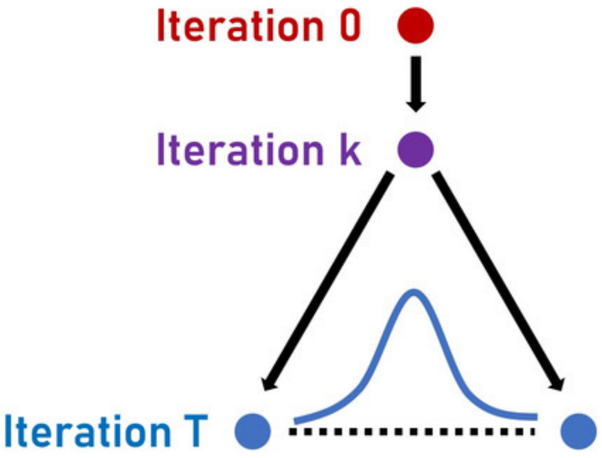


Linear Mode Connectivity

- Instability < threshold

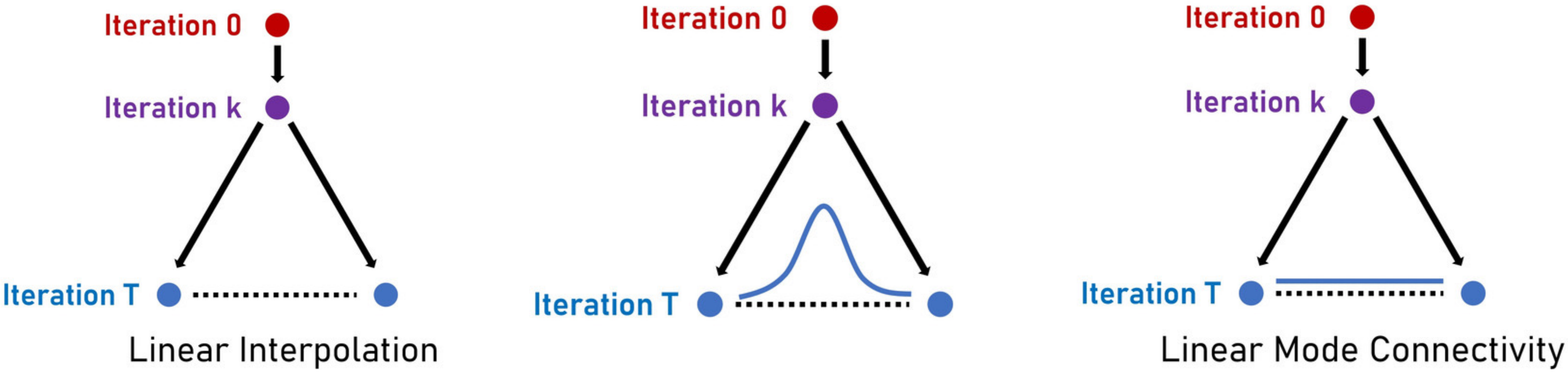


Linear Interpolation



Linear Mode Connectivity

- Instability < threshold



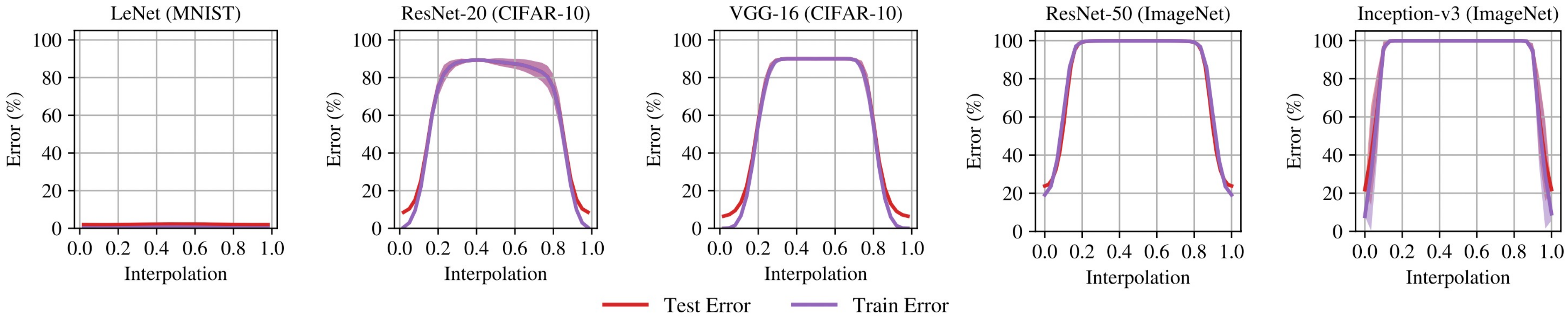
Instability Analysis of Unpruned Networks

Instability Analysis of Unpruned Networks

- SGD training is very unstable

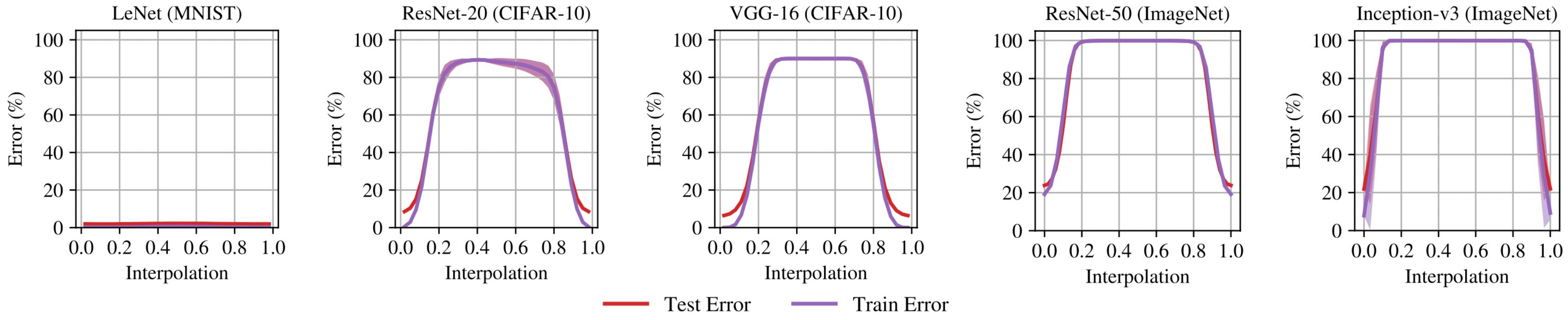
Instability Analysis of Unpruned Networks

- SGD training is very unstable



Instability Analysis of Unpruned Networks

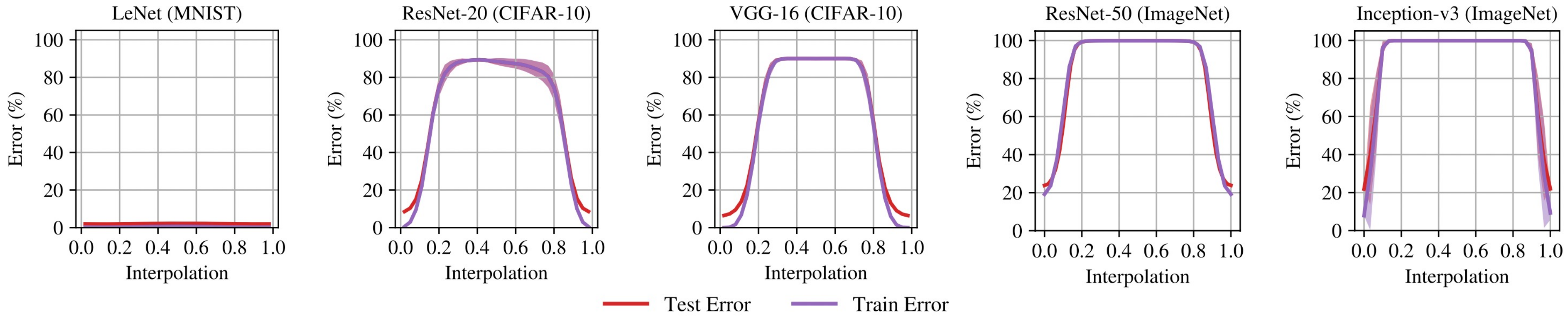
- SGD training is very unstable



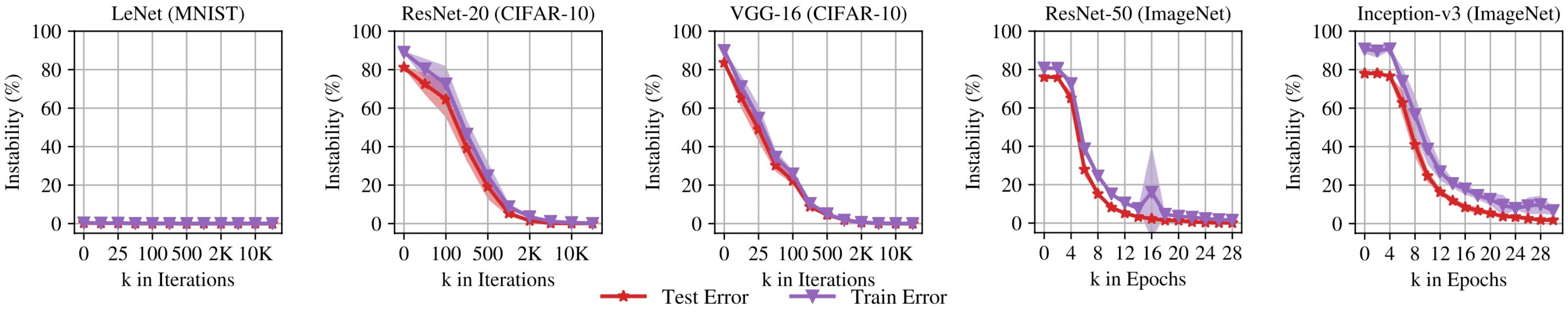
- The instability decreases as we increase k

Instability Analysis of Unpruned Networks

- SGD training is very unstable

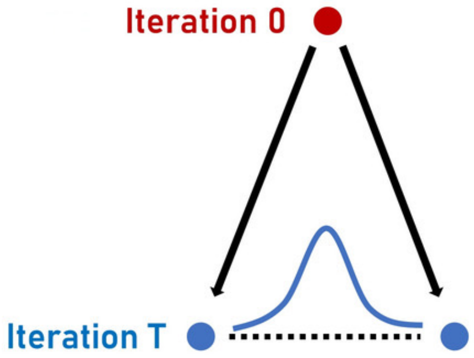


- The instability decreases as we increase k



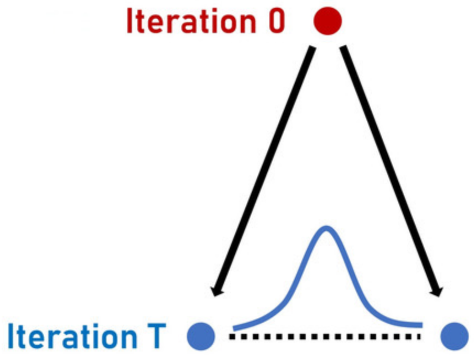
Linear Mode Connectivity in Early Training

Linear Mode Connectivity in Early Training

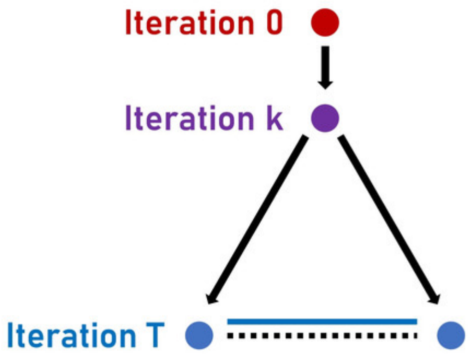


From iteration 0, we do not find linear mode connectivity.

Linear Mode Connectivity in Early Training

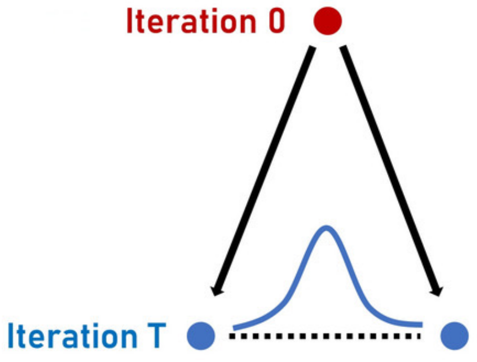


From iteration 0, we do not find linear mode connectivity.

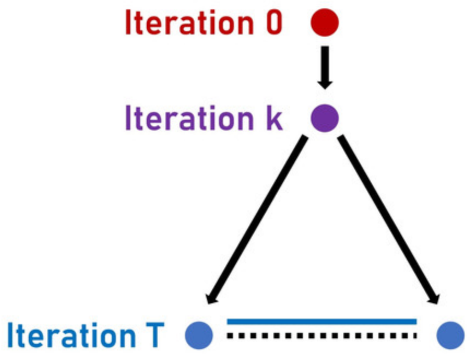


From early in training, we find linear mode connectivity

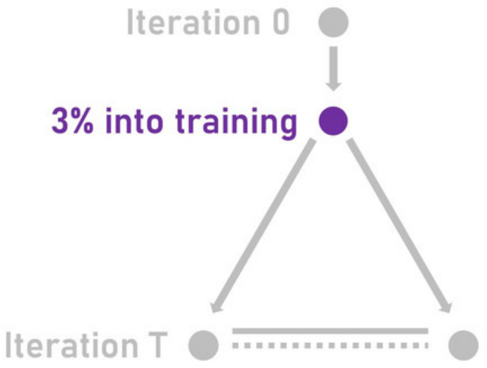
Linear Mode Connectivity in Early Training



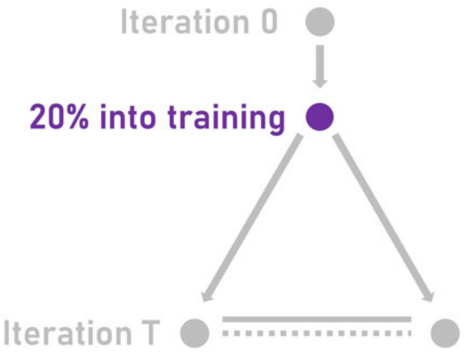
From iteration 0, we do not find linear mode connectivity.



From early in training, we find linear mode connectivity



ResNet-20 on CIFAR-10



ResNet-50 and Inception-v3 on ImageNet

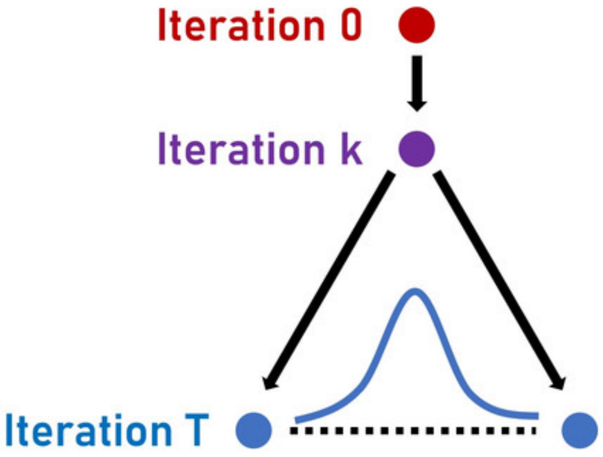
Disentangle Instability from Training Time

Disentangle Instability from Training Time

- Caveat: larger k will naturally lead to less instability

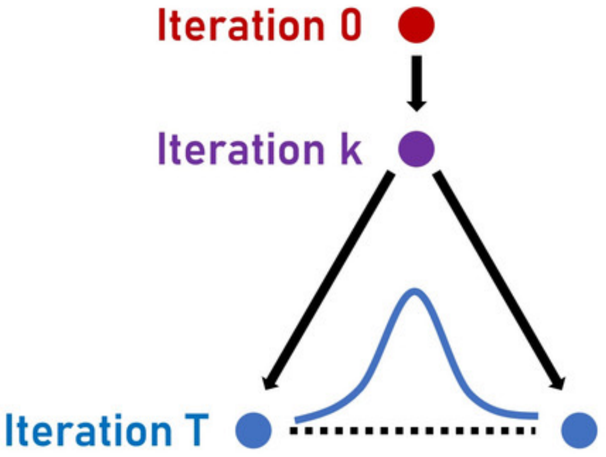
Disentangle Instability from Training Time

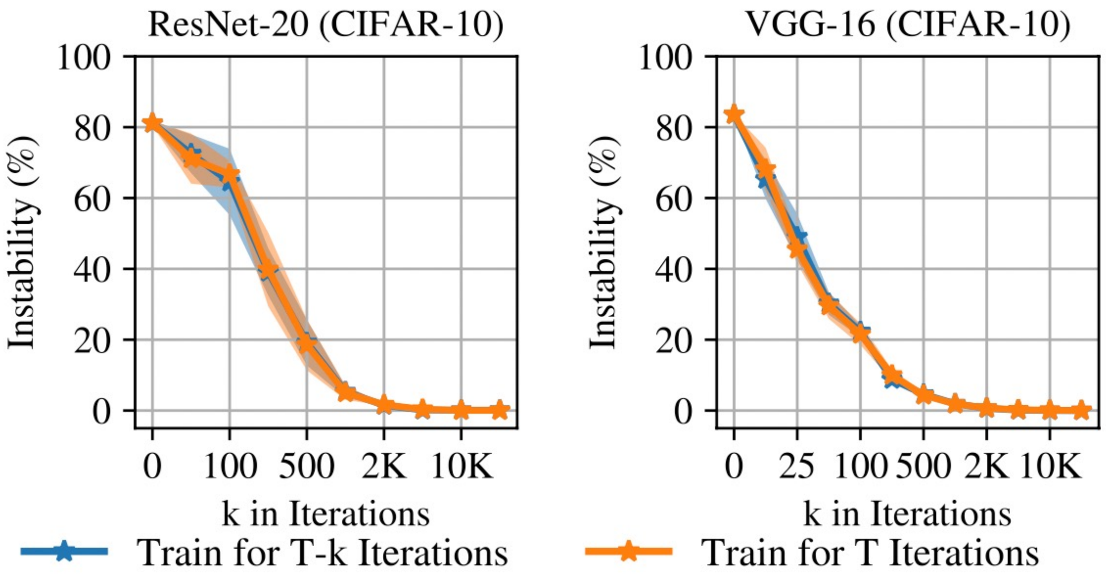
- Caveat: larger k will naturally lead to less instability



Disentangle Instability from Training Time

- Caveat: larger k will naturally lead to less instability





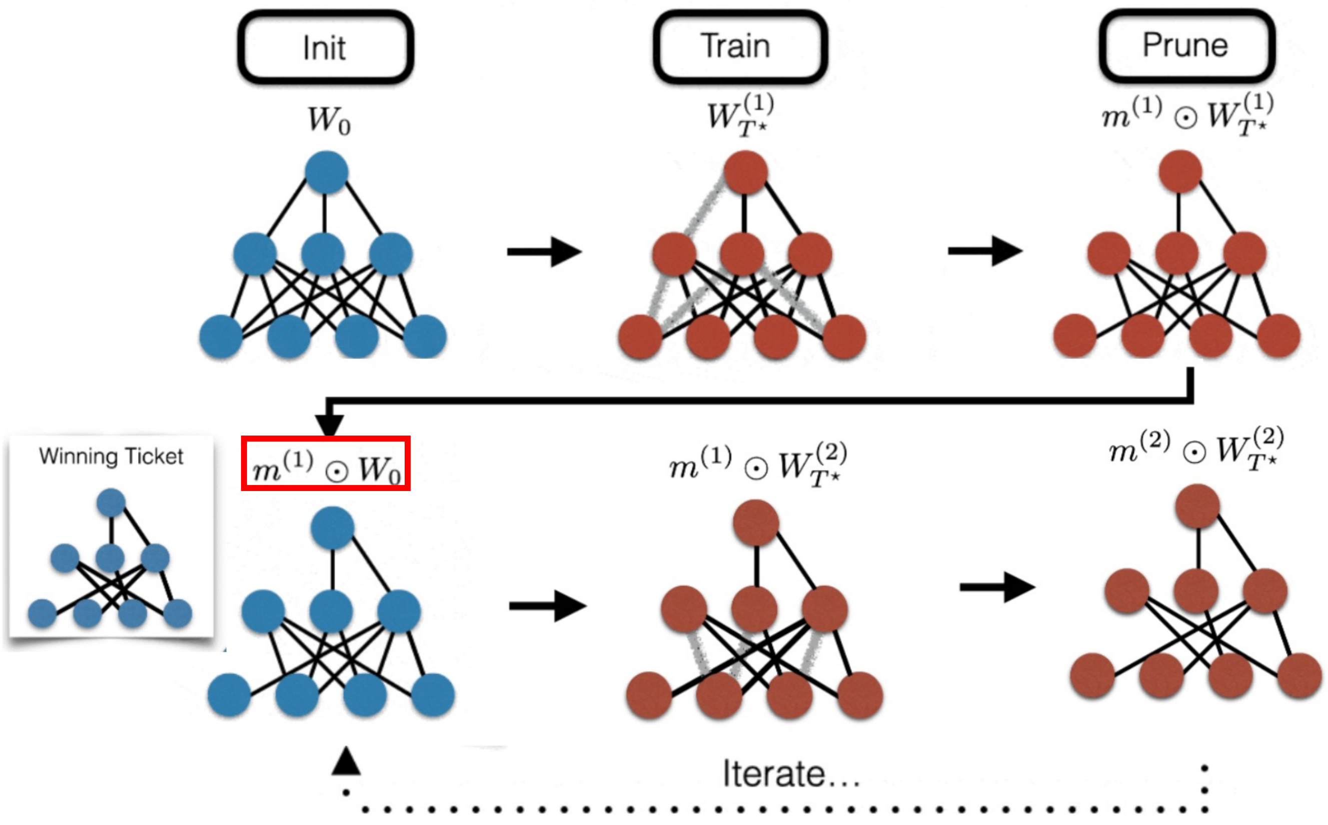
Instability Analysis for LTH

Instability Analysis for LTH

- Recall: Finding The Winning Tickets with IMP

Instability Analysis for LTH

- Recall: Finding The Winning Tickets with IMP



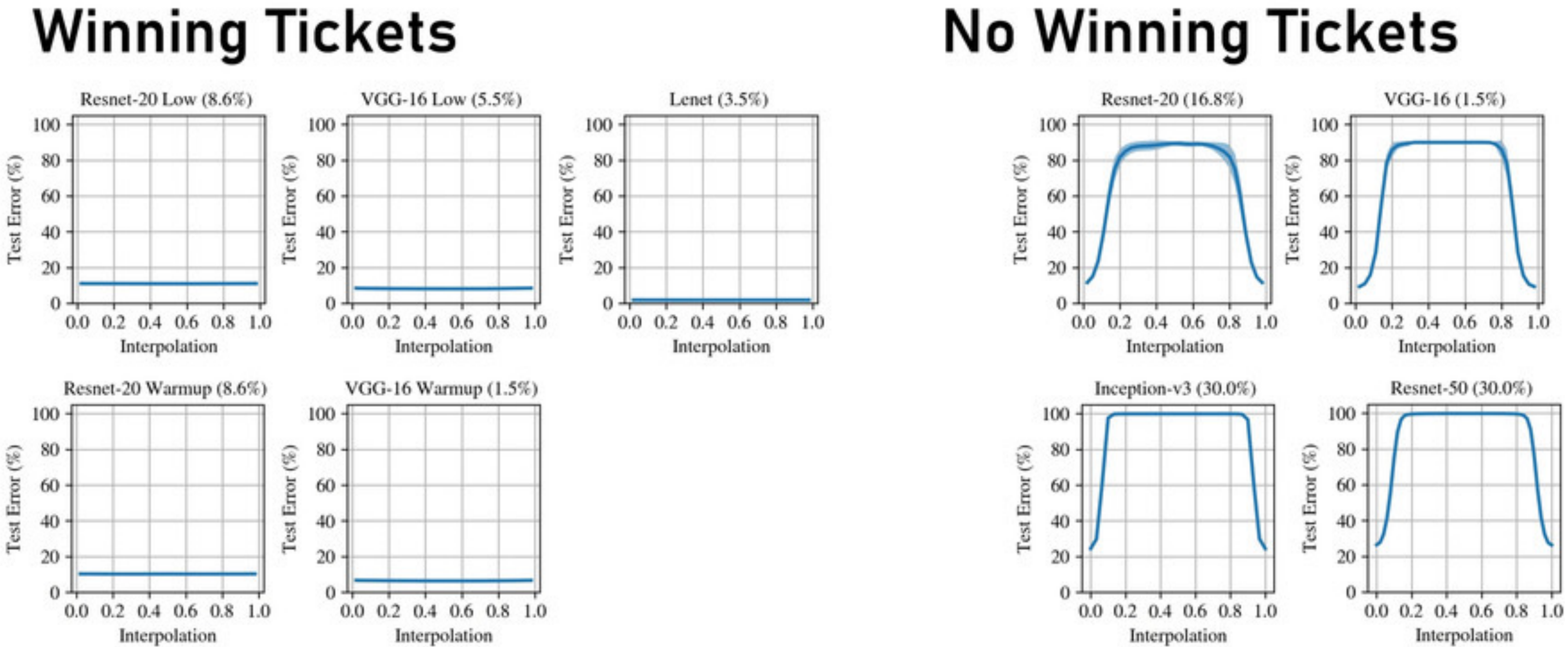
Lottery Ticket Hypothesis - IMP procedure [(Frankle & Carbin, 2019)](#)

Instability Analysis for LTH

- Linear connectivity correlates with winning tickets

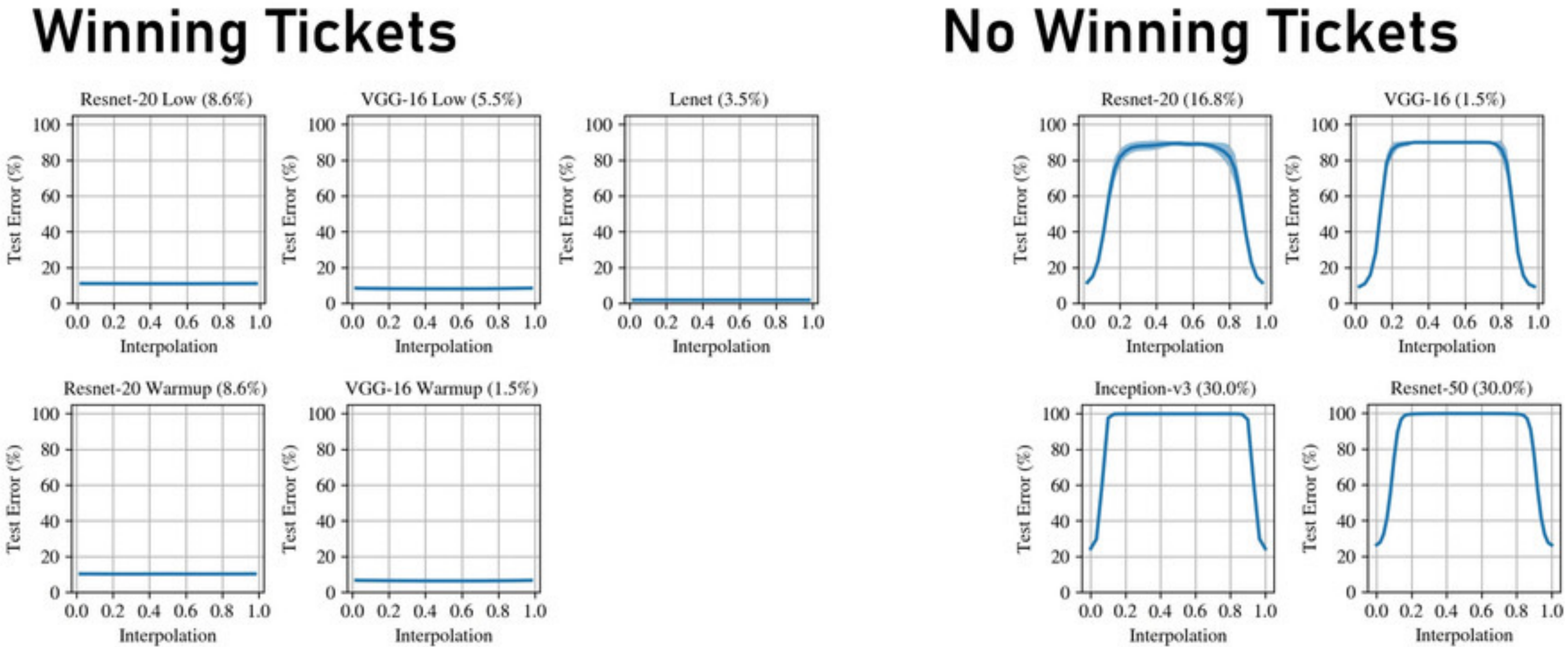
Instability Analysis for LTH

- Linear connectivity correlates with winning tickets



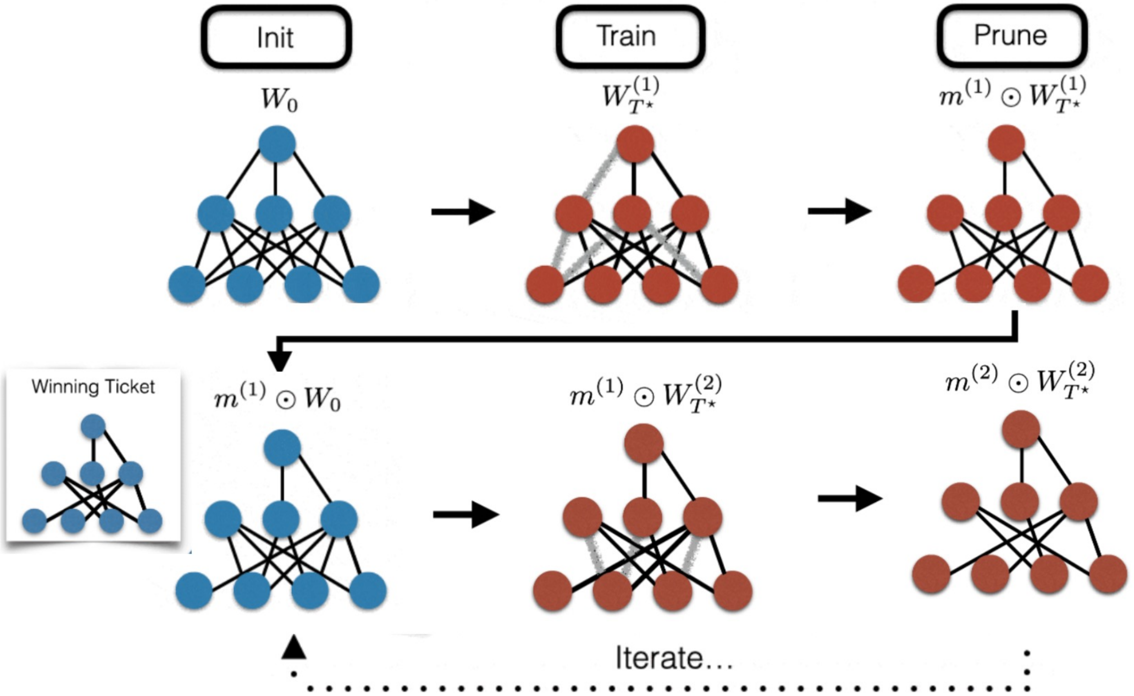
Instability Analysis for LTH

- Linear connectivity correlates with winning tickets
- Hypothesis: Instability is a problem for sparse NNs but not dense NNs, sparse NNs get stuck in local optimum



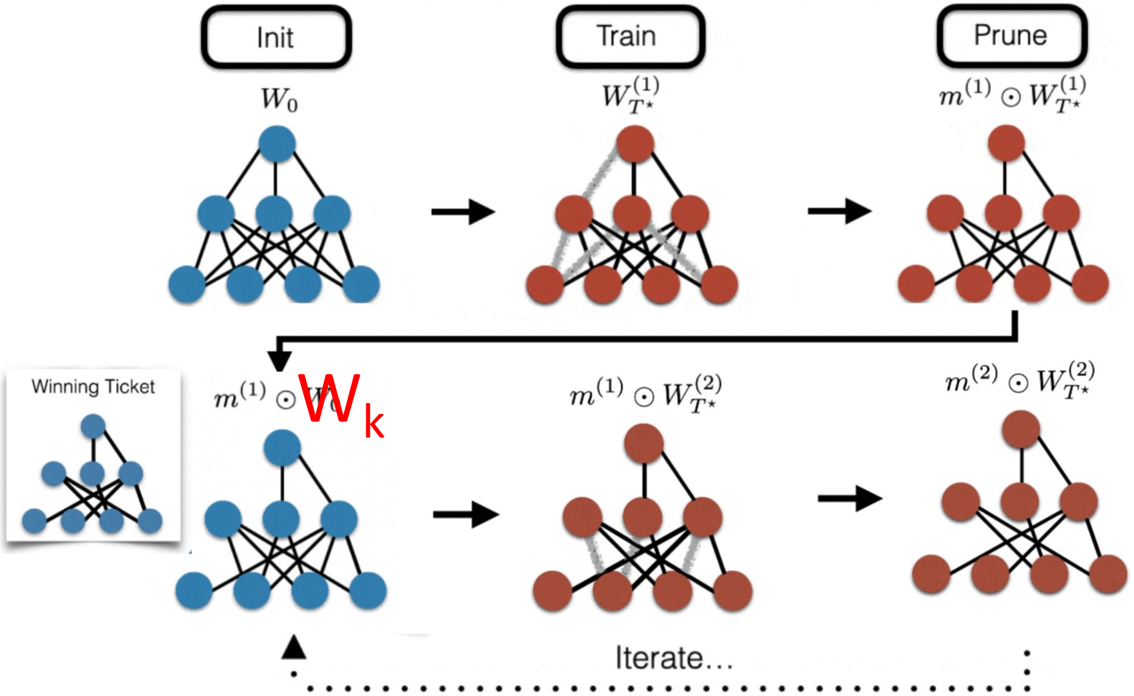
Rewind to Iteration k

Rewind to Iteration k



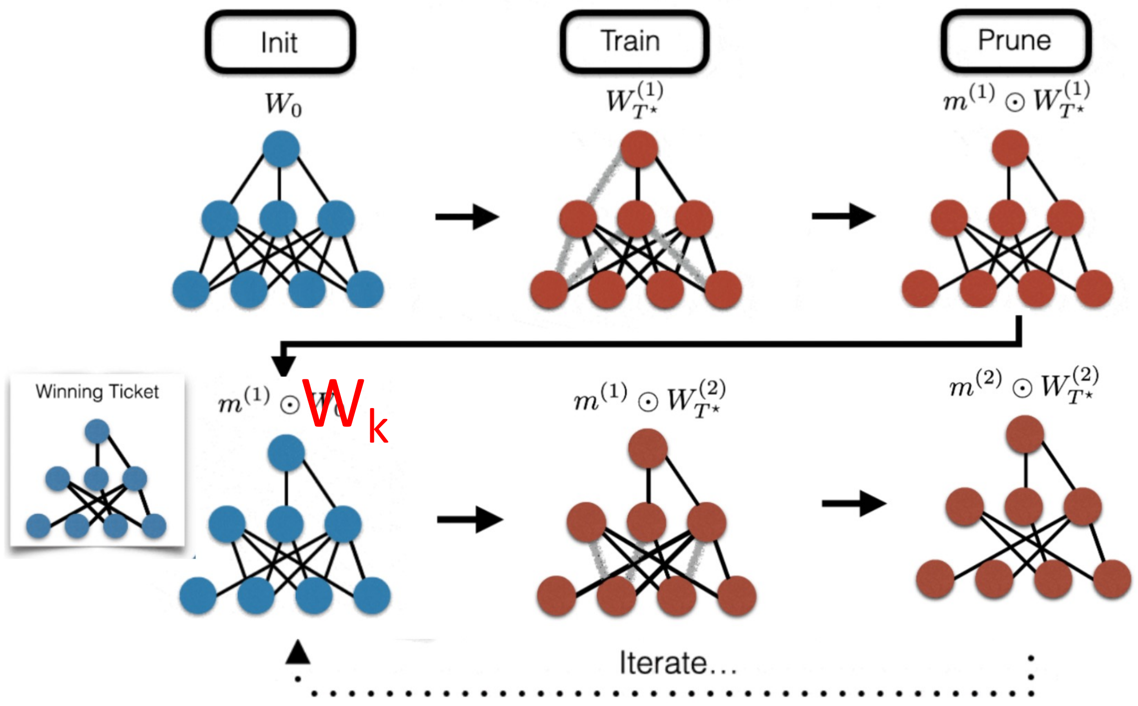
Lottery Ticket Hypothesis - IMP procedure [(Frankle & Carbin, 2019)]

Rewind to Iteration k



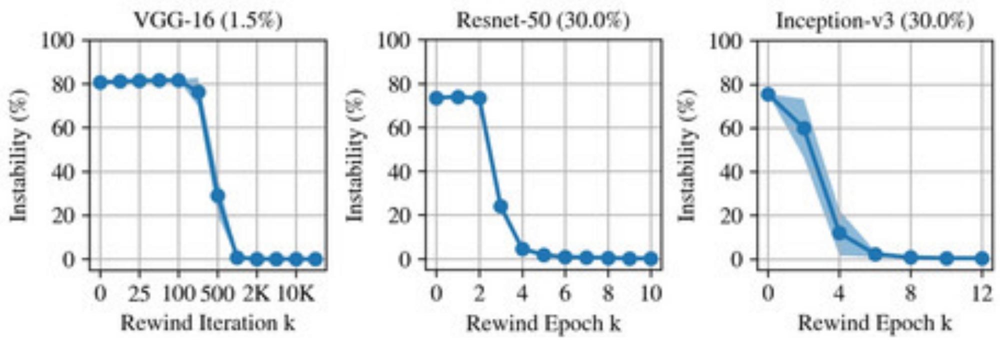
Lottery Ticket Hypothesis - IMP procedure (Frankle & Carbin, 2019)

Rewind to Iteration k

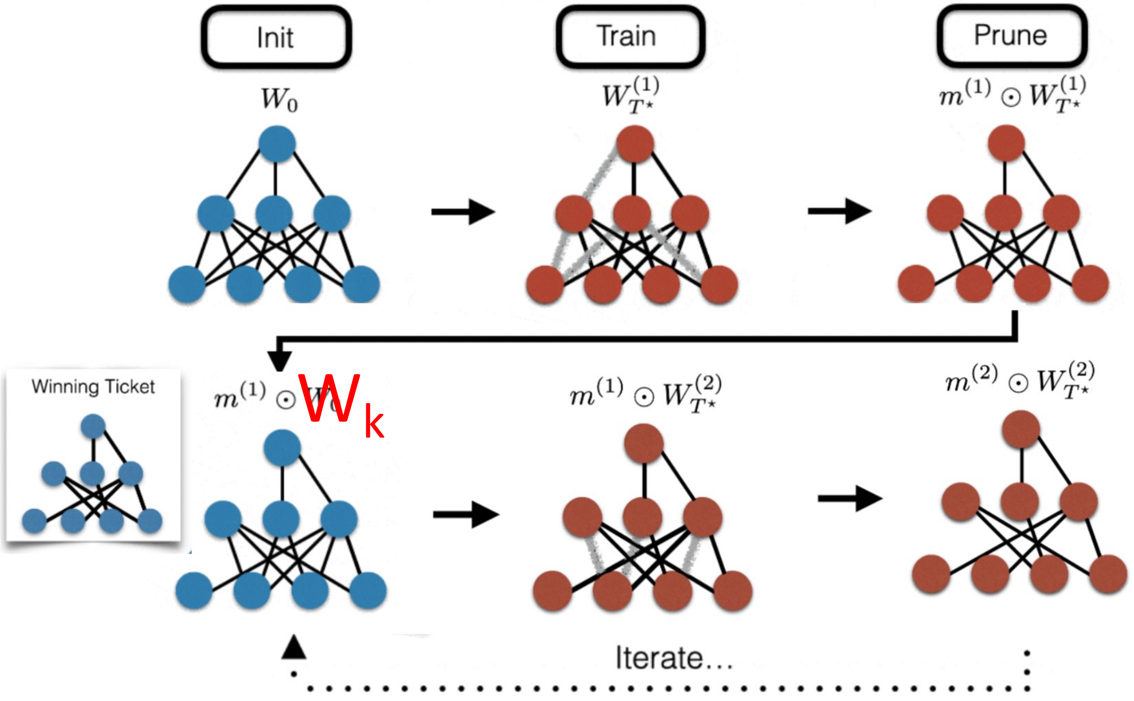


Lottery Ticket Hypothesis - IMP procedure [(Frankle & Carbin, 2019)]

No Winning Tickets

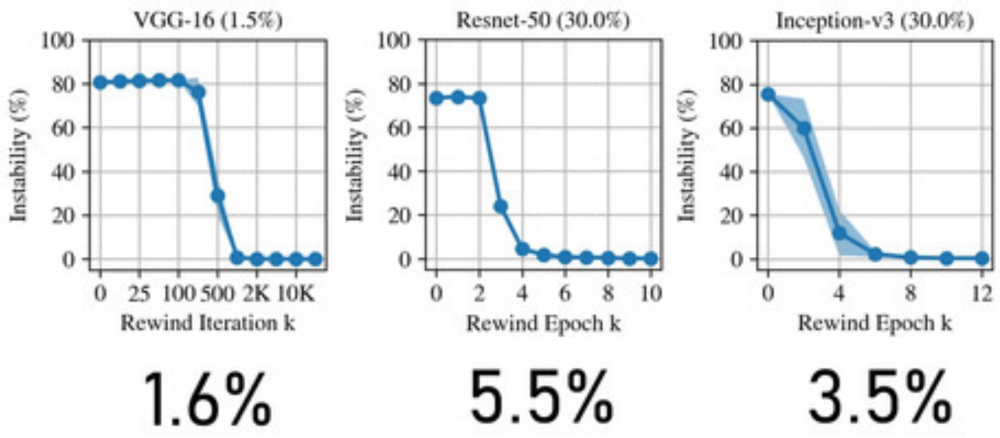


Rewind to Iteration k

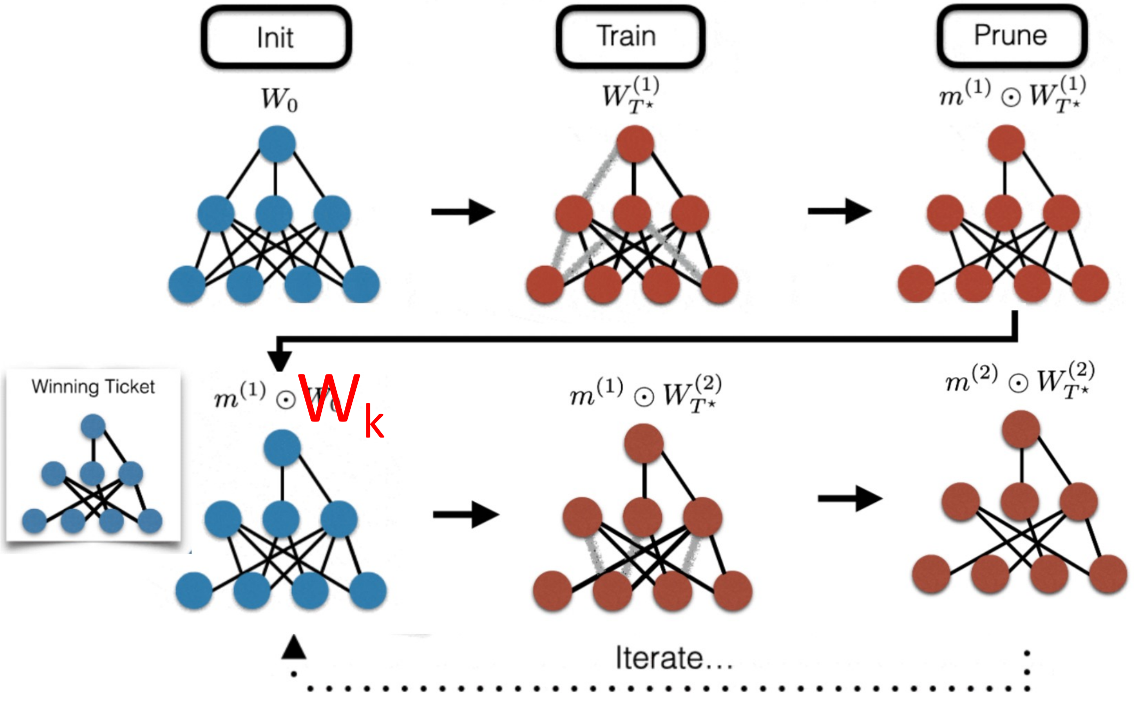


Lottery Ticket Hypothesis - IMP procedure [*(Frankle & Carbin, 2019)*]

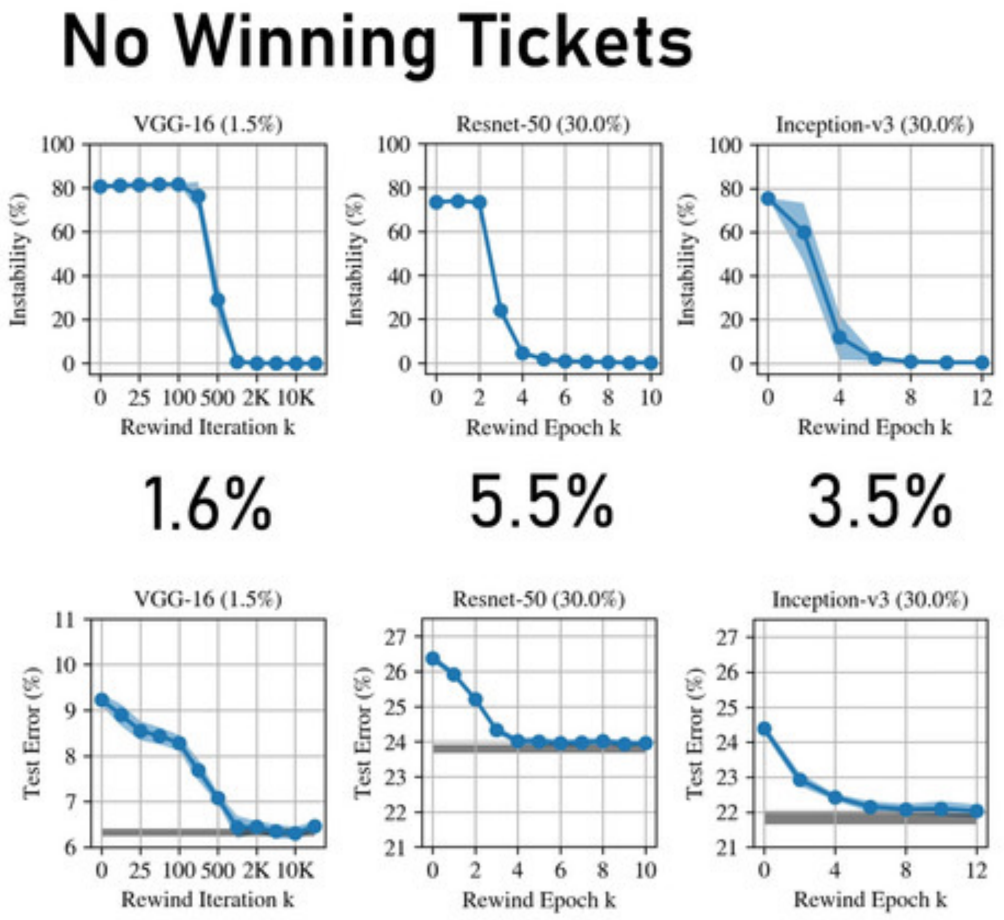
No Winning Tickets



Rewind to Iteration k



Lottery Ticket Hypothesis - IMP procedure [(Frankle & Carbin, 2019)](#)



LTH works again

Summary

Summary

- Propose the instability analysis, which can be used to study NN optimization dynamics
  - The LTH
  - Others optimization modules, e.g. the learning rate scheduler

Summary

- Propose the instability analysis, which can be used to study NN optimization dynamics
  - The LTH
  - Others optimization modules, e.g. the learning rate scheduler
- Solved one limitation of the original LTH
  - Sparse IMP subnetworks can be trained to full accuracy when they find the linearly connected iteration

Can LTH Become More Practical?

Can LTH Become More Practical?

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

ICLR 2020

Can LTH Become More Practical?

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

ICLR 2020

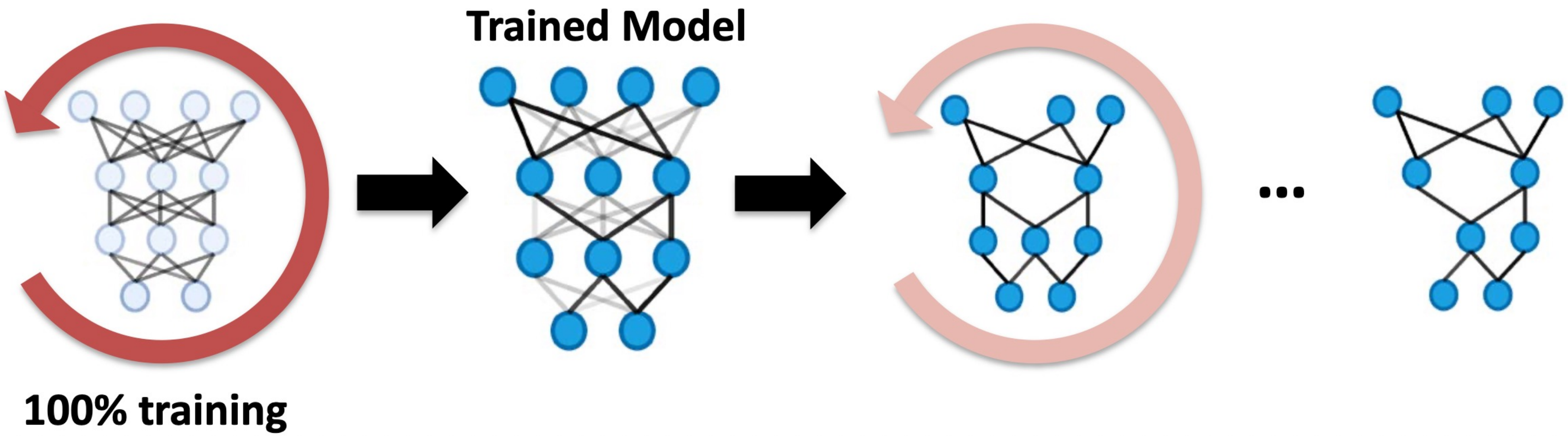
Yes. We don't need full training to find the winning tickets

Finding The Winning Tickets More Efficiently

Finding The Winning Tickets More Efficiently

IMP

Full training



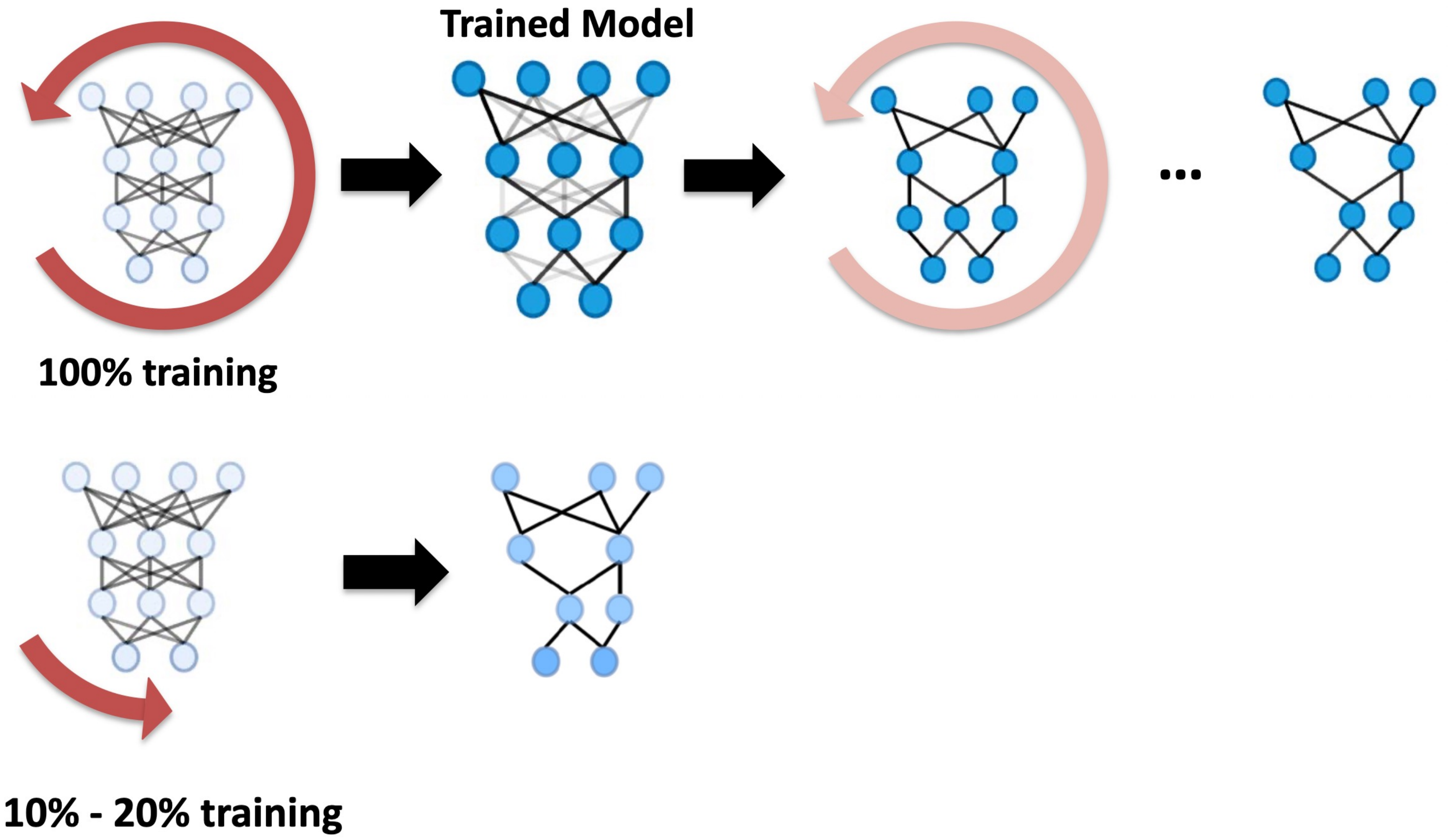
Finding The Winning Tickets More Efficiently

IMP

Full training

Early-bird (EB)

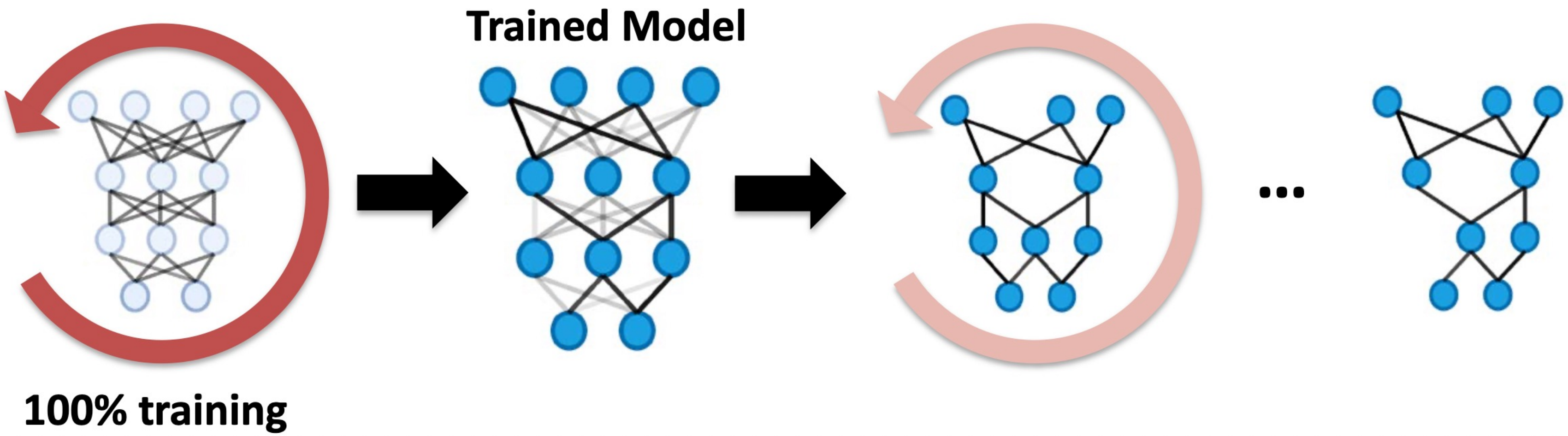
training



Finding The Winning Tickets More Efficiently

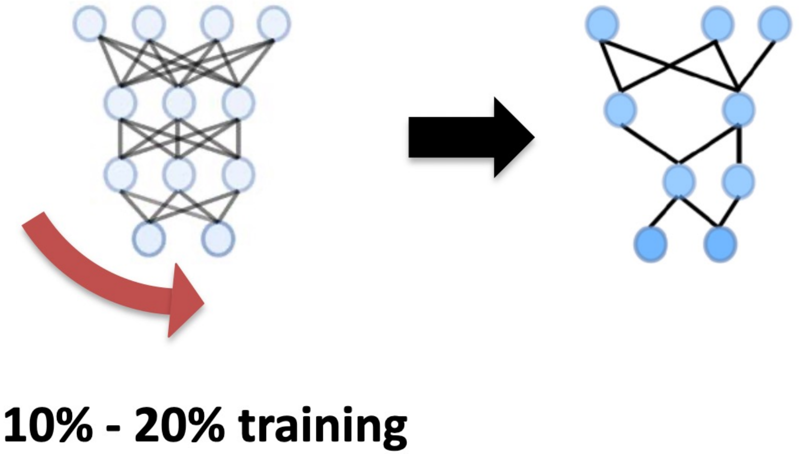
IMP

Full training



Early-bird (EB)

training

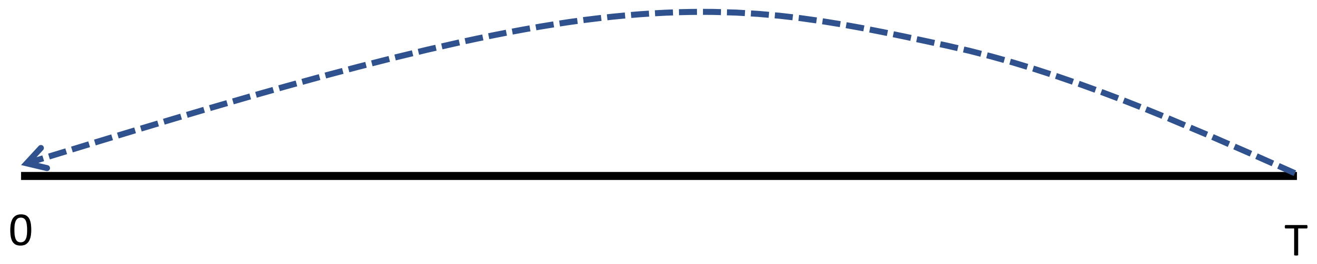


- Do EB tickets exist?
- How to find EB tickets?

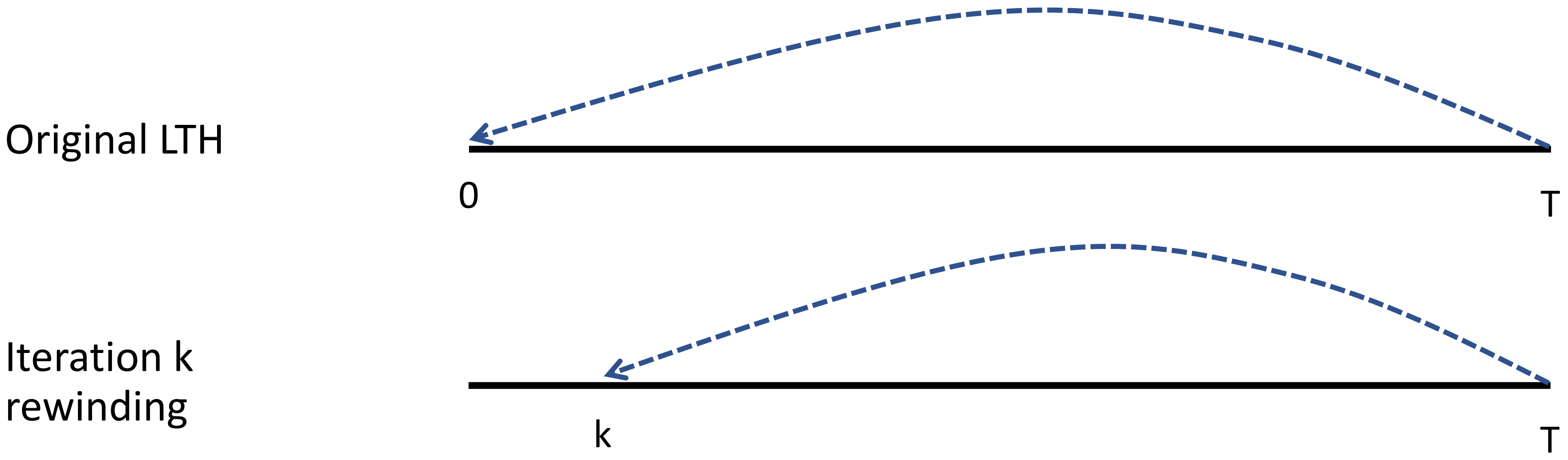
LTH vs. Iteration k Rewinding vs. EB

LTH vs. Iteration k Rewinding vs. EB

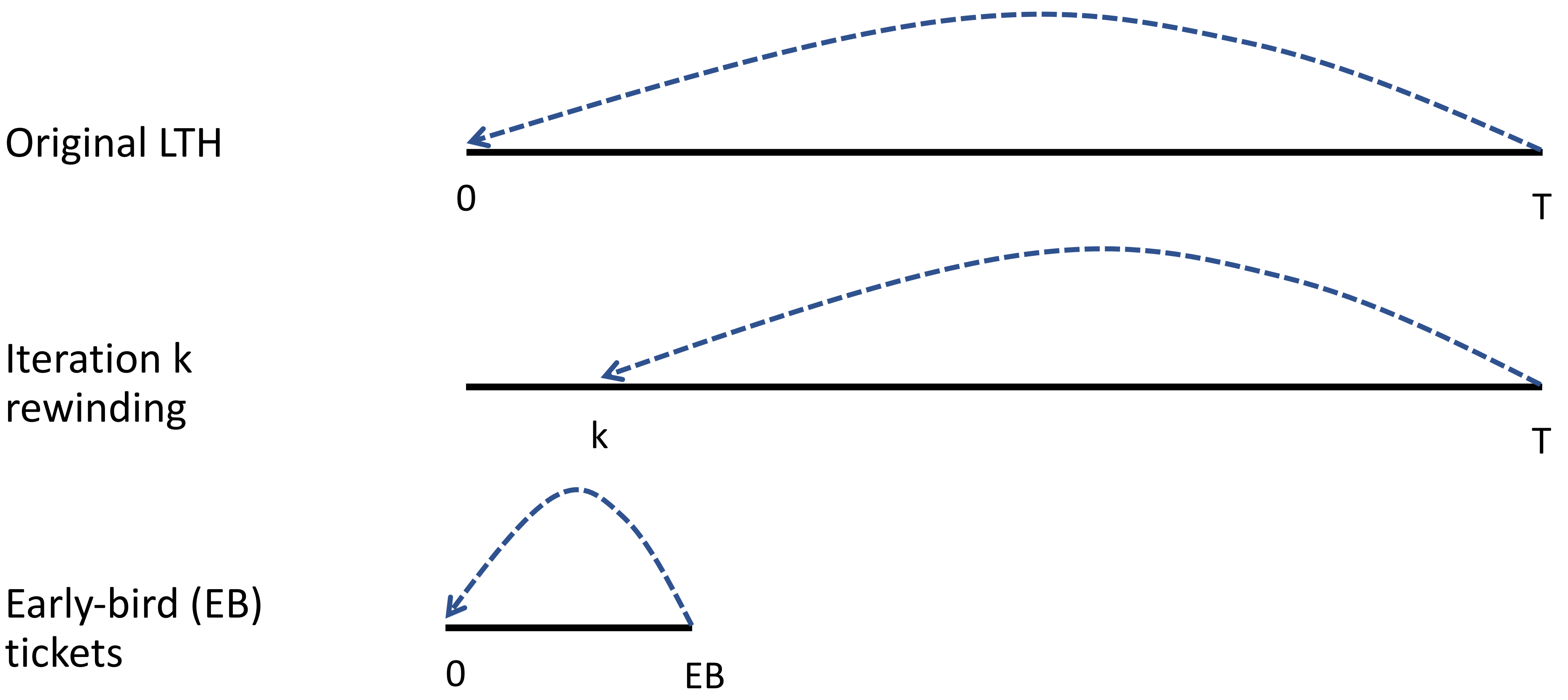
Original LTH



LTH vs. Iteration k Rewinding vs. EB



LTH vs. Iteration k Rewinding vs. EB



Existence of EB Tickets

Existence of EB Tickets

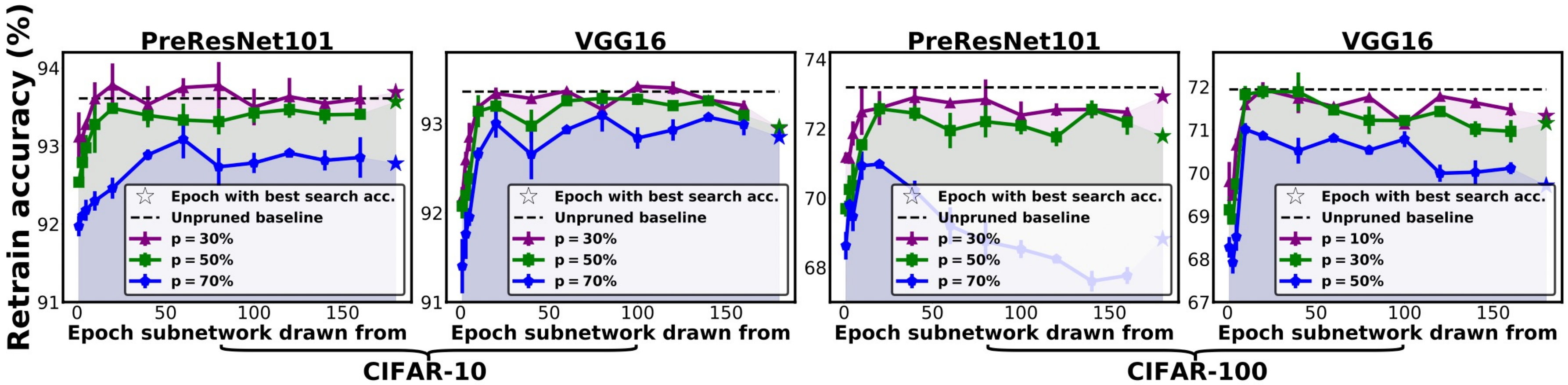


Figure 1: Retraining accuracy vs. epoch numbers at which the subnetworks are drawn, for both PreResNet101 and VGG16 on the CIFAR-10/100 datasets, where p indicates the channel pruning ratio and the dashed line shows the accuracy of the corresponding dense model on the same dataset, ☆ denotes the retraining accuracies of subnetworks drawn from the epochs with the best search accuracies, and error bars show the minimum and maximum of three runs.

Existence of EB Tickets

- Subnetworks drawn at early stages can achieve comparable or even better accuracy than the dense NN

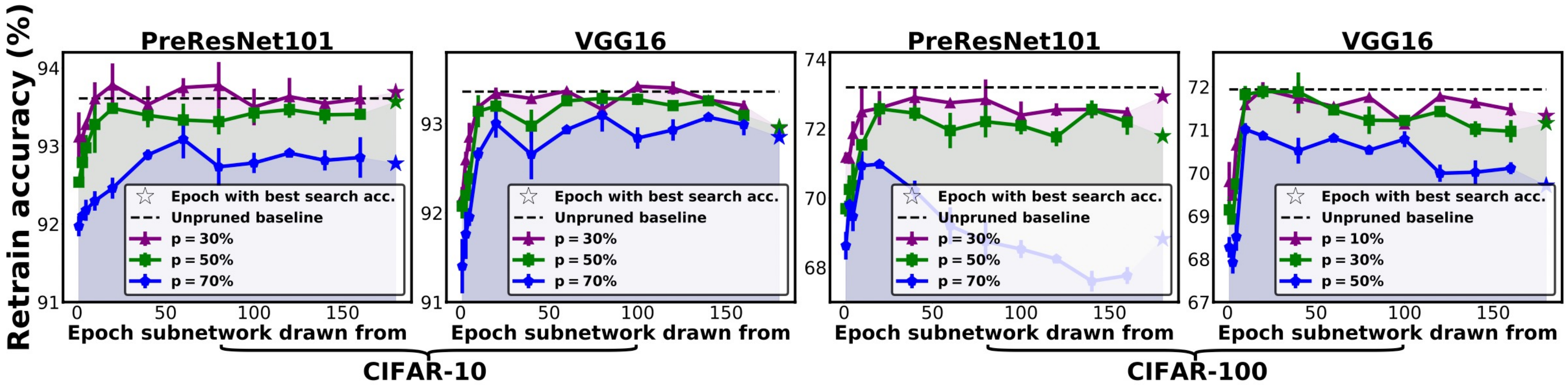


Figure 1: Retraining accuracy vs. epoch numbers at which the subnetworks are drawn, for both PreResNet101 and VGG16 on the CIFAR-10/100 datasets, where p indicates the channel pruning ratio and the dashed line shows the accuracy of the corresponding dense model on the same dataset, ☆ denotes the retraining accuracies of subnetworks drawn from the epochs with the best search accuracies, and error bars show the minimum and maximum of three runs.

Existence of EB Tickets

- Subnetworks drawn at early stages can achieve comparable or even better accuracy than the dense NN
- Identify NN connectivity patterns at later stage can be "overcooking"

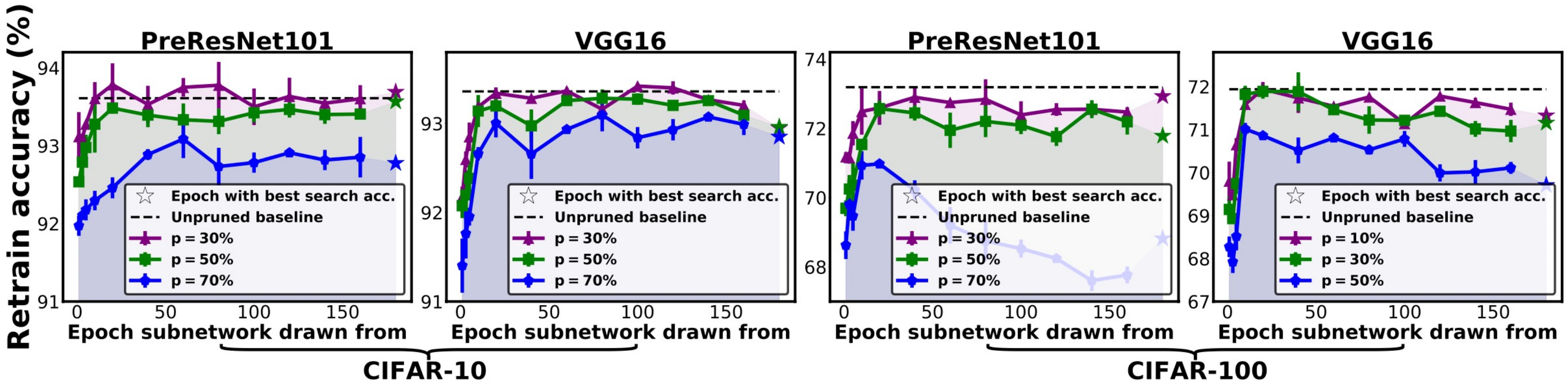
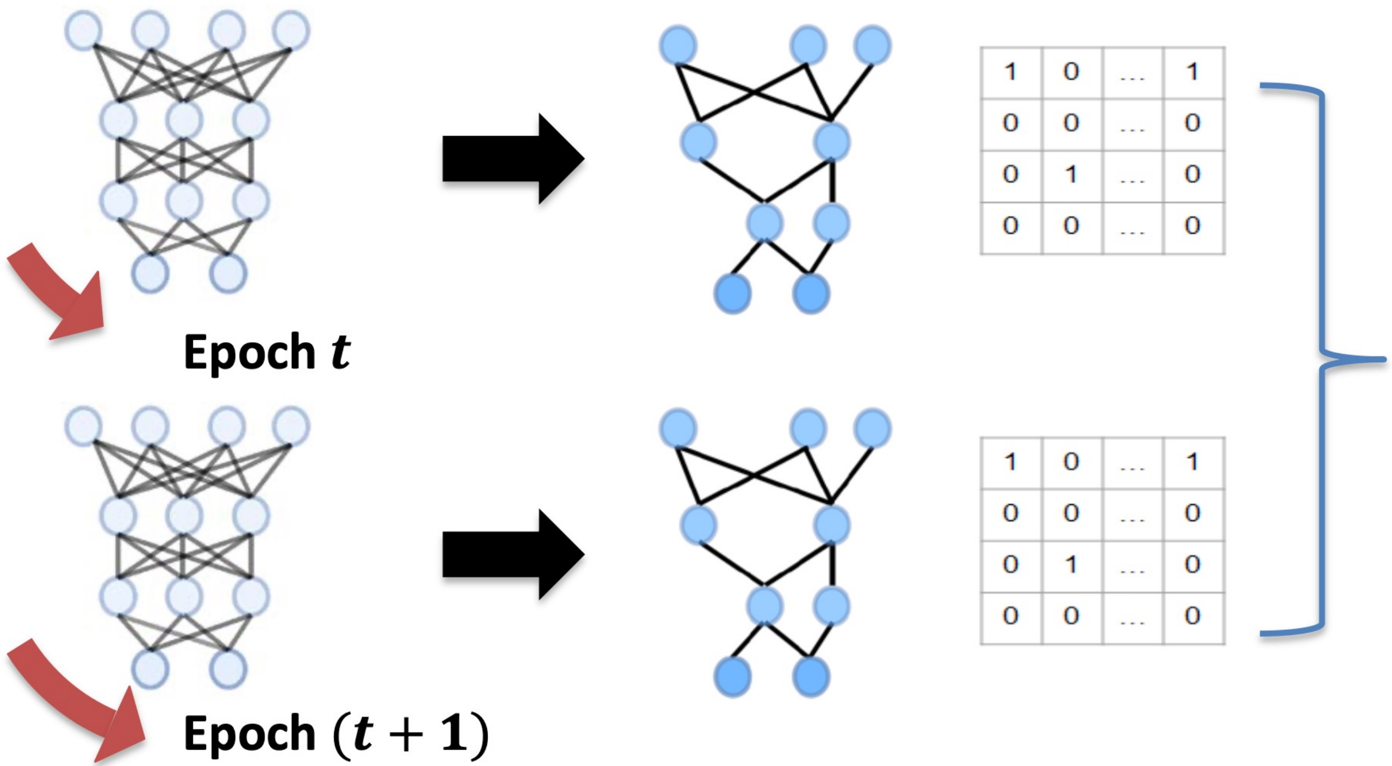


Figure 1: Retraining accuracy vs. epoch numbers at which the subnetworks are drawn, for both PreResNet101 and VGG16 on the CIFAR-10/100 datasets, where p indicates the channel pruning ratio and the dashed line shows the accuracy of the corresponding dense model on the same dataset, ☆ denotes the retraining accuracies of subnetworks drawn from the epochs with the best search accuracies, and error bars show the minimum and maximum of three runs.

Find EB Tickets with A Low-cost Detector

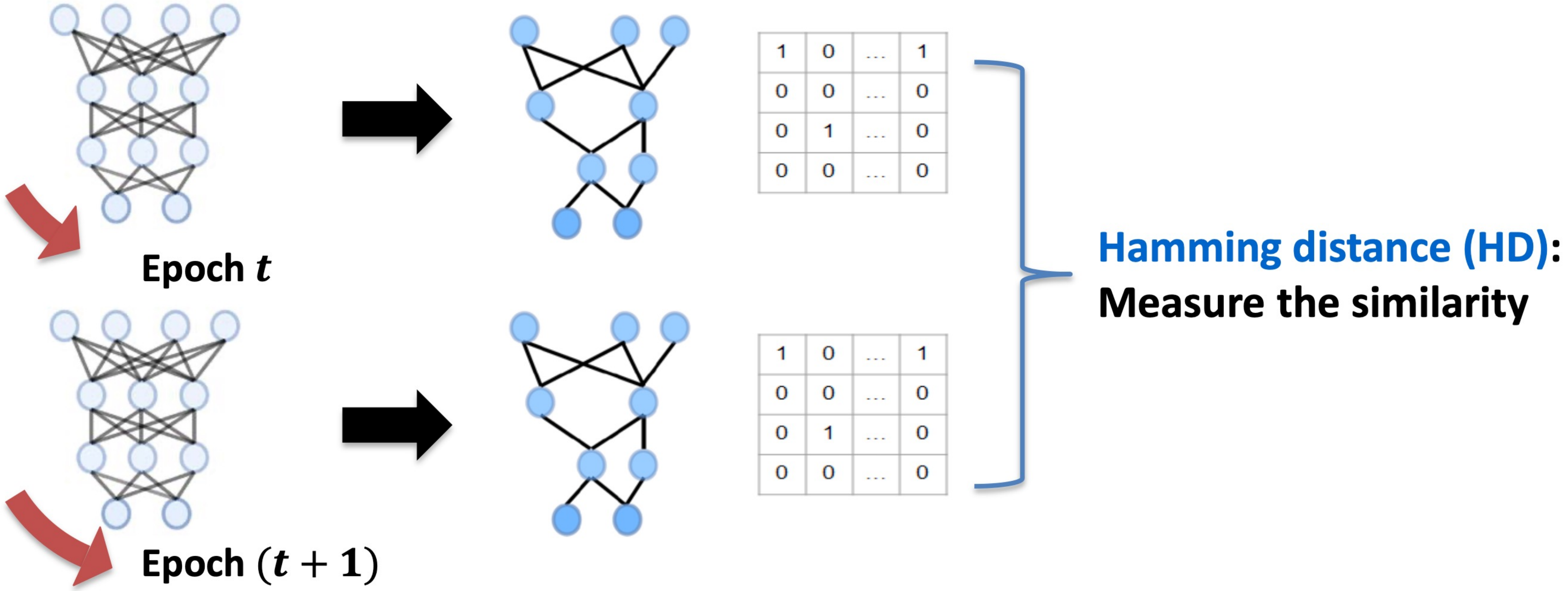
Find EB Tickets with A Low-cost Detector



Hamming distance (HD):
Measure the similarity

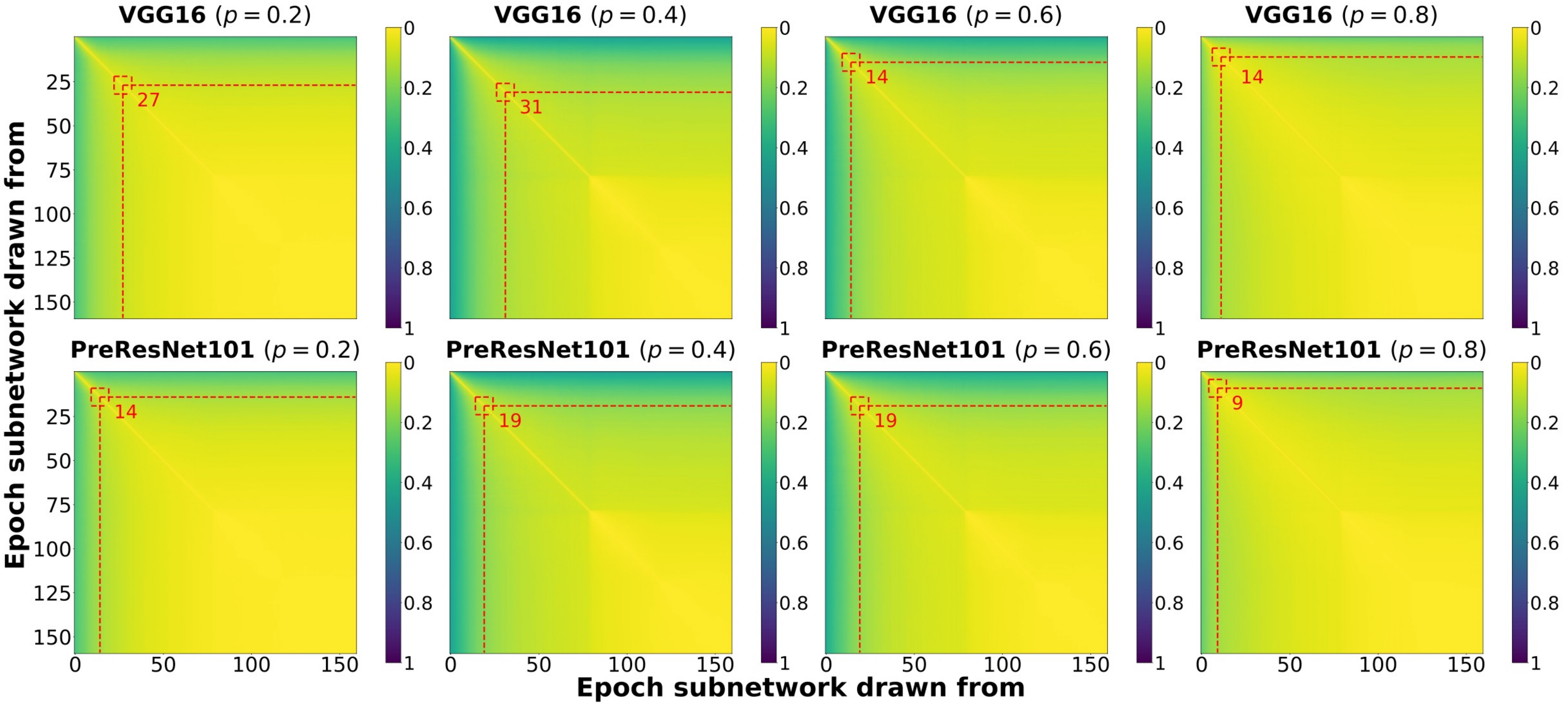
Find EB Tickets with A Low-cost Detector

- Draw EB Tickets when HDs of subnetworks between consecutive epochs are smaller than a specific threshold



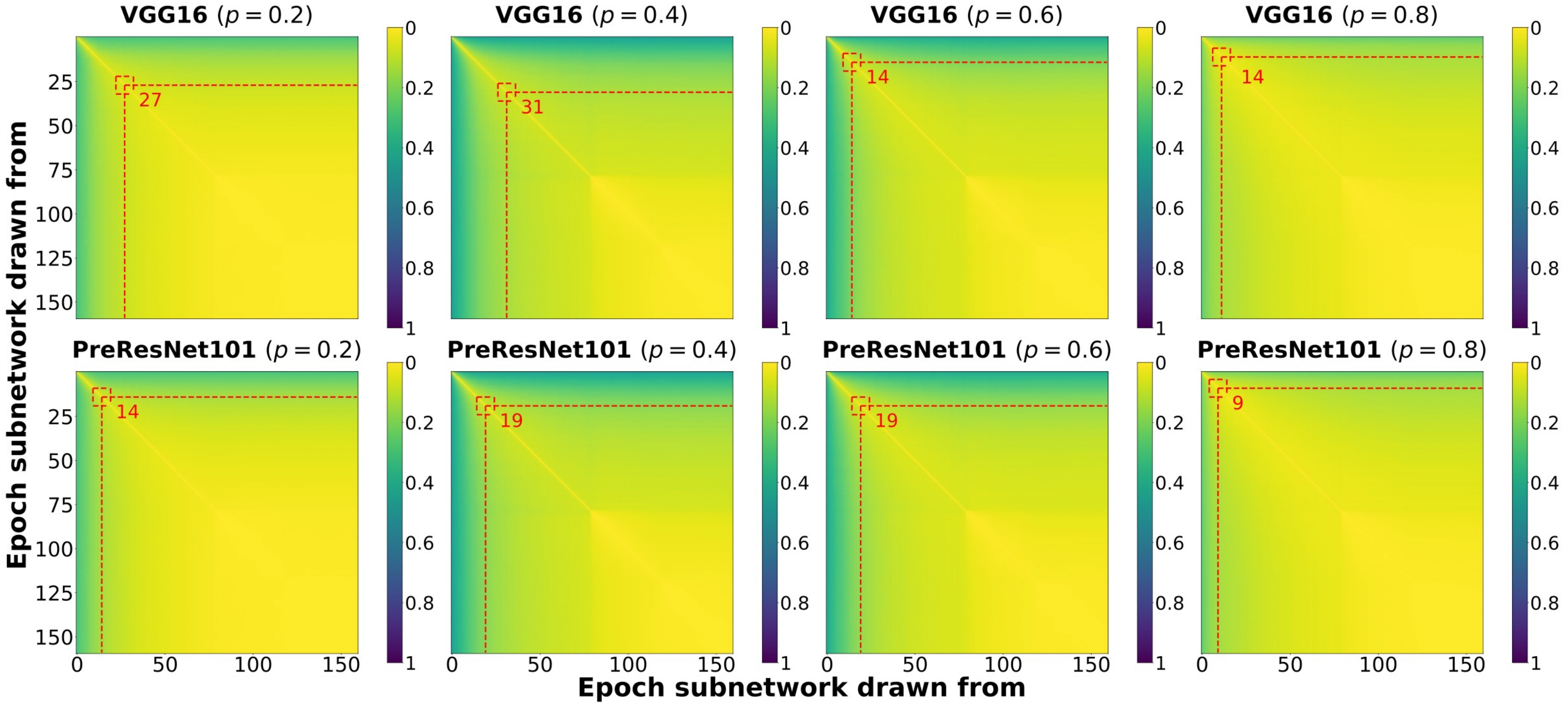
Find EB Tickets with A Low-cost Detector

Find EB Tickets with A Low-cost Detector



Find EB Tickets with A Low-cost Detector

- Green -> yellow: smooth distance



Efficient Training with EB Tickets

Efficient Training with EB Tickets

- Overall results on CIFAR-10/100 compared to the original LTH progressive pruning and training

Efficient Training with EB Tickets

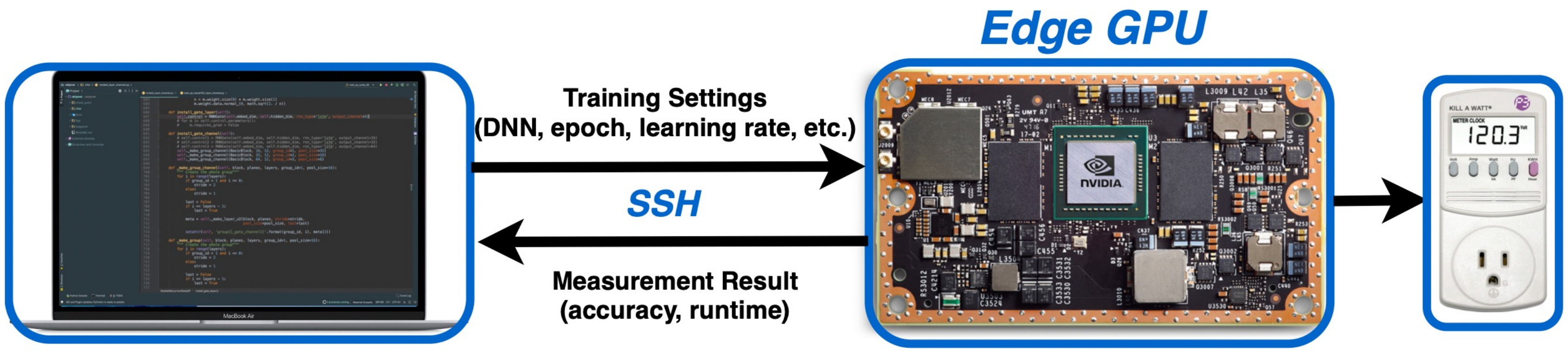
- Overall results on CIFAR-10/100 compared to the original LTH progressive pruning and training

Metrics	Overall Improvement
Total Training Energy Cost	**16.7% ~ 78.7%**
Accuracy	**-0.81% ~ +2.38%**

Efficient Training with EB Tickets

- Overall results on CIFAR-10/100 compared to the original LTH progressive pruning and training

Metrics	Overall Improvement
Total Training Energy Cost	**16.7% ~ 78.7%**
Accuracy	**-0.81% ~ +2.38%**



A power meter to measure energy cost

Summary

- Propose the EB Lottery Ticket, which shows that the progressive training in the original LTH can be stopped early
- EB tickets detector with hamming distance

LTH for Graphs

LTH for Graphs

A Unified Lottery Ticket Hypothesis for Graph Neural Networks

---

Tianlong Chen [*1] **Yongduo Sui** [*2] **Xuxi Chen** [1] **Aston Zhang** [3] **Zhangyang Wang** [1]

ICML 2021

LTH for Graphs

A Unified Lottery Ticket Hypothesis for Graph Neural Networks

Tianlong Chen [*1] **Yongduo Sui** [*2] **Xuxi Chen** [1] **Aston Zhang** [3] **Zhangyang Wang** [1]

ICML 2021

Prune a GNN and a graph jointly

The Pruning Problem on Graphs

The Pruning Problem on Graphs

- Pruning for vision and language are referring to NNs

The Pruning Problem on Graphs

- Pruning for vision and language are referring to NNs
- GNNs are usually not very large

The Pruning Problem on Graphs

- Pruning for vision and language are referring to NNs
- GNNs are usually not very large
- However, the graph itself can be huge
    - Training and inference on a dense graph can both be slow

The Pruning Problem on Graphs

- Pruning for vision and language are referring to NNs
- GNNs are usually not very large
- However, the graph itself can be huge
  - Training and inference on a dense graph can both be slow
- A well studied problem in the graph domain
  - Graph sampling
  - Graph sparsification
  - Graph coarsening

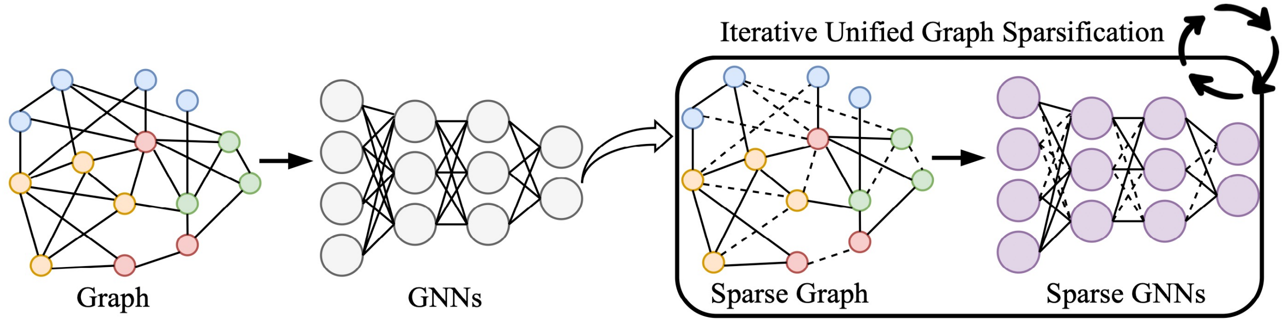
Generalize LTH to Graphs

Generalize LTH to Graphs

- Pruning both the GNN and the graph via IMP

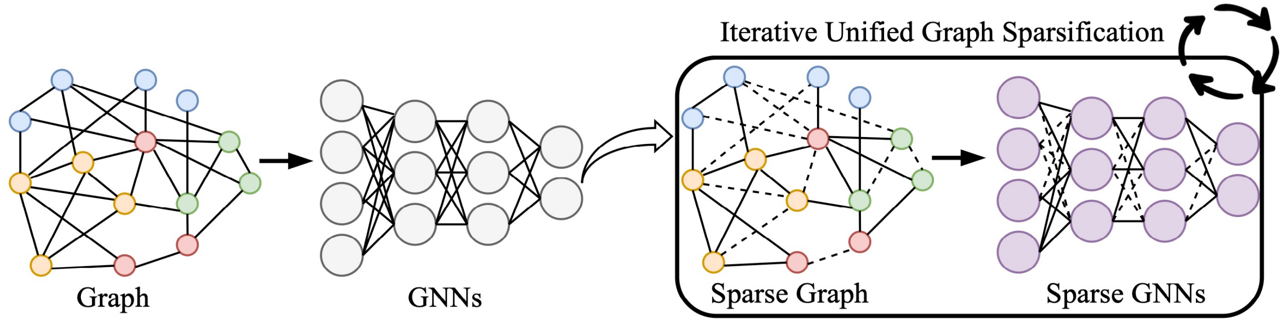
Generalize LTH to Graphs

- Pruning both the GNN and the graph via IMP



Generalize LTH to Graphs

- Pruning both the GNN and the graph via IMP



$$\mathcal{G} = \{\boldsymbol{A}, \boldsymbol{X}\}$$

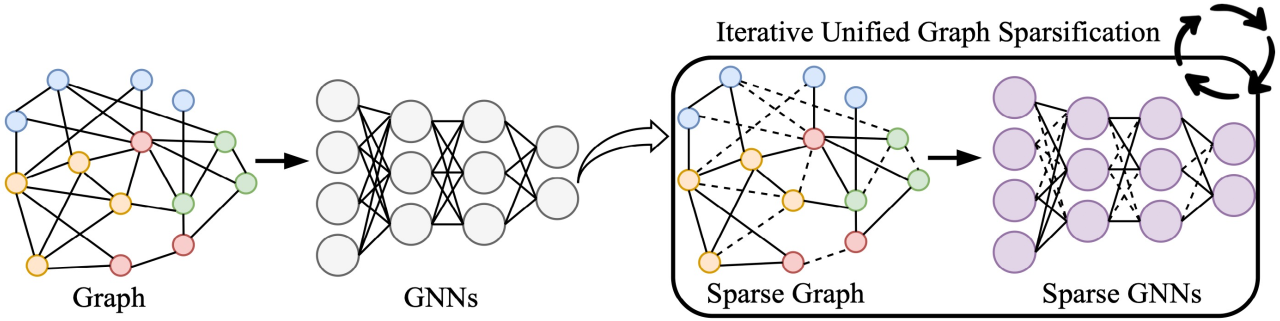
GNN $f(\cdot, \boldsymbol{\Theta})$

$$\{\boldsymbol{m}_g \odot \boldsymbol{A}, \boldsymbol{X}\}$$

$$f(\cdot, \boldsymbol{m}_\theta \odot \boldsymbol{\Theta})$$

Generalize LTH to Graphs

- Pruning both the GNN and the graph via IMP



$$\mathcal{G} = \{\boldsymbol{A}, \boldsymbol{X}\}$$

$$\text{GNN } f(\cdot, \boldsymbol{\Theta})$$

$$\{\boldsymbol{m}_g \odot \boldsymbol{A}, \boldsymbol{X}\}$$

$$f(\cdot, \boldsymbol{m}_\theta \odot \boldsymbol{\Theta})$$

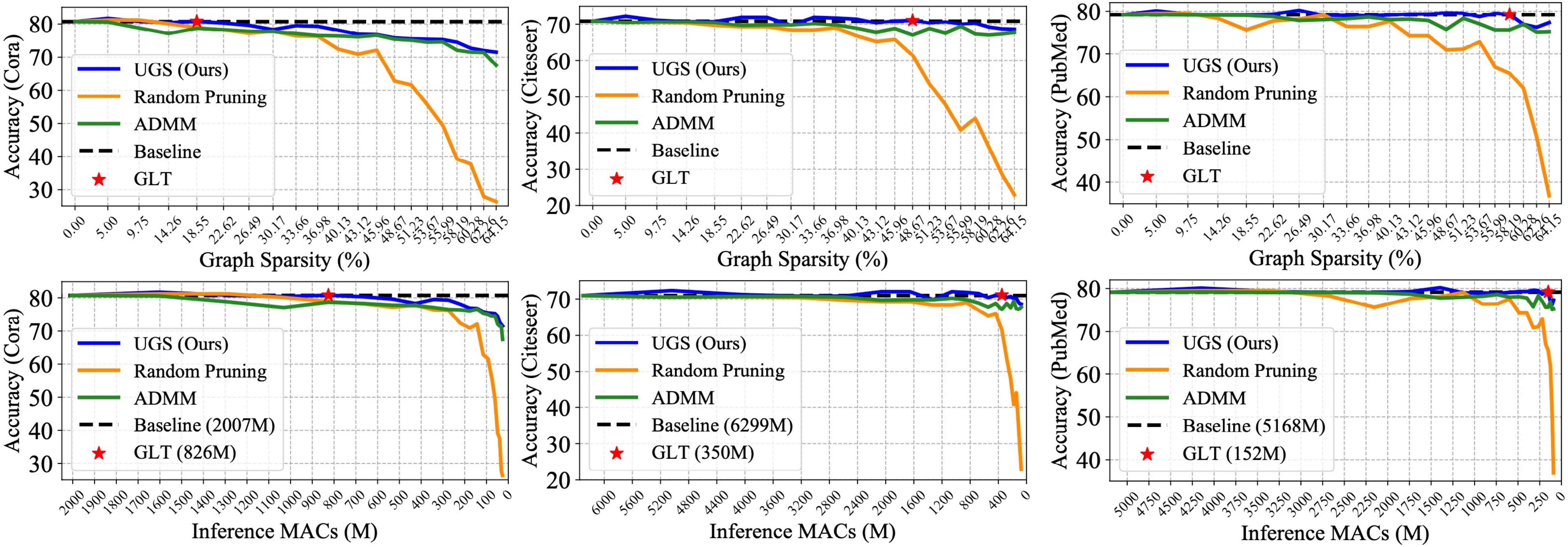
Algorithm 1 Unified GNN Sparsification (UGS)

Input: Graph $\mathcal{G} = \{\boldsymbol{A}, \boldsymbol{X}\}$, GNN $f(\mathcal{G}, \boldsymbol{\Theta}_0)$, GNN's initialization $\boldsymbol{\Theta}_0$, initial masks $\boldsymbol{m}_g^0 = \boldsymbol{A}$, $\boldsymbol{m}_\theta^0 = \boldsymbol{1} \in \mathbb{R}^{\|\Theta_0\|_0}$, Step size η, λ_g, and λ_θ.

Output: Sparsified masks $\boldsymbol{m}_g$ and $\boldsymbol{m}_\theta$

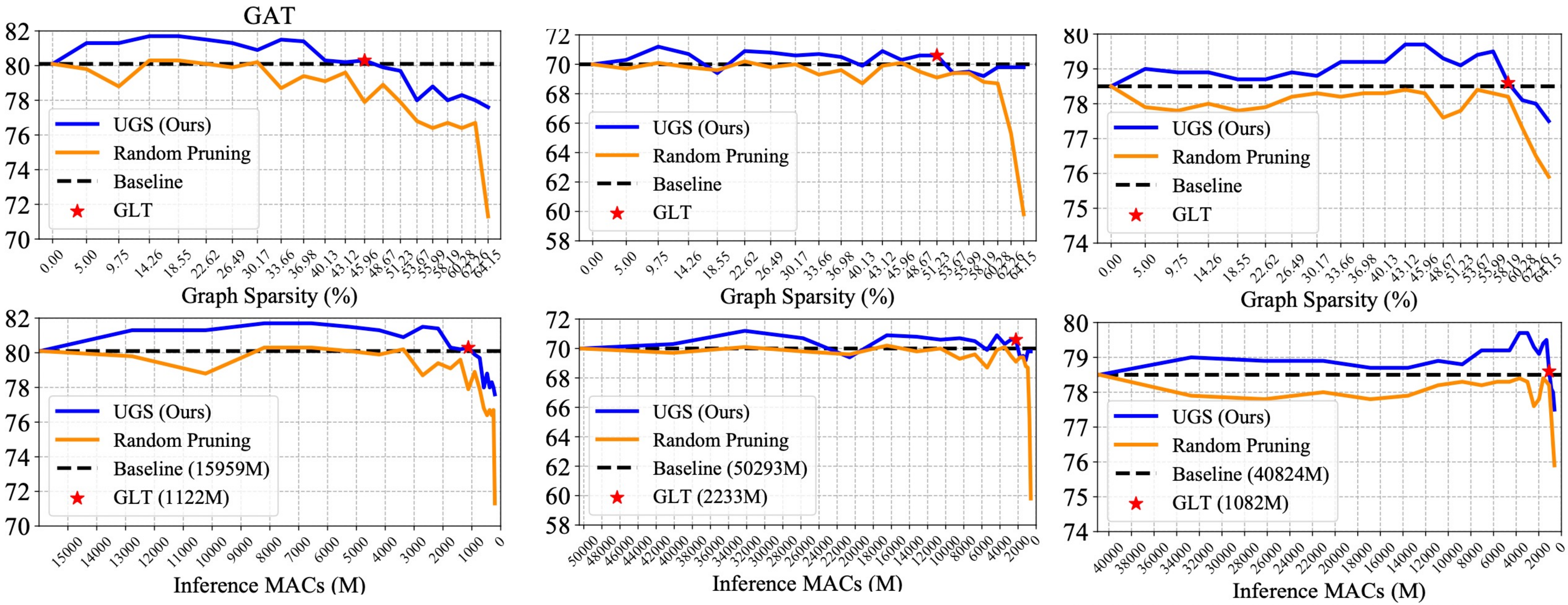
for iteration $i = 0, 1, 2, ..., \mathrm{N} - 1$ **do**
  Forward $f(\cdot, \boldsymbol{m}_\theta^i \odot \boldsymbol{\Theta}_i)$ with $\mathcal{G} = \{\boldsymbol{m}_g^i \odot \boldsymbol{A}, \boldsymbol{X}\}$ to compute the loss $\mathcal{L}_{\mathrm{UGS}}$ in Equation 3.
  Backpropagate to update $\boldsymbol{\Theta}_{i+1} \leftarrow \boldsymbol{\Theta}_i - \eta \nabla_{\boldsymbol{\Theta}_i} \mathcal{L}_{\mathrm{UGS}}$.
  Update $\boldsymbol{m}_g^{i+1} \leftarrow \boldsymbol{m}_g^i - \lambda_g \nabla_{\boldsymbol{m}_g^i} \mathcal{L}_{\mathrm{UGS}}$.
  Update $\boldsymbol{m}_\theta^{i+1} \leftarrow \boldsymbol{m}_\theta^i - \lambda_\theta \nabla_{\boldsymbol{m}_\theta^i} \mathcal{L}_{\mathrm{UGS}}$.
end for
Set $p_g = 5\%$ of the lowest magnitude values in $\boldsymbol{m}_g^{\mathrm{N}}$ to 0 and others to 1, then obtain $\boldsymbol{m}_g$.
Set $p_\theta = 20\%$ of the lowest magnitude values in $\boldsymbol{m}_\theta^{\mathrm{N}}$ to 0 and others to 1, then obtain $\boldsymbol{m}_\theta$.

GCN on Cora, Citeseer, and PubMed

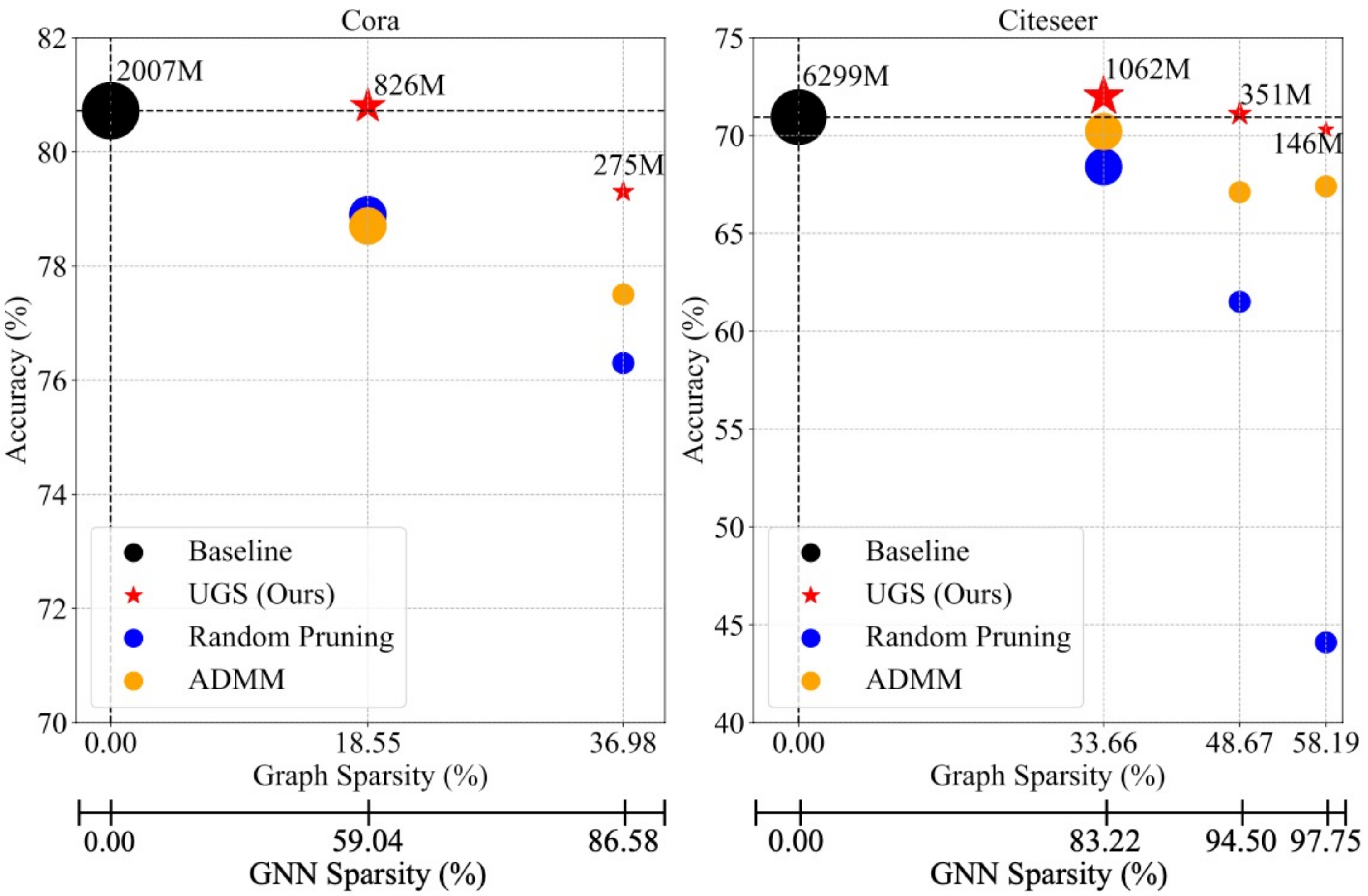


GAT on Cora, Citeseer, and PubMed

- GAT is more amenable to pruning



Performance Summary



Node size represents Inference multiply–accumulate operations (MACs)

Reference

- Han et al. (2015) Learning both Weights and Connections for Efficient Neural Networks https://arxiv.org/pdf/1506.02626.pdf
- Frankle & Carbin. (2019) The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks https://arxiv.org/pdf/1803.03635.pdf
- Frankle et al. (2020) Linear Mode Connectivity and the Lottery Ticket Hypothesis https://arxiv.org/pdf/1912.05671.pdf
- You et al. (2020) Drawing Early-bird Tickets: Towards More Efficient Training Of Deep Networks https://openreview.net/pdf?id=BJxsrgStvr
- Chen et al. (2021) A Unified Lottery Ticket Hypothesis for Graph Neural Networks https://arxiv.org/pdf/2102.06790.pdf
- You et al. (2022) Early-Bird GCNs: Graph-Network Co-Optimization Towards More Efficient GCN Training and Inference via Drawing Early-Bird Lottery Tickets https://arxiv.org/pdf/2103.00794.pdf

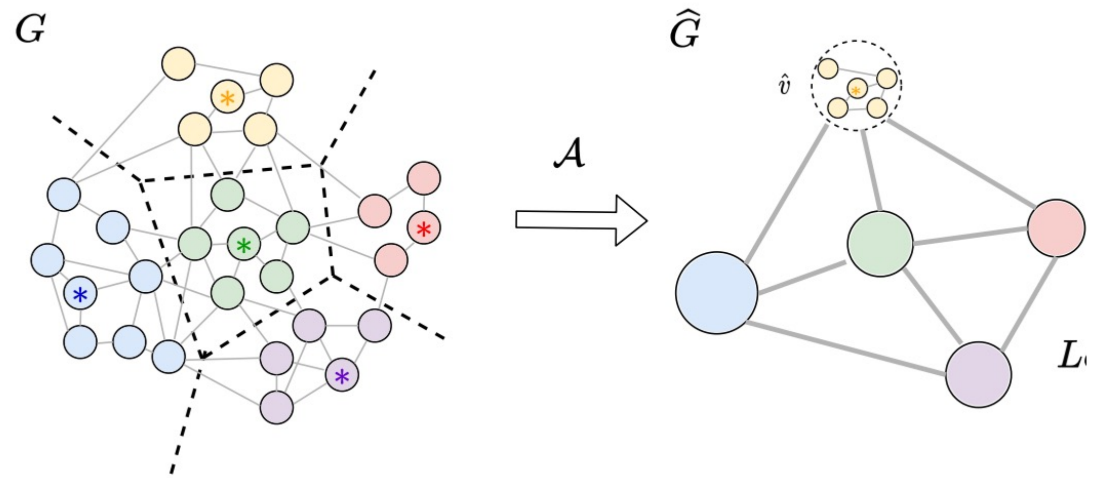
Appendix

Tell the Difference

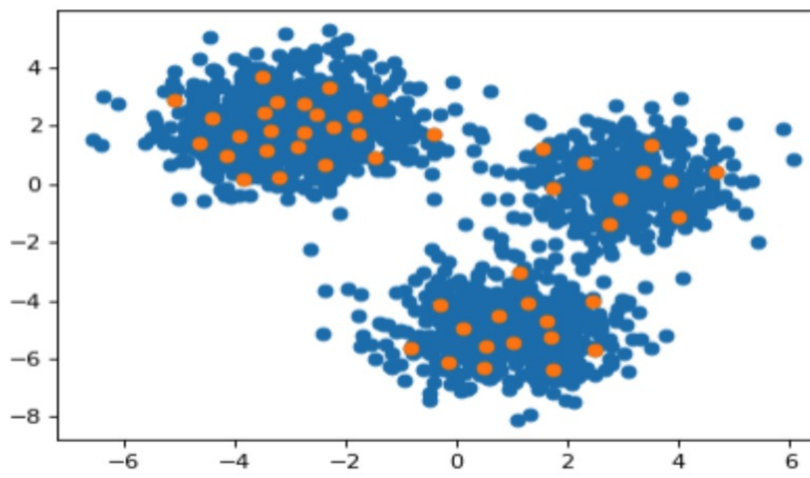
- Graph sampling
- Graph augmentation
- Graph pruning
- Graph sparsification
- Graph coarsening
- Graph pooling
- Graph structure learning
- Graph coreset
- Graph condensation
- Graph lottery ticket

$$\mathbf{G} = \{\mathbf{X}, \mathbf{A}\} \quad \mathbf{A} \in \mathbf{R}^{\mathbf{N},\mathbf{N}} \quad \mathbf{X} \in \mathbf{R}^{\mathbf{N},\mathbf{d}} \quad \Rightarrow \quad \mathbf{G}' = \{\mathbf{X}', \mathbf{A}'\} \quad \mathbf{A}' \in \mathbf{R}^{\mathbf{N}',\mathbf{N}'} \quad \mathbf{X}' \in \mathbf{R}^{\mathbf{N}',\mathbf{d}'}$$

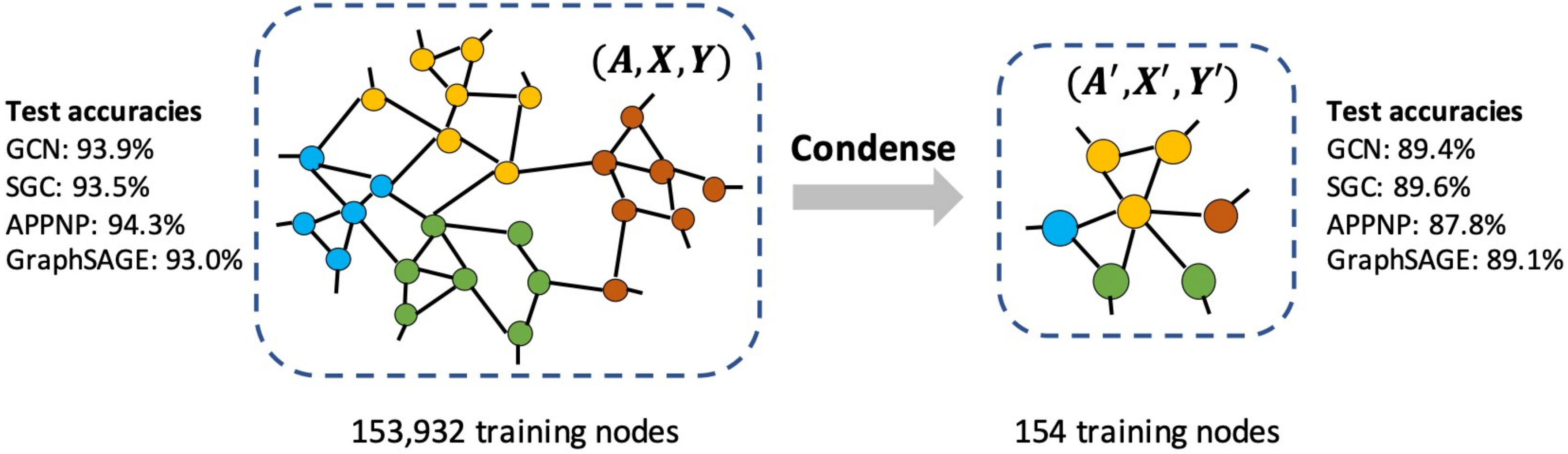
Method	Nodes	Edges	Features (the same nodes)	Learning?	Is it the goal?	Note
Sampling	N’ << N	A’ ⊆ A	X’ = X	No	No	A pre-learning process, usually locally
Augmentation	N’ ≠ N	A’ ≠ A	X’ ≠ X	No	No	A pre-learning process
Pruning	N’ < N	A’ ⊆ A	X’ ⊆ X	Yes	Yes	Usually used for the NN graph
Sparsification	N’ = N	A’ ⊆ A	X’ = X	Yes/No	Yes/No	
Coarsening /Pooling	N’ < N	A’ ⊆ A	X’ ≠ X	Yes	Yes/No	Coarsening is usually for node-level problems Pooling is usually for graph-level problems
Structure learning	N’ = N	A’ ≠ A	X’ = X	Yes	No?	1. **A** may not be given 2. Learning maybe jointly with a GNN
Coreset	N’ << N	A’ = A	X’ = X	Yes	No	An importance score may be learned for each node
Condensation	N’ << N	A’ ≠ A	X’ ≠ X	Yes	Yes	G’ is a brand-new generated graph
Lottery ticket (UGS)	N’ = N	A’ ⊆ A	X’ = X	Yes	Yes/No	1. Verifying the existence of the LT 2. Jointly learned with a GNN



Coarsening



Coreset



Condensation

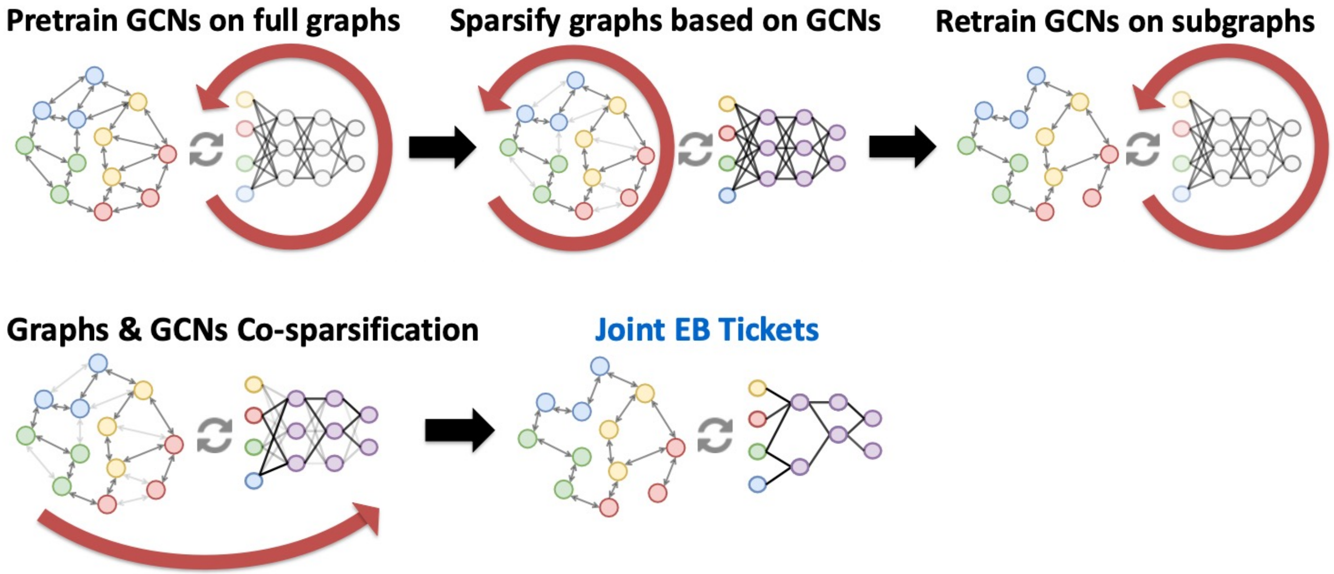
EB on GCN

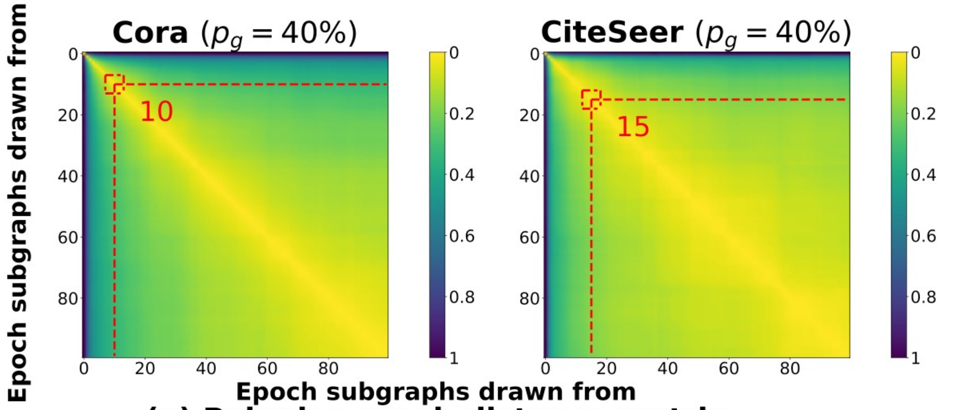
Early-Bird GCNs: Graph-Network Co-Optimization Towards More Efficient GCN Training and Inference via Drawing Early-Bird Lottery Tickets

Haoran You, Zhihan Lu, Zijian Zhou, Yonggan Fu, Yingyan Lin

Department of Electrical and Computer Engineering, Rice University
{haoran.you, zl55, zjzhou, yf22, yingyan.lin}@rice.edu

AAAI 2022





(a) Pairwise graph distance matrix