



Recent Advances on Neural Network Pruning at Initialization



Huan Wang¹



Can Qin¹



Yue Bai¹



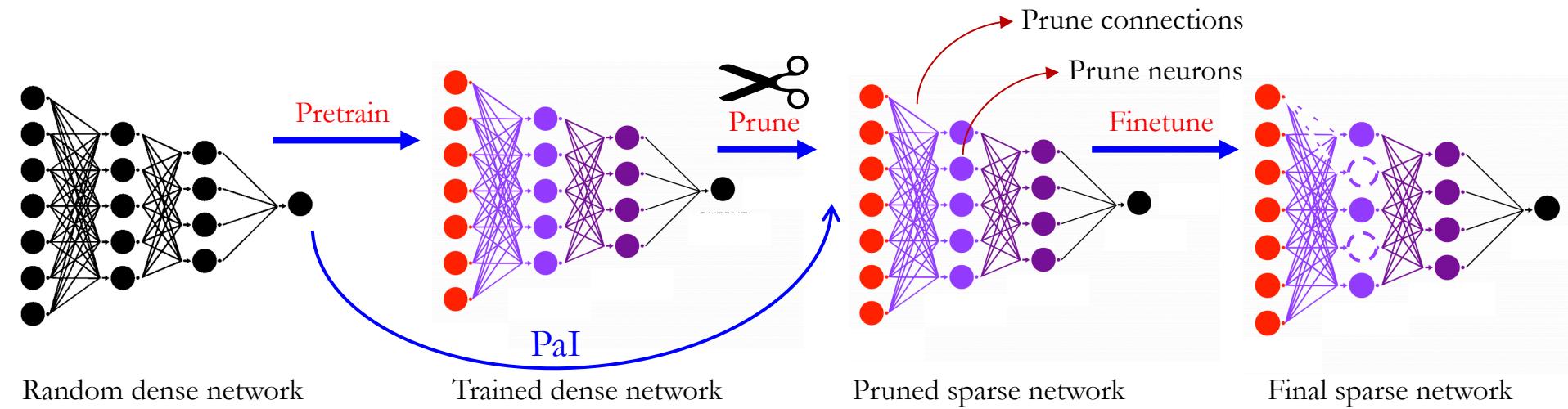
Yulun Zhang²



Yun Fu¹

¹Northeastern University, USA ²ETH Zürich, Switzerland

Background: What is Pruning at Initialization (PaI)?



The “conventional” 3-step pruning pipeline: **Pruning after training (PaT)**

Pruning at initialization (PaI) **saves the first step**, while maintaining comparable performance to PaT.

Background: Why We Need this Survey

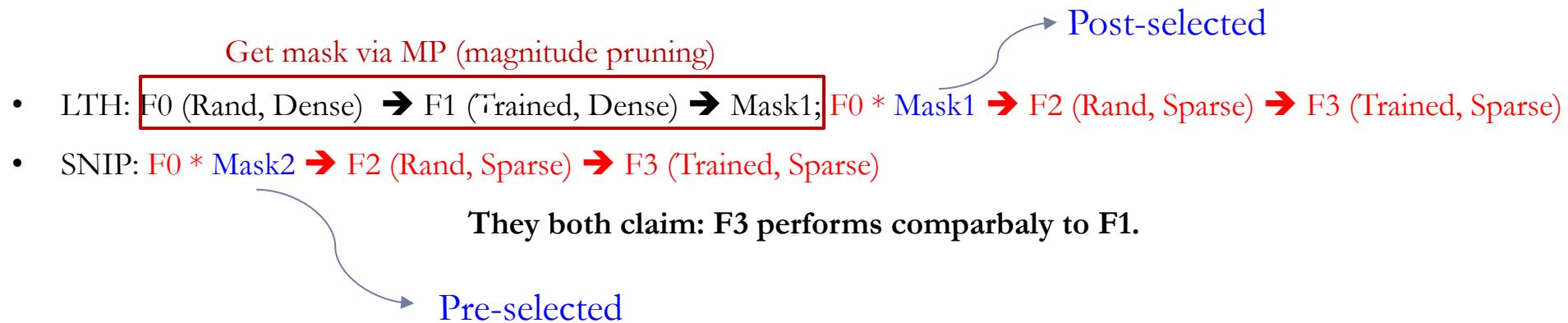
- 1993-TNN-[Pruning Algorithms -- A survey](#)
- 2017-Proceedings of the IEEE-[Efficient Processing of Deep Neural Networks: A Tutorial and Survey](#)
- 2018-FITEE-[Recent Advances in Efficient Computation of Deep Convolutional Neural Networks](#)
- 2018-IEEE Signal Processing Magazine-[Model compression and acceleration for deep neural networks: The principles, progress, and challenges](#)
- 2020-MLSys-[What is the state of neural network pruning](#)
- 2019.02-[The State of Sparsity in Deep Neural Networks](#)
- 2020-Proceedings of the IEEE-[Model Compression and Hardware Acceleration for Neural Networks: A Comprehensive Survey](#)
- 2021-JMLR-[Sparsity in Deep Learning: Pruning and growth for efficient inference and training in neural networks](#)
- 2021.6-[Efficient Deep Learning: A Survey on Making Deep Learning Models Smaller, Faster, and Better](#)

Existing pruning surveys. (src: [EfficientDNNs](#))

- They mainly focus on pruning after training (PaT).
- Only one recent survey (2021-JMLR-Sparsity in Deep Learning) scratches the topic of PaI.
- We intend to provide **the first comprehensive and systematic coverage concentrated on PaI.**

History Sketch: Debut of PaI (2 papers)

- ❑ Lottery Ticket Hypothesis (LTH) [Frankle and Carbin, ICLR, 2019] (**Best paper award**)
- ❑ Single-shot Network Pruning (SNIP) [Lee et al., ICLR, 2019]



Non-trivially sparse networks can be trained to full accuracy in isolation.
“PaI = Dense”

PaI Universe at Present

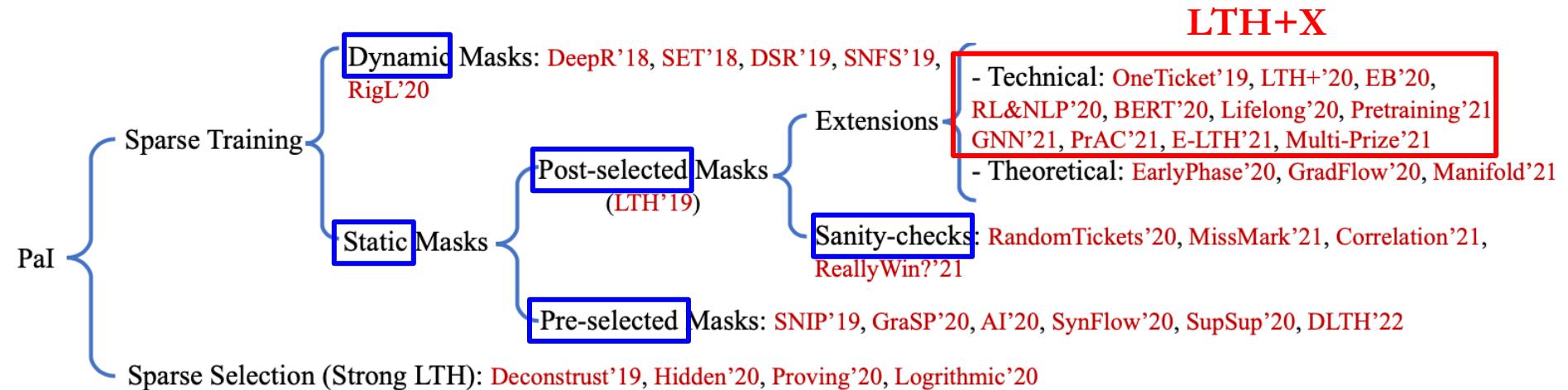


Figure 1: Overview of pruning at initialization (PaI) approaches, classified into two general groups: *sparse training* and *sparse selection*. For readability, references are omitted in this figure but the paper abbreviations (right beside the abbreviation is the year when the paper appeared). Please see Sec. 3 for detailed introductions of them. Due to limited length, this paper only outlines the primary methods, see *full* collection at <https://github.com/mingsun-tse/awesome-pruning-at-initialization>.

PaI universe tree presented in our paper. Note, **two major branches** of PaI: **sparse training** and **sparse selection**

Next, we talk about the **four classic topics** (or steps) in pruning and highlight how PaI makes any difference in each of them.

Classic Topics in Pruning: #1 Sparsity Structure

What to prune?

Terms: “Granularity of Sparsity; Fine-grained Pruning, Coarse-grained pruning”

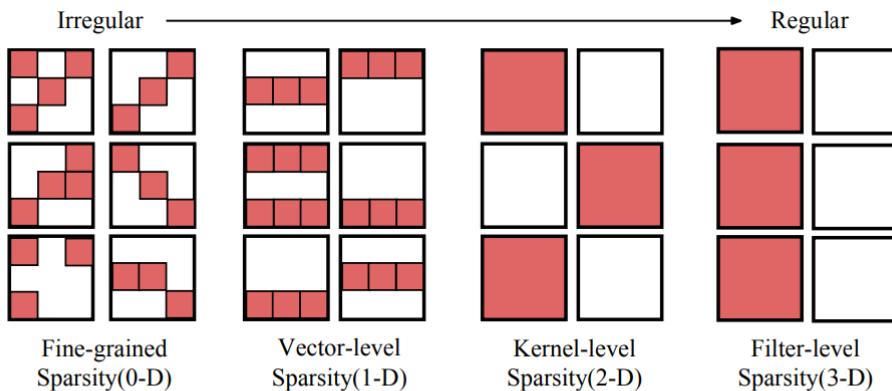


Figure 1. Different structure of sparsity in a 4-dimensional weight tensor. Regular sparsity makes hardware acceleration easier.

Illustration of different sparsity structures

[Mao et al., CVPRw, 2017]

- Same sparsity ratio, **more fine-grained, less performance drop.**
- Using which granularity is up to specific application scenario (**unstructured pruning: compression; structured pruning: acceleration**).

PaI Case

- PaI focuses on **unstructured pruning** (unlike PaT, which focuses on structured pruning now).
- LTH on structured pruning is still an open question.

Classic Topics in Pruning: #2 Pruning Ratio

How many to prune (for each layer)?

- **Indirect:** Regularization-based pruning
 - Larger penalty, more sparsity. E.g., SSL [Wen et al., NIPS, 2016])
- **Pre-defined** (more popular now🔥):
 - Global: Given a global sparsity ratio, layer-wise sparsity is “learned” by the pruning algorithm
 - Local: Given a global sparsity ratio, layer-wise sparsity is pre-defined

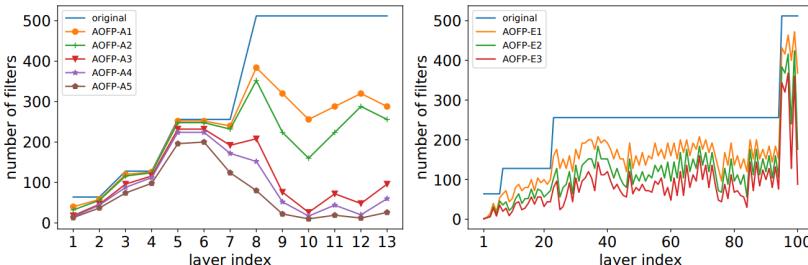


Figure 4. Layer width of pruned models. Left: VGG on CIFAR-10. Right: ResNet-152 on ImageNet (only the pruned layers).

Example of specifying global sparsity
AOFP [Ding et al., ICML, 2019]

Example of specifying local sparsity
GReg [Wang et al., ICLR, 2021]

Table 6: Pruning ratio summary.

| Dataset | Network | Speedup | Pruned top-1 accuracy (%) | Pruning ratio |
|----------|----------|---------|---------------------------|-----------------------------|
| CIFAR10 | ResNet56 | 2.55× | 93.36 | [0, 0.75, 0.75, 0.32] |
| CIFAR100 | VGG19 | 8.84× | 67.56 | [0:0, 1-15:0.70] |
| ImageNet | ResNet34 | 1.32× | 73.44 | [0, 0.50, 0.60, 0.40, 0]* |
| ImageNet | ResNet50 | 1.49× | 76.24 | [0, 0.30, 0.30, 0.30, 0.14] |
| ImageNet | ResNet50 | 2.31× | 75.16 | [0, 0.60, 0.60, 0.60, 0.21] |
| ImageNet | ResNet50 | 2.56× | 74.75 | [0, 0.74, 0.74, 0.60, 0.21] |
| ImageNet | ResNet50 | 3.06× | 73.50 | [0, 0.68, 0.68, 0.68, 0.50] |

* In addition to the pruning ratios, several layers are skipped, following the setting of L_1 (pruned-B) Li et al. (2017). Specifically, we refer to the implementation of Liu et al. (2019) at <https://github.com/Eric-mingjie/rethinking-network-pruning/tree/master/imagenet/l1-norm-pruning>.

Classic Topics in Pruning: #2 Pruning Ratio

How many to prune (for each layer)?

- **Indirect:** Regularization-based pruning
 - Larger penalty, more sparsity. E.g., SSL [Wen et al., NIPS, 2016])
- **Pre-defined:** (more popular now🔥):
 - Global: Given a global sparsity ratio, layer-wise sparsity is “learned” by the pruning algorithm
 - Local: Given a global sparsity ratio, layer-wise sparsity is pre-defined

PaI Case

- PaI prefers **global** sparsity. Only a few works use pre-defined layer-wise sparsities.

Classic Topics in Pruning: #3 Pruning Criterion

By what criterion, we consider a weight important or unimportant?

- ❑ LTH: Iterative Magnitude Pruning (IMP), SNIP (“connection sensitivity”)
- ❑ Most popular: Magnitude Pruning (L1-norm, L2-norm, etc.)

and ResNet-50 trained on ImageNet. Across thousands of experiments, we demonstrate that complex techniques (Molchanov et al., 2017; Louizos et al., 2017b) shown to yield high compression rates on smaller datasets perform inconsistently, and that simple magnitude pruning approaches achieve comparable or better results. Based on insights from our experiments, we achieve a new state-of-the-art sparsity-accuracy trade-off for ResNet-50 using only magnitude pruning. Additionally, we repeat the experiments performed by Frankle & Carbin (2018) and Liu et al. (2018) at scale and show that unstructured sparse architectures learned through pruning cannot be trained

MP is the SOTA (unstructured) pruning algorithm.
-- [Gale et al., Arxiv, 2019]

This statement is still true (in a large part) today!
MP or its variants are strong competitors, either unstructured pruning or structured pruning, especially for non-extreme sparsities.

Abstract crop of [Gale et al., Arxiv, 2019]

Classic Topics in Pruning: #3 Pruning Criterion

| Method | Pruning criterion |
|------------------------|--|
| Skeletonization (1989) | $-\nabla_{\mathbf{w}} \mathcal{L} \odot \mathbf{w}$ |
| OBD (1990) | $\text{diag}(H)\mathbf{w} \odot \mathbf{w}$ |
| Taylor-FO (2019) | $(\nabla_{\mathbf{w}} \mathcal{L} \odot \mathbf{w})^2$ |
| SNIP (2019) | $ \nabla_{\mathbf{w}} \mathcal{L} \odot \mathbf{w} $ |
| GraSP (2020) | $-H \nabla_{\mathbf{w}} \mathcal{L} \odot \mathbf{w}$ |
| SynFlow (2020) | $\frac{\partial \mathcal{R}}{\partial \mathbf{w}} \odot \mathbf{w}, \mathcal{R} = \mathbf{1}^\top (\Pi_{l=1}^L \mathbf{w}^{[l]}) \mathbf{1}$ |

Table 2: Summary of pruning criteria in *static-mask* sparse training methods. Above the dash line are PaT methods. \mathcal{L} denotes the loss function; H represents Hessian; $\mathbf{1}$ is the all ones vector; l denotes the l -th layer of all L layers. Skeletonization [Mozer and Smolensky, 1989]. Taylor-FO [Molchanov *et al.*, 2019].

Existing criteria are primarily made up with **two ingredients**: weight **magnitude** and/or (Hessian conditioned) **gradient**.

PaI Case

- PaI is **similar to PaT** in this regard.

Classic Topics in Pruning: #4 Pruning Schedule

How to schedule the pruning process?

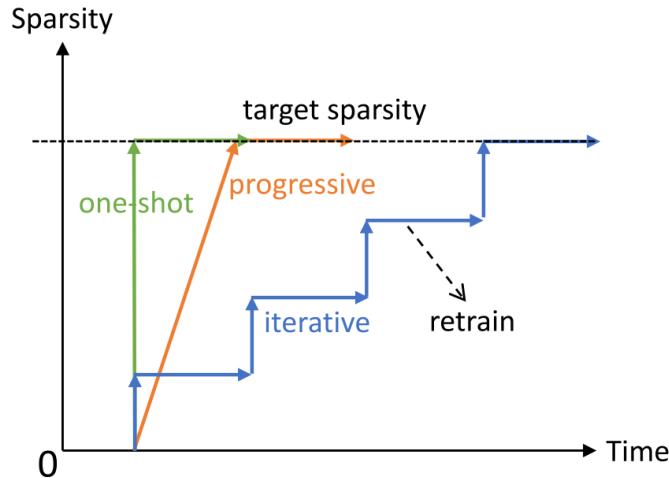


Fig. 1. Comparison among the three common pruning processes. A typical pruning process contains two steps: pruning and retraining. One-shot pruning achieves the target sparsity ratio in a single step followed by a long-time retraining; if the target sparsity ratio is divided into many sub-pruning processes, it is called iterative pruning; progressive pruning is the middle-way of one-shot and iterative pruning, where the sparsity grows gradually but it only needs one retraining, which is a better trade-off between flexibility and time cost.

- One-shot pruning
- progressive, iterative

Consensus:

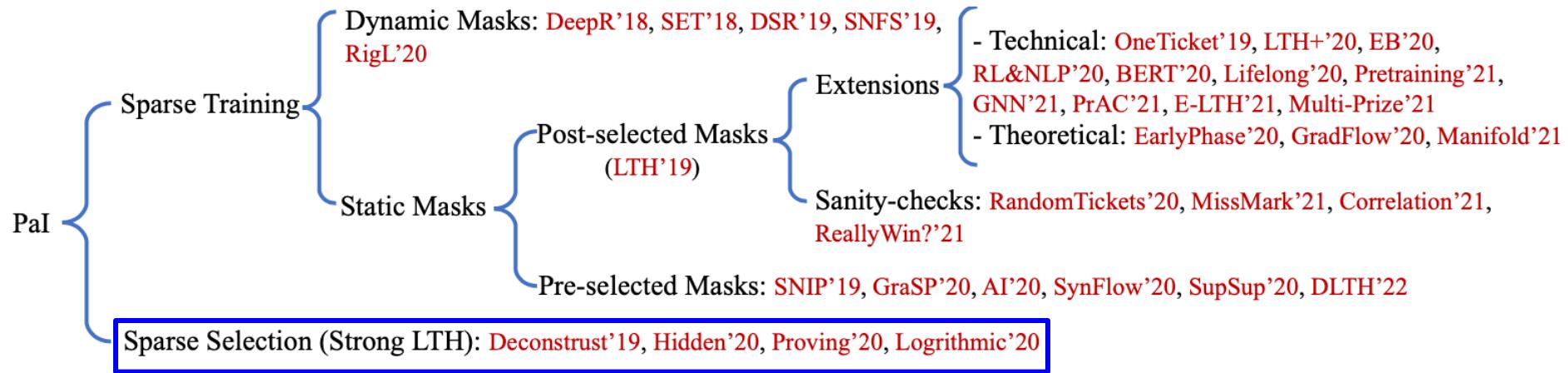
Progressive/iterative is better than one-shot (at more training cost) as it allows for the network more time to adapt over the pruning process.

PaI Case

- LTH uses iterative pruning.
- SNIP uses one-shot pruning.

Illustration of popular pruning schedules
[Wang et al., JSTSP, 2019]

How PaI is Different from PaT in the 4 Classic Pruning Topics



□ Sparse Training

| | PaT | PaI |
|--------------------|------------------------------|-------------------------|
| Sparsity structure | Mostly structured | Nearly all unstructured |
| Pruning ratio | Global & local | Mostly global |
| Pruning criterion | Magnitude+Gradient | Magnitude+Gradient |
| Pruning schedule | Mostly iterative/progressive | One-shot & iterative |

□ Sparse Selection (Brand-new🔥)

Most theoretical works are concentrated on this part.

Open Questions: #1 Under-performance

- ❑ Under-performance of PaI: PaI < PaT
 - ❑ It is very easy to find such examples in PaI papers.
 - ❑ **Why? Is it a fundamental gap or just because we do not find the right PaI method?**

Table 2: Test accuracy of pruned VGG19 and ResNet32 on CIFAR-10 and CIFAR-100 datasets. The bold number is the higher one between the accuracy of GraSP and that of SNIP.

| Dataset | CIFAR-10 | | | CIFAR-100 | | |
|--------------------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
| Pruning ratio | 90% | 95% | 98% | 90% | 95% | 98% |
| VGG19 (Baseline) | 94.23 | - | - | 74.16 | - | - |
| OBD (LeCun et al., 1990) | 93.74 | 93.58 | 93.49 | 73.83 | 71.98 | 67.79 |
| MLPrune (Zeng & Urtasun, 2019) | 93.83 | 93.69 | 93.49 | 73.79 | 73.07 | 71.69 |
| LT (original initialization) | 93.51 | 92.92 | 92.34 | 72.78 | 71.44 | 68.95 |
| LT (reset to epoch 5) | 93.82 | 93.61 | 93.09 | 74.06 | 72.87 | 70.55 |
| DSR (Mostafa & Wang, 2019) | 93.75 | 93.86 | 93.13 | 72.31 | 71.98 | 70.70 |
| SET Mocanu et al. (2018) | 92.46 | 91.73 | 89.18 | 72.36 | 69.81 | 65.94 |
| Deep-R (Bellec et al., 2018) | 90.81 | 89.59 | 86.77 | 66.83 | 63.46 | 59.58 |
| SNIP (Lee et al., 2018) | 93.63±0.06 | 93.43±0.20 | 92.05±0.28 | 72.84±0.22 | 71.83±0.23 | 58.46±1.10 |
| GraSP | 93.30±0.14 | 93.04±0.18 | 92.19±0.12 | 71.95±0.18 | 71.23±0.12 | 68.90±0.47 |
| ResNet32 (Baseline) | 94.80 | - | - | 74.64 | - | - |
| OBD (LeCun et al., 1990) | 94.17 | 93.29 | 90.31 | 71.96 | 68.73 | 60.65 |
| MLPrune (Zeng & Urtasun, 2019) | 94.21 | 93.02 | 89.65 | 72.34 | 67.58 | 59.02 |
| LT (original initialization) | 92.31 | 91.06 | 88.78 | 68.99 | 65.02 | 57.37 |
| LT (reset to epoch 5) | 93.97 | 92.46 | 89.18 | 71.43 | 67.28 | 58.95 |
| DSR (Mostafa & Wang, 2019) | 92.97 | 91.61 | 88.46 | 69.63 | 68.20 | 61.24 |
| SET Mocanu et al. (2018) | 92.30 | 90.76 | 88.29 | 69.66 | 67.41 | 62.25 |
| Deep-R (Bellec et al., 2018) | 91.62 | 89.84 | 86.45 | 66.78 | 63.90 | 58.47 |
| SNIP (Lee et al., 2018) | 92.59±0.10 | 91.01±0.21 | 87.51±0.31 | 68.89±0.45 | 65.22±0.69 | 54.81±1.43 |
| GraSP | 92.38±0.21 | 91.39±0.25 | 88.81±0.14 | 69.24±0.24 | 66.50±0.11 | 58.43±0.43 |

An example of PaI (red box) under-performs PaT (blue box)

src: [Wang et al., ICLR, 2020]

Open Questions: #1 Under-performance

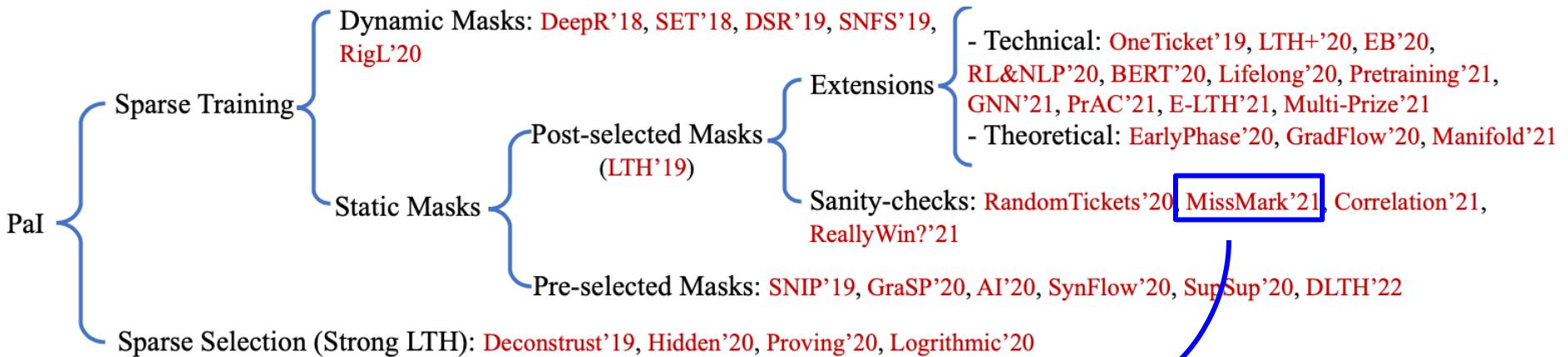


Figure 1: Overview of pruning at initialization (PaI) approaches, classified into two general groups: *sparse training* and *sparse selection*. For readability, references are omitted in this figure but the paper abbreviations (right beside the abbreviation is the year when the paper appeared). Please see Sec. 3 for detailed introductions of them. Due to limited length, this paper only outlines the primary methods, see *full* collection at <https://github.com/mingsun-tse/awesome-pruning-at-initialization>.

MissMark'21 (Pruning Neural Networks at Initialization: Why are We Missing the Mark?) presents several sanity-checking ablations. To our surprise, *none* of the popular PaI methods can pass (while PaT can pass) them. This discovery poses an acute challenge towards the principles and motivations of PaI at present.

Open Questions: #2 Not Really Faster/Saving

□ Under-development of sparse libraries

- Recall that PaI is most on **unstructured pruning**, which is hard to leverage for acceleration.
- Few works have reported *wall-time speedup* in sparse training.
- **The promised faster sparse training has not been substantiated so far.**

To sum, the major challenge facing PaI is to deliver the practical **training speedup** with **no (serious) performance compromised**, as it promises.



Thank you!

- ❑ Paper Collection: <https://github.com/mingsun-tse/Awesome-Pruning-at-Initialization>
- ❑ Code Base: <https://github.com/mingsun-tse/Smile-Pruning>

Welcome to add more PaI papers if you see appropriate!

Acknowledgments

We thank Jonathan Frankle, Alex Renda, and Michael Carbin from MIT for their very helpful suggestions to our work.

References

1. [Wen et al., NIPS, 2016] Learning Structured Sparsity in Deep Neural Networks
2. [Mao et al., CVPRw, 2017] Exploring the granularity of sparsity in convolutional neural networks
3. [Ding et al., ICML, 2019] Approximated Oracle Filter Pruning for Destructive CNN Width Optimization
4. [Lee et al., ICLR, 2019] Snip: Single-shot network pruning based on connection sensitivity
5. [Wang et al., JSTSP, 2019] Structured pruning for efficient convolutional neural networks via incremental regularization
6. [Gale et al., Arxiv, 2019] The State of Sparsity in Deep Neural Networks
7. [Frankle and Carbin., ICLR, 2019] The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks
8. [Wang et al., ICLR, 2020] Picking Winning Tickets Before Training By Preserving Gradient Flow
9. [Wang et al., ICLR, 2021] Neural Pruning via Growing Regularization