The Lottery Tickets Hypothesis for Supervised and Self-supervised Pre-training in Computer Vision Models

Tianlong Chen¹, Jonathan Frankle², Shiyu Chang³, Sijia Liu³, Yang Zhang³, Michael Carbin², Zhangyang Wang¹

¹University of Texas at Austin, ²MIT CSAIL, ³MIT-IBM Watson AI Lab, IBM Research

{tianlong.chen,atlaswang}@utexas.edu, {jfrankle,mcarbin}@csail.mit.edu, {shiyu.chang,sijia.liu,yang.zhang2}@ibm.com

Abstract

The computer vision world has been re-gaining enthusiasm in various pre-trained models, including both classical ImageNet supervised pre-training and recently emerged self-supervised pre-training such as simCLR [8] and MoCo [34]. Pre-trained weights often boost a wide range of downstream tasks including classification, detection, and segmentation. Latest studies suggest that the pre-training benefits from gigantic model capacity [9]. We are hereby curious and ask: after pre-training, does a pre-trained model indeed have to stay large for its universal downstream transferability?

In this paper, we examine the supervised and self-supervised pre-trained models through the lens of lottery ticket hypothesis (LTH) [26]. LTH identifies highly sparse matching subnetworks that can be trained in isolation from (nearly) scratch, to reach the full models’ performance. We extend the scope of LTH to questioning whether matching subnetworks still exist in the pre-training models, that enjoy the same downstream transfer performance. Our extensive experiments convey an overall positive message: from all pre-trained weights obtained by ImageNet classification, simCLR and MoCo, we are consistently able to locate such matching subnetworks at 59.04% to 96.48% sparsity that transfer universally to multiple downstream tasks, whose performance see no degradation compared to using full pre-trained weights. We find task-agnostic, universally transferable subnetworks at pre-trained initialization, for classification and segmentation tasks; while the detection task (in particular, two-stage object detector) seems to require task-specific search of matching subnetworks.

Figure 1. Overview of our work paradigm: from pre-trained CV models (both supervised and self-supervised), we study the existence of matching subnetworks that are transferable to many downstream tasks, with little performance degradation compared to using full pre-trained weights. We find task-agnostic, universally transferable subnetworks at pre-trained initialization, for classification and segmentation tasks; while the detection task (in particular, two-stage object detector) seems to require task-specific search of matching subnetworks.

1. Introduction

Deep neural networks pre-trained on large-scale datasets prevail as general-purpose feature extractors [18]. Moving beyond the most traditional greedy unsupervised pre-training [2], the most popular pre-training in computer vision (CV) nowadays is arguably to train the model for supervised classification on ImageNet [13]. Such supervised pre-training forces the network to learn a hierarchy of generalizable features [40]; it is widely acknowledged to not only benefit the subsequent fine-tuning on other visual classification datasets (especially in small datasets and few-shot learning [64, 67]), but also accelerate/improve the training for different, more complicated types of downstream vision tasks, such as object detection and semantic segmentation [55, 35].

Lately, various self-supervised pre-training approaches rise to popularity. Several state-of-the-art algorithms, such as simCLR [8, 9] and MoCo [34, 12], have demonstrated the success by making best use of massive unlabeled data in pretraining. Their methods refer to no actual labels in pre-training, but instead leverage self-generated pseudo la-
The paper carries out the first comprehensive experimental study, to seek these desired universal matching subnetworks, from pre-trained CV models in both supervised and self-supervised ways. Our principled methodology bridges pre-training and LTH from two perspectives: i) Initialization via pre-training. In the previous larger-scale settings of LTH for CV \cite{26, 62}, the matching subnetworks are found at an early point in training rather than at random initialization. Instead, we aim to identify these matching subnetworks from dense pre-trained models (self-supervised or supervised), which creates an initialization directly amenable to sparsification. ii) Transfer learning. Finding the matching subnetwork is an expensive investment, usually costing multiple rounds of pruning and re-training. To justify this extra investment, the found subnetwork must be able to be reused by various downstream tasks, as illustrated in Figure 1.

The course of this study presents the following findings:

- Using unstructured magnitude pruning as in \cite{26}, we identify matching sub-networks up to 67.23\%, 59.04\%, 95.60\% sparsity, at pre-trained weights from ImageNet-equipped supervised pre-training, simCLR and MoCo, respectively. We also find matching subnetworks at pre-trained initialization with sparsity from 73.79\% to 98.20\% in a variety of classification, detection and segmentation downstream tasks.

- Subnetworks at 67.23\%, 59.04\% and 59.04\% sparsity, found respectively using supervised ImageNet, simCLR and MoCo pre-training, are universally transferable to diverse downstream classification tasks with nearly same accuracies.

- Subnetworks at 48.80\%, 36.00\% and 83.22\% sparsity, found respectively by supervised ImageNet, simCLR and MoCo, can transfer to downstream segmentation tasks without sacrificing performance. Unfortunately, transferring these subnetworks to downstream detection tasks will incur performance degradation.

- Unlike previous matching subnetworks found at random initialization or early rewinding, we show that those identified at pre-trained initialization are more sensitive to structure perturbations. Also, different pre-training ways tend to yield diverse mask structures and perturbation sensitivities.

- Lastly, pruning from larger pre-trained models can also produce better transferable matching subnetworks.

Practically speaking, this work sets the first step toward replacing huge pre-trained models with smaller subnetworks, enabling much more efficient downstream tuning without inhibiting transfer performance. However, the problem is far from being fully resolved, and our comparisons on classification/segmentation versus detection tasks reveal more...
subtlety that invites follow-up studies. As pre-training becomes increasingly central in the CV field, our results shed light on the relevance of LTH in this new paradigm.

2. Related Works

Pruning and Lottery Tickets Hypothesis. A trained deep network could be pruned of excess capacity [43]. Pruning algorithms can be grouped into unstructured [33, 43, 32] and structured [47, 37, 79]: the former sparsifies based on weight magnitudes; while the latter considers hardware-friendliness by removing channels and so on.

The discovery of LTH [26] deviates from the convention of after-training pruning, and points to the existence of independently trainable sparse subnetworks from scratch that can match the performance of dense networks. Follow-up investigations [48, 30] scale up LTH by rewinding approaches [28, 62], that re-initializes the subnetwork from the early training stage checkpoint rather than from scratch. LTH has been widely explored in image classification [26, 48, 69, 23, 29, 65, 73, 74], natural language processing [30, 76, 62, 59, 7], and reinforcement learning [76]. Most of them adopt (iterative) unstructured weight magnitude pruning [32, 26], which is also followed in this work. [50, 51, 14] pioneer to study the transferability of the subnetworks identified on one image classification task to another. However, studying the universal transferability of LTH at pre-trained initializations among diverse CV tasks remains untouched.

One most relevant work [7] to ours is from the NLP field: the authors found universally transferable sparse matching subnetworks (at 40% to 90% sparsity), from the pre-trained initialization of BERT models. Finding their work inspiring, we stress that transplanting their NLP findings to our CV task of the same simplicity, e.g. “ImageNet” represents the supervised pre-training classification task on ImageNet.

| Settings | Pre-training | Downstream Classification | Downstream Detection | Downstream Segmentation |
|----------|--------------|---------------------------|----------------------|-------------------------|
|          | ImageNet     | CIFAR-10 | CIFAR-100 | SVHN | Fashion-MNIST | VisDA2017 | Pascal VOC2012/2007 | Pascal VOC 2012 |
| # Epochs/Iters | 10 | 10 | 10 | 182 | 182 | 182 | 182 | 20 | 24K Iters |
| Batch Size | 256 | 256 | 256 | 256 | 256 | 256 | 256 | 128 | 30K Iters |
| Learning Rate | 0.0001 | 0.0001 | 0.0003 | 0.1 | 0.1 | 0.1 | 0.1 | 0.0001 | 0.0005 |
| Optimizer | Fixed schedule | 0.1 at 91,136 epcoh | 0.1 at 10 epcoh | Polynomial decay with power 0.9 | Linear warmup 100 Iters |
| Weight Decay | 1 × 10⁻⁴ | 1 × 10⁻⁴ | 1 × 10⁻⁴ | 2 × 10⁻⁴ | 2 × 10⁻⁴ | 2 × 10⁻⁴ | 2 × 10⁻⁴ | 1 × 10⁻⁴ |
| Eval. Metric | Accuracy | Retrieval Accuracy | Accuracy | Accuracy | Accuracy | Accuracy | Accuracy | AP, AP₉₀, AP₇₅ |

3. Preliminaries and Setups

In this section, we provide the detailed experimental settings and our approaches to find matching subnetworks.

Network. We use the official ResNet-50 [36] network architecture as our default backbone, while we will later compare on ResNet-152 in Section 5.2. For a particular classification downstream task, a task-specific final linear layer is added following [8]. Faster RCNN [61] and DeepLabV3+ [6] are adopted for the detection and segmentation downstream tasks respectively, which also take ResNet-50 as the backbone. Due to the various input and output scales, the first convolution layer in ResNet-50 is shared among the classification, detection, segmentation heads are never pruned. Specifically, we let $f(x; \theta, \gamma)$ be the output of a ResNet-50 model with parameters $\theta \in \mathbb{R}^{d_1}$ (excluding the first convolution layer) and task-specific parameters $\gamma \in \mathbb{R}^{d_2}$ on an input image $x$.

Note that, for complicated computer vision tasks such as object detection, the large variety of model design options may likely have impact on our observation. For example, object detection models are well-known to fall under two-stage and one-stage categories, the former often achieving higher accuracy while the latter typically being much faster. Our current study prioritizes the two-stage object detectors,
using Faster RCNN as a subject example. However, we plan to expand our experiments to one-stage detectors soon, and plan to report new results in the short foreseeable future.

**Pre-training.** For the supervised pre-training, we use the official pre-trained ResNet-50\(^1\) on the ImageNet dataset [13]. For the self-supervised, we adopt the pre-trained ResNet-50 models with simCLR\(^2\) [8] and MoCov2\(^3\) [12] on ImageNet.

**Datasets, Training and Evaluation.** All pre-training experiments are conducted on ImageNet. For downstream tasks, we consider classification, object detection and semantic segmentation on multiple datasets. We use four natural image and one synthetic datasets to verify the transferability on classification: Fashion-MNIST [72], SVHN [52], CIFAR-10 [42], CIFAR-100 [42], and VisDA2017 [58]. These datasets vary remarkably in terms of sample size, color space, resolution, image source, and classes. Following [34, 12], we train object detection models on the combined training and validation set of Pascal VOC 2012 [24] and Pascal VOC 2007 [25], then evaluate them on the Pascal VOC 2007 test set. We train and evaluate semantic segmentation models on Pascal VOC 2012 training and validation sets. We follow the standard hyperparameters and evaluation metrics\(^4\) for all pre-training and downstream tasks, as in Table 1.

**Subnetworks.** For a network \(f(x; \theta, \gamma)\) with task-specific modules \(\gamma\), its subnetworks can be depicted as \(f(x; m \odot \theta, \gamma)\) with a pruning binary mask \(m \in \{0, 1\}^d\), where \(\odot\) is the element-wise product. Let \(A_i^T(f(x; \theta, \gamma))\) be a training algorithm (e.g., SGD with certain hyperparameters) that trains a network \(f(x; \theta, \gamma)\) on a task \(T\) (e.g., CIFAR-10) for \(t\) iterations. Let \(\theta_p \in \{\theta_{\text{img}}, \theta_{\text{sim}}, \theta_{\text{MoCo}}\}\) be the pre-trained weights on ImageNet, where \(\theta_{\text{img}}\) is the supervised pre-trained weight, \(\theta_{\text{sim}}\) and \(\theta_{\text{MoCo}}\) are from the self-supervised pre-training by simCLR [8] and MoCov2 [12]. Let \(\theta_0\) be the random initialization, and \(\theta_t\) be the network weights at the \(t\)th epoch which is trained from \(\theta_0\). Let \(E^T(f(x; \theta, \gamma))\) be the evaluation function of model \(f\) on the corresponding task \(T\). Below we define:

1. **Matching subnetworks.** Following the definition in [27, 7], a subnetwork \(f(x; m \odot \theta, \gamma)\) is **matching** if it satisfies the following condition:

\[
E^T(A_i^T(f(x; m \odot \theta, \gamma))) \geq E^T(A_i^T(f(x; \theta_p, \gamma)))
\]

That is, matching subnetworks perform no worse than the full dense models under the same training algorithm \(A_i^T\) and evaluation metric \(E^T\).

2. **Winning ticket.** If \((x; m \odot \theta, \gamma)\) is a matching subnetwork with \(\theta = \theta_p\) for \(A_i^T\), it is a **winning ticket** for \(A_i^T\).

3. **Universal subnetwork.** A subnetwork \((x; m \odot \theta, \gamma_T)\) with task-specific configurations of \(\gamma_T\), is **universal** for tasks \(\{T_i\}_{i=1}^N\) if and only if it is matching for each \(A_i^T\). The task set \(\{T_i\}_{i=1}^N\) could be a group of (diverse) downstream tasks, such as classification, detection and segmentation.

**Pruining Methods.** To find the subnetworks \(f(x; m \odot \theta, \gamma)\), we adopt the classical iterative magnitude pruning (IMP) approach that is commonly used by the LTH literature [26, 27, 7]. We prune the network by first training the unpruned dense network to completion on a task \(T\) (i.e., applying \(A_i^T\) ) and then removing a portion of weights with the globally smallest magnitudes [32, 62]. As revealed by previous works, in order to identify the most competitive matching subnetworks, the process needs to be iteratively repeated for several rounds. Algorithm 1 outlines the full IMP procedure in the supplement.

Although beyond the current scope, our future work plans to examine the practical speedup results on a hardware platform for our training and/or inference phases. For example, in the range of 70%-90% unstructured sparsity, XNNPACK [21] has already shown significant speedups over dense baselines on smartphone processors. Integrating structured pruning will be another future direction of our interest [74].

### 4. Transfer of Pre-training Winning Tickets

In this section, we first show that there exist winning tickets using the pre-trained initialization on both self-supervised and supervised pre-training tasks. As shown in Figure 2, We find winning tickets with 67.23\%, 59.04\% and 95.60\% spar-
Figure 3. Performance of IMP subnetworks with a range of sparsity from 0.00% to 98.20% (i.e., remaining weight from 100% to 1.80%) on downstream classification tasks, including CIFAR-10, CIFAR-100, SVHN and Fashion-MNIST. \((m_{\text{Img}}, \theta_{\text{Img}}), (m_{\text{sim}}, \theta_{\text{sim}}), (m_{\text{MoCo}}, \theta_{\text{MoCo}})\) denote transfer performance of subnetworks found at pre-training tasks. Subnetworks \((m_{T_i}, \theta_p), T_i \in \{\text{CIFAR-10, CIFAR-100, SVHN, Fashion-MNIST}\}\) and \(\theta_p \in \{\theta_{\text{Img}}, \theta_{\text{sim}}, \theta_{\text{MoCo}}\}\) are identified on the downstream task \(T_i\) with pre-trained weights \(\theta_p\). Curves with errors (shadow regions) are the average across three independent runs, with the standard deviations: same hereinafter.

Figure 4. Performance of IMP subnetworks with a range of sparsity from 0.00% to 98.20% on the synthetic dataset, VisDA2017.

sity for supervised ImageNet, self-supervised simCLR and MoCo pre-training tasks.

Then, we investigate to what extent IMP subnetworks found for pre-training tasks can (universally) transfer to different downstream tasks. We ask the following questions:

**Q1:** Are winning tickets \(f(x; m_P \odot \theta_p, \cdot)\), found on the pre-training task \(P\), also winning tickets for other downstream tasks \(T\)?

**Q2:** Are there common patterns in the transferability of winning tickets from different pre-trainings (e.g., supervised versus self-supervised)?

**Q3:** Can the transferred subnetworks \(f(x; m_P \odot \theta_p, \cdot)\) outperform the subnetworks \(f(x; m_T \odot \theta_{T_i}, \cdot) (\theta_{T_i} \in \{\theta_0, \theta_{5\%}, \theta_p\})\), found on a specific task \(T_i\)?

### 4.1. Transfer to Classification Tasks

As shown in Figures 3 and 4, evaluated subnetworks are divided into three groups, according to sources of \((m, \theta)\): i) transferred subnetworks with \((m_P, \theta_p), P \in \{\text{Img, sim, MoCo}\}\) and \(\theta_p \in \{\theta_{\text{Img}}, \theta_{\text{sim}}, \theta_{\text{MoCo}}\}\); ii) subnetworks found on a specific downstream tasks with pre-trained weights \((m_T, \theta_p), T \in \{\text{CIFAR-10, CIFAR-100, SVHN, Fashion-MNIST, VisDA2017}\}\); iii) subnetworks consist of \((m_{T_i}, \theta_{T_i}), \theta_{T_i} \in \{\theta_0, \theta_{5\%}\}\), identified with the original random initialization \(\theta_0\) or early rewinding weights \(\theta_{5\%}\). Early weight rewinding [62, 27] improves the quality of found matching subnetworks. As indicated by [27], the best rewinding points usually lie in the first 1% to 5% training epochs. We take 5% for default comparison.
A1: Subnetworks with \((m_p, \theta_p)\) universally transfer to diverse downstream classification tasks. As shown in Figure 3 and Figure 4, compared with unpruned dense models, subnetworks found on pre-training tasks \(f(x; m_{\text{img}} \odot \theta_{\text{img}}, \cdot)\), \(f(x; m_{\text{sim}} \odot \theta_{\text{sim}}, \cdot)\), \(f(x; m_{\text{MoCo}} \odot \theta_{\text{MoCo}}, \cdot)\) transfer without sacrificing performance\(^6\) by sparsity (91.41%, 91.41%, 91.41%) to CIFAR-10, (86.58%, 86.58%, 89.26%) to CIFAR-100, (91.41%, 96.48%, 93.13%) to SVHN, (89.26%, 89.26%, 91.41%) to Fashion-MNIST, and (67.23%, 59.04%, 59.04%) to VisDA2017. Therefore, we observe that subnetworks produced by supervised ImageNet, self-supervised simCLR and MoCo pre-training tasks, universally transfer to four downstream natural image datasets with sparsity (86.58%, 86.58%, 89.26%), respectively. However, it requires larger network capacity, i.e., (67.23%, 59.04%, 59.04%), to transfer to the synthetic VisDA2017 dataset without loss of performance.

A2: Winning tickets from different pre-training ways, have diverse behaviors, that are also affected by the downstream task properties. On natural image datasets, subnetworks found with self-supervised pre-training (i.e., simCLR and MoCo) outperform subnetworks found with supervised ImageNet pre-training at the extreme sparsity level (e.g., more than 93.13%). Specifically, \(f(x; m_{\text{sim}} \odot \theta_{\text{sim}}, \cdot)\) consistently achieves superior generalization across four downstream datasets. \(f(x; m_{\text{MoCo}} \odot \theta_{\text{MoCo}}, \cdot)\) performs worse than \(f(x; m_{\text{img}} \odot \theta_{\text{img}}, \cdot)\) at the low and middle level sparsity of subnetworks. However, the conclusions are almost flipped when transferring \(f(x; m_{\text{pr}} \odot \theta_{\text{pr}}, \cdot)\) to the synthetic VisDA2017 dataset. Subnetworks \(f(x; m_{\text{img}} \odot \theta_{\text{img}}, \cdot)\) surpass others with a large performance margin, at the sparsity from 0.00% to 89.26%. For the extreme sparsity, MoCo pre-training task generates a better transferable subnetworks. These observations suggest that supervised ImageNet pre-training allows subnetworks to transfer to the downstream datasets even with domain gaps to the pre-training datasets (e.g., from natural to synthetic images); self-supervised pre-trainings (e.g., simCLR and MoCo) produce more transferable subnetworks especially at the extreme sparsity, when natural image datasets are at downstream.

A3: Transferred subnetworks \(f(x; m_{\text{pr}} \odot \theta_{\text{pr}}, \cdot)\) perform the best until extreme sparsity. Subnetworks \(f(x; m_{\text{T}} \odot \theta_{\text{T}}, \cdot)\), found on a specific downstream task with pre-trained weights, can be considered as “performance unbound” for all our IMP subnetworks. \(f(x; m_{\text{T}} \odot \theta_{\text{T}}, \cdot)\) is identified as matching subnetworks with the sparsity (98.20%, 91.41%, 73.79%) for CIFAR-10, (91.41%, 91.41%, 20.00%) for CIFAR-100, (91.41%, 95.60%, 91.41%) for SVHN, (89.26%, 96.48%, 73.79%) for Fashion-MNIST, and (73.79%, 59.04%, 67.23%) for VisDA2007.

For universal transferable subnetworks, we observe: i) \(f(x; m_{\text{img}} \odot \theta_{\text{img}}, \cdot)\) and \(f(x; m_{\text{sim}} \odot \theta_{\text{sim}}, \cdot)\) match the corresponding \(f(x; m_{\text{T}} \odot \theta_{\text{T}}, \cdot)\) with at most 59.04% sparsity; ii) On the natural image datasets, \(f(x; m_{\text{MoCo}} \odot \theta_{\text{MoCo}}, \cdot)\) steadily outperform \(f(x; m_{\text{T}} \odot \theta_{\text{T}}, \cdot)\) by a clear margin across all sparsity levels, especially for CIFAR-100; On the synthetic dataset, it fails to match under an excessive sparsity (i.e., > 83.22%). Note that subnetworks with \(\theta_0\) and \(\theta_{90}\) are inferior on all downstream tasks, compared to subnetworks with pre-trained initialization \(\theta_p\).

4.2. Transfer to Detection and Segmentation

Training detection and segmentation models commonly starts from pre-trained initializations [55, 35, 12]. We compare the transferred subnetworks with \((m_p, \theta_p)\) versus the downstream task subnetworks with \((m_t, \theta_p)\), as shown in Figure 5. Observations are organized as follows:

A1: Subnetworks \(f(x; m_{\text{pr}} \odot \theta_{\text{pr}}, \cdot)\) transfer to the segmentation task successfully, but NOT so on the detection task. The winning tickets \(f(x; m_{\text{pr}} \odot \theta_{\text{pr}}, \cdot)\) found on the pre-training tasks are no longer matching subnetworks on the detection task, which incurs performance degradation compared to unpruned dense models \(f(x; \theta_{\text{pr}}, \cdot)\). Fortunately, we still manage to find transferable winning tickets on the segmentation task with the sparsity (48.80%, 36.00%, 83.22%) for supervised ImageNet pre-training, self-supervised simCLR and MoCo pre-training tasks respectively.

A2: Unlike classification, winning ticket from diverse pre-training tasks behave similarly on downstream detection and segmentation tasks. In Figure 5, we observe the evident ranking of achieved transfer performance across all sparsity levels: \(\mathcal{E}^T(f(x; m_{\text{MoCo}} \odot \theta_{\text{MoCo}}, \cdot)) > \mathcal{E}^T(f(x; m_{\text{img}} \odot \theta_{\text{img}}, \cdot)) > \mathcal{E}^T(f(x; m_{\text{sim}} \odot \theta_{\text{sim}}, \cdot)), \quad T \in \{\text{detection, segmentation}\}\). It suggests that MoCo pre-trained weights are most favorable for transferring to detection and segmentation tasks [12].

A3: Subnetworks \(f(x; m_{\text{T}} \odot \theta_{\text{T}}, \cdot)\) surpass subnetworks \(f(x; m_{\text{pr}} \odot \theta_{\text{pr}}, \cdot)\) by a non-negligible margin. As shown in Figure 5, with the assistance from the pre-trained initialization \((\theta_{\text{img}}, \theta_{\text{sim}}, \theta_{\text{MoCo}})\), we find winning tickets with the sparsity at level (95.60%, 93.13%, 97.75%) and (73.79%, 67.23%, 86.58%) for detection and segmentation respectively. These identified winning tickets consistently outperform transferred subnetwork with \((m_p, \theta_p)\).
Figure 5. Performance of IMP subnetworks with a range of sparsity from 0.00% to 98.20% on the downstream detection and segmentation tasks. Subnetworks \((m_{\text{VOC2007}}, \theta_p)\) and \((m_{\text{VOC2012}}, \theta_p)\), \(\theta_p \in \{\theta_{\text{Img}}, \theta_{\text{sim}}, \theta_{\text{MoCo}}\}\) are identified on the downstream detection and segmentation tasks with pre-trained weights \(\theta_p\), respectively.

Figure 6. Top: The relative mask similarity between subnetworks which identified on supervised ImageNet, simCLR and MoCo pre-training tasks. Bottom: The number of completely pruned (zero) kernels in subnetworks found on different pre-training tasks.

Figure 7. Kernel-wise heatmap visualizations of subnetworks with 79.03% sparsity found on supervised ImageNet, simCLR and MoCo pre-training tasks. From left to right, we visualization all kernels of subnetworks from the input to the output layers. The bright dots \(\ast\) represent the completely pruned (zero) kernels and the dark dots \(\bullet\) the kernels having at least one unpruned weight.

\(i, j \in \{\text{Img}, \text{sim}, \text{MoCo}\}\). We find that subnetworks for pre-training tasks are remarkably heterogeneous: they share less than 6.55% locations in common after five-round IMP; the more sparsified, the larger differences.

We also calculate the number of completely pruned (zero) kernels of subnetworks in Figure 6, which roughly reveals the weight clustering status in the sparse models. We observe that the remaining weights of subnetworks identified on the MoCo pre-training task are more clustered (i.e., more zero kernels) than the ones from ImageNet and simCLR, until reaching an extreme sparsity like 95.60%.

Specifically, we provide kernel-wise heatmap visualizations of subnetworks with 79.03% sparsity in Figure 7. We find that the completely pruned (zero) kernels are mainly clustered in the early layers of subnetworks, and appear rarely in the later layers. Among three kinds of subnetworks, the one from MoCo has the most dispersed distribution of completely pruned kernels. In general, more structured sparse subnetworks (i.e., more all-zero kernels) may have a stronger potential for hardware speedup [21].

5.2. Pre-training versus Random Initialization

A signature of our setting is to treat pre-trained weights as the initialization, in contrast to most LTH works starting from random initialization [26, 27]. These two configurations produce matching subnetworks with diverse behaviors, including generalization performance and the structure sensitivity of obtained masks. We performs IMP on CIFAR-100 with the original random initialization \(\theta_0\), early rewinding weights \(\theta_{5\%}\), and the pre-trained weights \(\theta_{\text{Img}}\) respectively, and then generates subnetworks consisting of \((m_{\text{CIFAR-100}}, \theta)\), \(\theta \in \{\theta_0, \theta_{5\%}, \theta_{\text{Img}}\}\). As for comparison baselines, we consider there mask variants, the complementary masks \(m^{\text{CIFAR-100}}\), randomly pruned masks \(m_r\), and the perturbed masks \(m_{\text{CIFAR-100}} + \Delta m_{10\%}\) as in Figure 8. Several observations can be draw as follows:

- Starting from \(\theta_0\) or \(\theta_{5\%}\), identified subnetworks are resilient to structure perturbations. In other words, there only exist marginal performance differences across subnetworks with masks \(m^{\text{CIFAR-100}}, m^{\text{CIFAR-100}} + m_r\), and \(m^{\text{CIFAR-100}} + \Delta m_{10\%}\). However, the found subnetworks with the pre-trained initialization behave in sharp contrast, that all complementary masks, random pruned masks and perturbed masks substantially de-
Subnetworks with Early Weight Rewinding
Subnetworks with Pre-trained Initialization

Figure 8. Performance comparison across subnetworks found on CIFAR-100 with the original random initialization \( \theta_0 \), early rewinding weight \( \theta_{\text{rew}} \), and the pre-trained weights \( \theta_{\text{pre}} \). \( m_{\text{CIFAR-100}} \) = masks found by IMP; \( m_{\text{CIFAR-100}} \) = the complementary masks of \( m_{\text{CIFAR-100}} \), where \( m \cap m^c = m \cup m^c = 1 \in \mathbb{R}^d \); \( m_r \) = random pruned mask; \( \Delta m_{10\%} \) = mask perturbations by randomly flipping 10% “1” and 10% “0” in the mask \( m \in \{0, 1\}^d \) to its opposite value. Curves are the average across three independent runs.

graded the performance w.r.t. the IMP masks. A possible explanation is that the pre-trained initialization are already highly structured, and perturbations can destroy the intrinsic structure. As evidenced by the right subfigure of Figure 8, subnetworks with \( (m_{\text{CIFAR-100}}, \theta_{\text{rew}}) \) are no better than subnetwork with \( (m_{\text{CIFAR-100}}, \theta_0) \).

- Comparing the randomly pruned subnetworks in Figure 8, we observe that pre-trained initialization consistently benefits the accuracy until subnetworks reaching some high sparsity (e.g., 67.23%). After that, the performance of random pruned subnetworks is no longer affected by different initializations.

5.3. More Ablation Studies for Pre-training

Larger Pre-training Model? [46] reveals that heavily compressed, large transformer models achieve higher performance than lightly compressed, small transformer models in natural language processing. We re-confirm this claim for self-supervised simCLR pre-training, in terms of the transferability\(^7\) of found matching subnetworks.

In Figure 9, with the same number of remaining weights, subnetworks pruned from simCLR\(^8\) pre-trained ResNet-152, achieve consistently superior accuracy on the downstream CIFAR-100 task than the ones from simCLR pre-trained ResNet-50 (around one-third size of ResNet-152). At least for simCLR, pruning from larger pre-trained models produces better transferable matching subnetworks.

Our observation is also aligned with the advocates of [9], to first pretrain a big model and then compress it. The key difference is that, [9] uses standard model compression (knowledge distillation) after downstream fine-tuning is done; in contrast, our results can be seen as a possible second pre-training stage: after the initial pre-training (and before any fine-tuning), performing IMP to find equally-capable matching subnetwork with far fewer parameters.

Temperature Hyperparameter. The temperature scaling hyperparameter is known to play a significant role in the quality of the simCLR pre-training [8, 9, 10]. It motivates us to investigate the impact of the temperature scaling factor on the transferability of pre-training winning tickets found in Section 4. Without loss of the generality, we consider the training and pruning process of subnetworks with the sparsity from 67.23% to 73.79%. Specifically, we start from training subnetworks at the sparsity level 67.23% for 10 epochs, on the simCLR task with different temperature scaling factors. Then, they are pruned to the level of 73.79% sparsity by IMP. In the end, subnetworks are fine-tuned and evaluated on the downstream CIFAR-100 task. Results in Table 2 show that found subnetworks have close transfer performance if the temperature scaling factor lies in a moderate range (i.e., [0.1, 0.5]), and the performance will degrade at extreme temperatures (e.g., 20.0).

| Temperature | 0.1 | 0.2 | 0.5 | 1.0 | 2.0 | 10.0 | 20.0 |
|-------------|-----|-----|-----|-----|-----|------|------|
| Accuracy (%)| 81.81 | 81.91 | **82.22** | 81.24 | 80.76 | 81.46 | 80.18 |

\(^7\)In the supplement, we also report the pre-training task performance of subnetworks generated from small- and large-scale pre-trained simCLR.

\(^8\)For a fair comparison, here we adopt the simCLRv2\([9]\) pre-trained ResNet-152 and ResNet-50 models, since only simCLRv2 released the official pre-trained ResNet-152 model.

Figure 9. Transfer performance on CIFAR-100 over the number of remaining weights. Subnetworks are found on the simCLR pre-training task with pre-trained ResNet-50 and ResNet-152 weights.
6. Conclusion

We study the lottery ticket hypothesis in the context of CV pre-training, via both supervised (e.g., ImageNet classification) and self-supervised (e.g., simCLR and MoCo) ways. Despite the complicity of our goal, by performing IMP from the pre-trained initializations, we are consistently able to find matching subnetworks at non-trivial sparsity levels, that can be independently trained to full model performance, on both pre-training and downstream tasks. We also present a detailed discussion of cross-task universal transferability. Our future work plans to extend our experiments and observations, to both more model types (e.g., one-stage object detector), and more computer vision tasks (e.g., 3D vision).

Acknowledgments

We would like to express our deepest gratitude to the MIT-IBM Watson AI Lab, in particular John Cohn for generously providing us with the computing resources necessary to conduct this research. Wang’s work is in part supported by the NSF Energy, Power, Control, and Networks (EPCN) program (Award number: 1934755), and by an IBM faculty research award.

References

[1] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In Advances in Neural Information Processing Systems, pages 15535–15545, 2019. 3
[2] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. Advances in neural information processing systems, 19:153–160, 2006. 1
[3] Han Cai, Chuang Gan, Ligeng Zhu, and Song Han. Tiny transfer learning: Towards memory-efficient on-device learning. arXiv preprint arXiv:2007.11622, 2020. 3
[4] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In Proceedings of the European Conference on Computer Vision (ECCV), pages 132–149, 2018. 3
[5] Mathilde Caron, Piotr Bojanowski, Julien Mairal, and Armand Joulin. Unsupervised pre-training of image features on non-curated data. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), pages 2959–2968, 2019. 3
[6] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European conference on computer vision (ECCV), pages 801–818, 2018. 3
[7] Tianlong Chen, Jonathan Frankle, Shiyu Chang, Sijia Liu, Yang Zhang, Zhangyang Wang, and Michael Carbin. The lottery ticket hypothesis for pre-trained bert networks. arXiv preprint arXiv:2007.12223, 2020. 3, 4, 12
[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. arXiv preprint arXiv:2002.05709, 2020. 1, 2, 3, 4, 8, 12
[9] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. arXiv preprint arXiv:2006.10029, 2020. 1, 2, 3, 8
[10] Ting Chen and Lala Li. Intriguing properties of contrastive losses, 2020. 8
[11] Tianlong Chen, Sijia Liu, Shiyu Chang, Yu Cheng, Lisa Amini, and Zhangyang Wang. Adversarial robustness: From self-supervised pre-training to fine-tuning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 699–708, 2020. 3
[12] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297, 2020. 1, 3, 4, 6
[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pages 248–255. Ieee, 2009. 1, 4
[14] Shrey Desai, Hongyuan Zhan, and Ahmed Aly. Evaluating lottery tickets under distributional shifts. arXiv preprint arXiv:1910.12708, 2019. 3
[15] Tim Dettmers and Luke Zettlemoyer. Sparse networks from scratch: Faster training without losing performance. arXiv preprint arXiv:1907.04840, 2019. 2
[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018. 2, 3
[17] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In Proceedings of the IEEE international conference on computer vision, pages 1422–1430, 2015. 2, 3
[18] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In International conference on machine learning, pages 647–655, 2014. 1
[19] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929, 2020. 2
[20] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. In Advances in neural information processing systems, pages 766–774, 2014. 2, 3
[21] Erich Elsen, Marat Dukhan, Trevor Gale, and Karen Simonyan. Fast sparse convnets. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 14629–14638, 2020. 4, 7
[22] Utku Evci, Trevor Gale, Jacob Menick, Pablo Samuel Castro, and Erich Elsen. Rigging the lottery: Making all tickets winners. In International Conference on Machine Learning, 2020. 2
[52] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. 4
[53] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In European Conference on Computer Vision, pages 69–84. Springer, 2016. 3
[54] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748, 2018. 3
[55] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1717–1724, 2014. 1, 6
[56] Deepak Pathak, Ross Girshick, Piotr Dollár, Trevor Darrell, and Bharath Hariharan. Learning features by watching objects move. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2701–2710, 2017. 3
[57] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2536–2544, 2016. 3
[58] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. arXiv preprint arXiv:1710.06924, 2017. 4
[59] Sai Prasanna, Anna Rogers, and Anna Rumschisky. When bert plays the lottery, all tickets are winning. arXiv preprint arXiv:2005.00561, 2020. 3
[60] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019. 2
[61] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, pages 91–99, 2015. 3
[62] Alex Renda, Jonathan Frankle, and Michael Carbin. Comparing reweighting and fine-tuning in neural network pruning. In International Conference on Learning Representations, 2020. 2, 3, 4, 5
[63] Sebastian Ruder. An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747, 2016. 3
[64] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. International journal of computer vision, 115(3):211–252, 2015. 1
[65] Pedro Savares, Hugo Silva, and Michael Maire. Winning the lottery with continuous sparsification, 2020. 3
[66] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. arXiv preprint arXiv:1906.05849, 2019. 3
[67] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? arXiv preprint arXiv:2003.11539, 2020. 1
[68] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In Proceedings of the 25th international conference on Machine learning, pages 1096–1103, 2008. 3
[69] Chaoqi Wang, Guodong Zhang, and Roger Grosse. Picking winning tickets before training by preserving gradient flow. In International Conference on Learning Representations, 2020. 2, 3
[70] Daniel E Worrall, Stephan J Garbin, Daniyar Turmukhambetov, and Gabriel J Brostow. Harmonic networks: Deep translation and rotation equivariance. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5028–5037, 2017. 3
[71] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3733–3742, 2018. 3
[72] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07744, 2017. 4
[73] Shibui Yin, Kyu-Hyouk Kim, Jinwook Oh, Naigang Wang, Mauricio Serrano, Jae-Sun Seo, and Jungwook Choi. The sooner the better: Investigating structure of early winning lottery tickets, 2020. 3
[74] Haoran You, Chaojian Li, Pengfei Xu, Yonggan Fu, Yue Wang, Xiaohan Chen, Richard G. Baraniuk, Zhangyang Wang, and Yingyan Lin. Drawing early-bird tickets: Toward more efficient training of deep networks. In International Conference on Learning Representations, 2020. 2, 3, 4
[75] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. Advances in Neural Information Processing Systems, 33, 2020. 3
[76] Haoran Yu, Sergey Edunov, Yuandong Tian, and Ari S Morcos. Playing the lottery with rewards and multiple languages: lottery tickets in rl and nlp. arXiv preprint arXiv:1906.02768, 2019. 3
[77] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In European conference on computer vision, pages 649–666. Springer, 2016. 3
[78] Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1058–1067, 2017. 3
[79] Hao Zhou, Jose M Alvarez, and Fatih Porikli. Less is more: Towards compact cnns. In European Conference on Computer Vision, pages 662–677. Springer, 2016. 3
[80] Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. Local aggregation for unsupervised learning of visual embeddings. In Proceedings of the IEEE International Conference on Computer Vision, pages 6002–6012, 2019. 3
A. More Technical Details

A.1. Pruning Algorithm

Following the routines in previous LTH [26, 7] works, the algorithm 1 outlines the full iterative magnitude pruning (IMP) procedure.

Algorithm 1 Iterative Magnitude Pruning (IMP)

1: Set the initial mask to $m = 1^{d_1}$, with the pre-training $\theta_p$.
2: repeat
3: \hspace{1em} Train $f(x; m \odot \theta_p, \gamma_p)$ for $t$ epochs with algorithm $A^T$, i.e., $A^T (f(x; m \odot \theta_p, \gamma_p))$.
4: \hspace{1em} Prune 20\% of remaining weights in $A^T (f(x; m \odot \theta_p, \gamma_p))$ and update $m$ accordingly.
5: until the sparsity of $m$ reaches the desired sparsity level $s$
6: Return $f(x; m \odot \theta_p)$.

A.2. Top-1 Retrieval Accuracy

Here we presents the detailed calculation of top-1 retrieval accuracy for self-supervised pretraining tasks, including simCLR [8] and MoCo [34]. Given a batch of data with $n$ samples, \{ $z_1, \cdots, z_n$ \} and \{ $z'_1, \cdots, z'_n$ \} donates the feature representations from the two branches of simCLR or MoCo models. $z_i$ and $z'_i$ are computed from the same input sample with different data augmentations.

For each $z_i$, we calculate the cosine similarity between $z_i$ and other representations and obtain $D_i = \{ d(z_i, z) \mid z \in \{ z_j, z'_j \}_{j=1}^n \}$, where $d(\cdot, \cdot)$ is the cosine similarity measurement. If $\arg \max_{z \in D_i} D_i = z'_i$, it suggests the top-1 retrieval is correct. In the same way, we perform a similar retrieval process for $z'_i$ and $D'_i$. The concrete calculation formulation of top-1 retrieval accuracy is depicted as follows:

$$\sum_{i=1}^{n} \left[ \frac{\mathbb{I}(\arg \max_{z \in D_i} D_i = z'_i) + \mathbb{I}(\arg \max_{z \in D'_i} D'_i = z_i)}{2 \times n} \right] \times 100\%,$$

(2)

where $\mathbb{I}(\cdot)$ is the indicator function.

B. More Experimental Results

B.1. Detection Results with Other Metrics

In this section, we report the other two evaluation metrics, i.e., $AP_{50}$ and $AP_{75}$, for detection experiments. As shown in Figure 10, the most different observation is that there a subnetwork $f(x; m_{MoCo} \odot \theta_{MoCo}, \cdot)$ transfer to the detection task successfully (i.e., without performance degradation compared with full unpruned models) at the 59.04% sparsity level under the $AP_{50}$ metric. Other conclusions are consistent with the ones in the main text.

B.2. Ablation about Larger Pre-training Models

Figure 11 collects the pre-training task performance of subnetworks generated from small- and large-scale pre-trained simCLR models. We observe that heavily compressed, large simCLR models (e.g., ResNet-50) obtain superior performance to lightly compressed, small simCLR models (e.g., ResNet-152), which is consistent with [46]. However, subnetworks found on the small-scale pre-trained simCLR model show a slightly better top-1 retrieval accuracy after the sparsity approaches an extreme level.