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

Tianlong Chen\textsuperscript{1}, Jonathan Frankle\textsuperscript{2}, Shiyu Chang\textsuperscript{3}, Sijia Liu\textsuperscript{3,4}, Yang Zhang\textsuperscript{3}, Michael Carbin\textsuperscript{2}, Zhangyang Wang\textsuperscript{1}
\textsuperscript{1}University of Texas at Austin, \textsuperscript{2}MIT CSAIL \textsuperscript{3}MIT-IBM Watson AI Lab, \textsuperscript{4}Michigan State University
{tianlong.chen,atlaswang}@utexas.edu, {jfrankle,mcarbin}@csail.mit.edu, {shiyu.chang,yang.zhang2}@ibm.com, liusiji5@msu.ed

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 \cite{chen2020simple} and MoCo \cite{he2020moco}. Pre-trained weights often boost a wide range of downstream tasks including classification, detection, and segmentation. Latest studies suggest that pre-training benefits from gigantic model capacity \cite{lottery}. We are hereby curious and ask: after pre-training, does a pre-trained model indeed have to stay large for its downstream transferability?

In this paper, we examine supervised and self-supervised pre-trained models through the lens of the lottery ticket hypothesis (LTH) \cite{frankle2019lottery}. LTH identifies highly sparse matching subnetworks that can be trained in isolation from (nearly) scratch yet still reach the full models’ performance. We extend the scope of LTH and question whether matching subnetworks still exist in pre-trained computer vision 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. Further analyses reveal that subnetworks found from different pre-training tend to yield diverse mask structures and perturbation sensitivities. We conclude that the core LTH observations remain generally relevant in the pre-training paradigm of computer vision, but more delicate discussions are needed in some cases. Codes and pre-trained models will be made available at: https://github.com/VITA-Group/CV_LTH_Pre-training.

1. Introduction

Deep neural networks pre-trained on large-scale datasets prevail as general-purpose feature extractors \cite{he2016deep}. Moving beyond the most traditional greedy unsupervised pre-training \cite{hinton2006revisiting}, the most popular pre-training in computer vision (CV) is arguably to train the model for supervised classification on ImageNet \cite{deng2009imagenet}. Such supervised pre-training enables the network to learn a hierarchy of generalizable features \cite{vasudevan2019neural}; it is widely acknowledged \cite{olah2017a1} to not only benefit the subsequent fine-tuning on other visual classification datasets (especially in small datasets and few-shot learning \cite{wang2019closer,li2020class}), but also to accelerate/improve the training for different, more complicated types of downstream vision tasks, such as object detection and semantic segmentation \cite{zhong2020rethinking,lin2017mmdetection}.

Several state-of-the-art self-supervised pre-training, such as simCLR \cite{chen2020simple,lottery} and MoCo \cite{he2020moco,chen2020mocov3}, have demonstrated that it is instead possible to use unlabeled data in pre-training. Their methods refer to no actual labels in pre-training, but instead leverage self-generated pseudo labels \cite{caron2020unsupervised,chen2020simple} or contrasting augmented views \cite{chen2020simple}. Impressively, self-supervised pre-training yields pre-trained weights with comparable or even better transferability and generalization, for various downstream tasks, compared to their supervised pre-training counterparts.

A few recent efforts have shown to successfully scale up pre-training in CV. That is perhaps most natural for self-
supervised pre-training, since unlabeled images are cheap and easily accessible. Chen et al. [11] investigated to boost simCLR with massive unlabeled data in a task-agnostic way, and pointed out the key ingredient to be the use of big (deep and wide) networks during pretraining and fine-tuning. The authors found that, the fewer the labels, the more this approach (task-agnostic use of unlabeled data) benefits from a bigger network. After fine-tuning, the big network is reduced into a much smaller one with little performance loss by using task-specific distillation. We additionally note the latest works suggesting that supervised fine-tuning can also scale up to larger models and datasets beyond ImageNet [24].

The extraordinary cost of pre-training can be amortized by transferring to many downstream tasks. However, such explosive sizes of pre-trained models can even make fine-tuning computationally demanding, urging us to ask: can we aggressively trim down the complexity of pre-trained models, without damaging their downstream transferability? Note that, the question asked is drastically different from the conventional scope of model compression [38] in CV, where a model is trained, compressed and/or tuned on the same dataset and specific task. In comparison, any simplification for a pre-trained model has to ensure its intact transferability to a variety of possible downstream tasks.

To address this research gap, we turn our attention to lottery ticket hypothesis (LTH) [20, 27, 31, 50, 76, 81], a fast-rising field that investigates the sparse trainable subnetworks within full dense networks. The original LTH [31, 32] demonstrated small-scale networks contain sparse matching subnetworks capable of training in isolation from initialization to full accuracy. In other words, we could have trained smaller networks from the start if only we had known which subnetworks to choose. Recent investigations [59, 58] showed those matching subnetworks to transfer between related classification tasks. However, no study has closely examined the tantalizing possibility of universal transferability in LTH for CV models, i.e., if we treat the pre-trained weights as our initialization, whether matching subnetworks still exist in the pre-training models, that also enjoy the same downstream transfer performance? Are there universal subnetworks that can transfer to many tasks with no degradation in performance?

The paper carries out the first comprehensive experimental study to seek these desired universal matching subnetworks, from both supervised and self-supervised pre-trained CV models. 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 [31, 69], the matching subnetworks are found at an early point in training. 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 iterative unstructured magnitude pruning [31], we identify matching subnetworks 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 the same accuracies.
- Subnetworks at 73.79%/48.80%, 48.80%/36.00% and 73.79%/83.22% sparsity, found respectively by supervised ImageNet, simCLR and MoCo, can transfer to downstream detection/segmentation tasks without sacrificing performance.
- Unlike previous matching subnetworks found at random initialization or early in training, 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 large pre-trained models with smaller subnetworks, enabling much more efficient downstream tuning without inhibiting transfer performance. 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 [49]. Pruning algorithms can be grouped into unstructured [39, 49, 38] and structured [54, 43, 86]: the former sparsifies based on weight magnitudes; while the latter considers hardware-friendliness by removing channels and so on.

The discovery of LTH [31] 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 [55, 35] scale up LTH by rewinding approaches [33, 69], that re-initializes the subnetwork from
the early training stage checkpoint rather than from scratch. LTH has been widely explored in image classification [31, 55, 76, 28, 34, 72, 80, 81, 56, 15], natural language processing [17, 8], generative adversarial networks [18], graph neural networks [14], and reinforcement learning [83]. Most of them adopt (iterative) unstructured weight magnitude pruning [38, 31], [58, 59, 19] 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 [9] to ours is from the natural language processing (NLP) field: the authors found universally transferable sparse matching subnetworks (at 40% to 90% sparsity), from the pre-trained initialization of BERT models [21]. Finding their work inspiring, we stress that transplanting their NLP findings to our CV models is highly nontrivial due to multiple barriers: (1) pre-training BERT in [9] uses only a self-supervised objective called “masked language model” (MLM) [21], while pre-training CV models has a significant variety of popular options, ranging from the supervised fashion [46], to self-supervision yet with numerous objectives [22, 40, 10]; (2) BERT models consist of self-attention and fully-connected sub-layers, differing much from the standard convolutional architectures in CV; (3) further complicating the issue is that different CV downstream tasks are known to rely on different priors and invariances; for example, while classification often calls on shift invariance, detection assumes location shift equivariance [77, 57]. That questions the feasibility of asking for one mask to transfer among them all. Such complicity is well manifested by our delicate observations.

**Pre-training in Computer Vision.** Supervised ImageNet pre-training has been a main CV workhorse [36, 41]. The recent surge of self-supervised pre-training suggest the potential of unlabeled data; examples include recovering the artificially corrupted inputs [65, 75, 84, 85], predicting pseudo-labels [22, 25, 61, 64, 5, 6, 13], or contrasting augmented views [1, 44, 45, 62, 73, 78, 87, 10, 40, 11, 16, 47, 82]. The state-of-the-art simCLR [10, 11] and MoCo [40, 16] pre-training can reduce the amount of labels needed for tuning downstream image classifiers, by two magnitudes. Pre-trained networks are usually subsequently fine-tuned, with the architectures unchanged. One exception is [4] which is the first to adapt the backbone architecture to fit different target datasets. It pre-trains a large super-net that contains many weight-shared sub-nets that can individually operate.

### 3. Preliminaries and Setup

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

**Network.** We use the official ResNet-50 [42] 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 [10]. YOLOv4 [3] and DeepLabv3+ [7] 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 and all classification, detection, segmentation heads are never pruned. Specifically, we let $f(x; \theta, \gamma)$ be the output of a ResNet-50 model with parameters $\theta \in \mathbb{R}^{d1}$ (excluding the first convolution layer) and task-specific parameters $\gamma \in \mathbb{R}^{d2}$ on an input image $x$. **Pre-training.** For the supervised pre-training, we use the official pre-trained ResNet-50 on the ImageNet dataset [18]. For the self-supervised, we adopt the pre-trained ResNet-50 models with simCLR [10] and MoCoV2 [16] 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 [79], SVHN [60], CIFAR-10 [48], CIFAR-100 [48], and VisDA2017 [66]. These datasets vary remarkably in terms of sample size, color space, and number of classes.

---

1. For complicated CV tasks, the large variety of model design options may possibly impact our observation. For example, object detectors fall under two-stage and one-stage categories, the former often achieving higher accuracy while the latter typically being faster. YOLOv4 [3] is a popular one-stage detector. We also include the results for two popular two-stage detectors, Faster RCNN [68] and SSD [53], in Section B.2.
3. The official simCLR model zoo at [https://github.com/google-research/simclr](https://github.com/google-research/simclr)
4. The official MoCov2 model zoo at [https://github.com/facebookresearch/moco](https://github.com/facebookresearch/moco)
resolution, image source, and classes. Following [40, 16], we train object detection models on the combined training and validation set of Pascal VOC 2012 [29] and Pascal VOC 2007 [30], 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\(^5\) for all pre-training and downstream tasks, as in Table 1.

**Subnetworks.** For a network \(f(x; \theta, \cdot)\) with task-specific modules \(\gamma\), its subnetworks can be depicted as \(f(x; m \odot \theta, \cdot)\) with a pruning binary mask \(m \in \{0, 1\}^{d_1}\), where \(\odot\) is the element-wise product. Let \(A^T_t (f(x; \theta, \gamma))\) 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 [10] and MoCov2 [16]. Let \(\theta_t\) be the random initialization, and \(\theta_i\) be the network weights at the \(i_{\text{th}}\) epoch which is trained from \(\theta_0\). Let \(E^T (f(x; \theta, \gamma))\) be the evaluation function of model \(f\) returned from \(A^T_t\) on the corresponding task \(T\). Below we define:

1. **Matching subnetworks.** Following the definition in [32, 9], a subnetwork \(f(x; m \odot \theta, \gamma)\) is matching if it satisfies the following condition:

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

(1)

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

2. **Winning ticket.** If \(f(x; m \odot \theta, \gamma)\) is a matching subnetwork with \(\theta = \theta^p\) for \(A^T_t\), it is a winning ticket for \(A^T_t\).

3. **Universal subnetwork.** A subnetwork \(f(x; m \odot \theta, \gamma_T)\) 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 \(A^T_{t_i}\). The task set \(\{T_i\}_{i=1}^N\) could be a group of (diverse) downstream tasks, such as classification, detection and segmentation.

**Pruning 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 [31, 32, 9]. We prune the network by first training the unpruned dense network to completion on a task \(T\) (i.e., applying \(A^T_T\)) and then removing a portion of weights with the globally smallest magnitudes [38, 69]. 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.

\(^5\)For detection experiments, we report the other evaluation metrics, \(AP_{50}\) and \(AP_{75}\) [16] in the supplement. The technical details of calculating the retrieval accuracy for simCLR and MoCo pre-training tasks are also included in the supplement.

\(^6\)Early weight rewinding [69, 32] improves the quality of found matching subnetworks. As indicated by [32], the best rewinding points usually lie in the first 1% ~ 5% training epochs. We take 5% for default comparison.

![Figure 2. Performance of pre-training tasks on ImageNet. The masks \((m_{\text{img}}, m_{\text{sim}}\) and \(m_{\text{MoCo}}\) of evaluated subnetworks are found on supervised, simCLR and MoCo pre-training tasks respectively by IMP. \(\theta_{\text{img}}\) = the pre-trained weights of the supervised ImageNet classification; \(\theta_{\text{sim}}\) = the pre-trained weights of simCLR [10]; \(\theta_{\text{MoCo}}\) = the pre-trained weights of MoCov2 [16]. 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 [26] has already shown significant speedups over dense baselines on smartphone processors. Integrating structured pruning will be another future direction of our interest [81].

### 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% sparsity 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 \(f(x; m_\mathcal{T} \odot \theta_{\mathcal{T}}, \cdot)\), found on the pre-training task \(\mathcal{T}\), also winning tickets for other downstream tasks \(\mathcal{P}\)?

**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 \(f(x; m_{\mathcal{T}} \odot \theta_{\mathcal{T}}, \cdot)\) outperform the subnetworks \(f(x; m_{\mathcal{T}} \odot \theta_{\mathcal{T}}, \cdot)\) \((\theta_{\mathcal{P}_i} \in \{\theta_0, \theta_{\mathcal{P}_i}, \theta_{\mathcal{P}_i} \odot, \theta_{\mathcal{P}_i} \cdot\})\), found on a specific task \(\mathcal{T}\)?

### 4.1. Transfer to Classification Tasks

As shown in Figures 3 and 4, evaluated subnetworks are divided into three groups, according to sources of

...
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. \((\text{m}_\text{Img}, \theta_{\text{Img}})\), \((\text{m}_\text{sim}, \theta_{\text{sim}})\) and \((\text{m}_\text{MoCo}, \theta_{\text{MoCo}})\) denote transfer performance of subnetworks found at pre-training tasks. Subnetworks with \((\text{m}_T, \theta_p)\), \(T \in \{\text{CIFAR-10}, \text{CIFAR-100}, \text{SVHN}, \text{Fashion-MNIST}\}\) and \(\theta_p \in \{\theta_{\text{Img}}, \theta_{\text{sim}}, \theta_{\text{MoCo}}\}\) are identified on the downstream task \(T\) with pre-trained weights \(\theta_p\). Subnetworks \((\text{m}_T, \theta_0)\) and \((\text{m}_T, \theta_{5\%})\) are found on the task \(T\) with the random initialization \(\theta_0\) and an early rewinding weights \(\theta_{5\%}\). 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. \((\text{m}, \theta)\): i) transferred subnetworks with \((\text{m}_p, \theta_p), P \in \{\text{Img}, \text{sim}, \text{MoCo}\}\) 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 \((\text{m}_T, \theta_p)\), \(T \in \{\text{CIFAR-10}, \text{CIFAR-100}, \text{SVHN}, \text{Fashion-MNIST}, \text{VisDA2017}\}\); iii) subnetworks consists of \((\text{m}_T, \theta_i), \theta_i \in \{\theta_0, \theta_{5\%}\}\), identified with the original random initialization \(\theta_0\) or early rewinding weights \(\theta_{5\%}\) on downstream tasks \(T\). Summarizing all comprehensive results, our main observations are:

**A1: Subnetworks with \((\text{m}_p, \theta_p)\) 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 \(f(x; \text{m}_\text{Img} \odot \theta_{\text{Img}}, \cdot), f(x; \text{m}_\text{sim} \odot \theta_{\text{sim}}, \cdot), f(x; \text{m}_\text{MoCo} \odot \theta_{\text{MoCo}}, \cdot)\) transfer without sacrificing performance\(^7\) 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 \( f(x; m_{\text{ImageNet}} \odot \theta_{\text{ImageNet}}, \cdot) \) at the low and middle level sparsity of subnetworks. However, the conclusions are almost flipped when transferring \( f(x; m_{T} \odot \theta_{T}, \cdot) \) to the synthetic VisDA2017 dataset. Subnetworks \( f(x; m_{\text{ImageNet}} \odot \theta_{\text{ImageNet}}, \cdot) \) surpass others with a large performance margin, at the sparsity from 0.00% to 89.26%. For the extreme sparsity, the MoCo pre-training task generates a better transferable subnetwork. 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-training (e.g., simCLR and MoCo) produce more transferable subnetworks especially at the extreme sparsity, when natural image datasets are at downstream.

A3: Transferred subnetworks \( f(x; m_{T} \odot \theta_{T}, \cdot) \) perform the best until extreme sparsity. Subnetworks \( f(x; m_{\text{MoCo}} \odot \theta_{\text{MoCo}}, \cdot) \) found on a specific downstream task with pre-trained weights, can be considered as “performance upbound” for all our IMP subnetworks. \( f(x; m_{T} \odot \theta_{T}, \cdot) \) 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) \( f(x; m_{\text{ImageNet}} \odot \theta_{\text{ImageNet}}, \cdot) \) and \( f(x; m_{\text{sim}} \odot \theta_{\text{sim}}, \cdot) \) match the corresponding \( f(x; m_{T} \odot \theta_{T}, \cdot) \) with at most 59.04% sparsity; ii) On the natural image datasets, \( f(x; m_{\text{MoCo}} \odot \theta_{\text{MoCo}}, \cdot) \) steadily outperform \( f(x; m_{T} \odot \theta_{T}, \cdot) \) 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 \( \theta_{T} \) and \( \theta_{5\%} \) are inferior on all downstream tasks, compared to subnetworks with pre-trained initialization \( \theta_{T} \).

4.2. Transfer to Detection and Segmentation

Training detection and segmentation models commonly starts from pre-trained initializations [63, 41, 16]. We compare the transferred subnetworks with \( (m_{T}, \theta_{T}) \) versus the downstream task subnetworks with \( (m_{T}, \theta_{p}) \), as shown in Figure 5. Observations are organized as follows:

A1: Subnetworks \( f(x; m_{T} \odot \theta_{T}, \cdot) \) transfer to the detection and segmentation tasks successfully. Figure 5 demonstrates it is manageable to find transferable winning tickets on the detection and segmentation with the sparsity (73.79%, 48.80%, 73.79%) and (48.80%, 36.00%, 83.22%) for supervised ImageNet pre-training, self-supervised simCLR and MoCo pre-training tasks respectively.

A2: Unlike classification, winning tickets 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: \( \mathcal{E}^{T}(f(x; m_{\text{MoCo}} \odot \theta_{\text{MoCo}}, \cdot)) > \mathcal{E}^{T}(f(x; m_{\text{ImageNet}} \odot \theta_{\text{ImageNet}}, \cdot)) > \mathcal{E}^{T}(f(x; m_{\text{sim}} \odot \theta_{\text{sim}}, \cdot)), \) \( T \in \{\text{detection, segmentation}\} \). It suggests that MoCo pre-trained weights are most favorable for transferring to detection and segmentation tasks [16].

A3: Subnetworks \( f(x; m_{T} \odot \theta_{T}, \cdot) \) surpass subnetworks \( f(x; m_{p} \odot \theta_{p}, \cdot) \) by a non-negligible margin. As shown in Figure 5, with the assistance from the pre-trained initialization \( (\theta_{\text{ImageNet}}, \theta_{\text{sim}}, \theta_{\text{MoCo}}) \), 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 \( (m_{p}, \theta_{p}) \).

5. Analyzing Properties of Pre-training Tickets

5.1. Comparing Masks from Different Pre-trainings

In Figure 6, we compare the overlap in sparsity patterns found for different pre-training tasks. Relative similarity (i.e., \( m_{i} \odot m_{j} \) in [9]) are reported, which reflects the overlap degree between hamming masks \( m_{i} \) and \( m_{j} \), where \( i, j \in \{\text{ImageNet}, \text{simCLR}, \text{MoCo}\} \). We find that subnetworks for pre-training tasks are remarkably heterogeneous: they share less than 6.65% 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 struc-
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 [31, 32]. These two configurations produce matching subnetworks with diverse behaviors, including generalization performance and the structure sensitivity of obtained masks. We perform 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 three 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 degraded the performance w.r.t. the IMP masks. A possible explanation is that the pre-trained initializations are already highly structured, and perturbations can destroy the intrinsic structure. As evidenced by the right subfigure of Figure 8, subnetworks with $(m_{\text{CIFAR}-100}, \theta_{\text{img}})$ are no better than subnetwork with $(m_{\text{CIFAR}-100}, \theta_0)$. It shows that pre-training with damaged weight distributions no longer leads to the generalization gains.

- 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.
Subnetworks with the Original Random Initialization  
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 $\theta_{0\%}$, and the pre-trained weights $\theta_{pre}$. $m_{CIFAR-100}$ = masks found by IMP; $m_{CIFAR-100}^\perp$ = the complementary masks of $m_{CIFAR-100}$, where $m \cap m^\perp = 0$ and $m \cup m^\perp = 1 \in \mathbb{R}^d$; $m_r$ = 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.

5.3. More Ablation Studies for Pre-training

Larger Pre-training Model? [52] 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\(^8\) of found matching subnetworks.

In Figure 9, with the same number of remaining weights, subnetworks pruned from simCLR\(^9\) 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 [11], to first pretrain a big model and then compress it. The key difference is that, [11] 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 [10, 11, 12]. 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 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. Finally, 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).

Table 2. Ablation study of temperature parameter in simCLR. Transfer performance (i.e., accuracy) of subnetworks ($m_{sim}, \theta_{sim}$) with 73.79% sparsity on CIFAR-100 downstream task.

<table>
<thead>
<tr>
<th>Temperature</th>
<th>0.1</th>
<th>0.2</th>
<th>0.5</th>
<th>1.0</th>
<th>2.0</th>
<th>10.0</th>
<th>20.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy (%)</td>
<td>81.81</td>
<td>81.91</td>
<td><strong>82.22</strong></td>
<td>81.24</td>
<td>80.76</td>
<td>81.46</td>
<td>80.18</td>
</tr>
</tbody>
</table>

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.

\(^8\)In the supplement, we also report the pre-training task performance of subnetworks generated from small- and large-scale pre-trained simCLR.

\(^9\)For a fair comparison, here we adopt the simCLRv2 [11] pre-trained ResNet-152 and ResNet-50 models, since only simCLRv2 released the official pre-trained ResNet-152 model.

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.

Acknowledgments

We are grateful for the MIT-IBM Watson AI Lab, in particular John Cohn for generously providing 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


[59] Ari Morcos, Haoran Yu, Michela Paganini, and Yuandong Tian. One ticket to win them all: generalizing lottery ticket