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

Tianlong Chen[1], Jonathan Frankle[2], Shiyu Chang[3], Sijia Liu[3,4], Yang Zhang[3],
Michael Carbin[2], Zhangyang Wang[1]
[1]University of Texas at Austin, [2]MIT CSAIL [3]MIT-IBM Watson AI Lab, [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 [10] and MoCo [40]. 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 [11]. 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) [31]. 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 [23]. Moving beyond the most traditional greedy unsupervised pre-training [2], the most popular pre-training in computer vision (CV) is arguably to train the model for supervised classification on ImageNet [18]. Such *supervised pre-training* enables the network to learn a hierarchy of generalizable features [46]; it is widely acknowledged [36] to not only benefit the subsequent fine-tuning on other visual classification datasets (especially in small datasets and few-shot learning [71, 74]), but also to accelerate/improve the training for different, more complicated types of downstream vision tasks, such as object detection and semantic segmentation [63, 41].

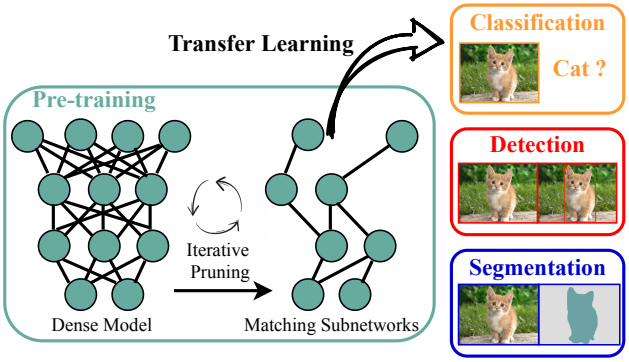

Figure 1. Overview of our work paradigm: from pre-trained CV models (both supervised and self-supervised), we study the existence of matching subnetworks that are transferable to many downstream tasks, with little performance degradation compared to using full pre-trained weights. We find *task-agnostic,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

Several state-of-the-art *self-supervised pre-training*, such as simCLR [10, 11] and MoCo [40, 16], 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 [22, 25] or contrasting augmented views [10]. 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 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 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

Table 1. Details of pre-training and fine-tuning. We use the default implementations and hyperparameters [69, 10, 40, 16, 7, 51, 3]. The evaluation metrics also follow the standards [69, 16, 7, 3]. For the supervised learning, we use the training dataset to name the corresponding task for the same of simplicity, e.g. "ImageNet" represents the supervised pre-training classification task on ImageNet.

Settings	Pre-training			Downstream Classification					Downstream Detection	Downstream Segmentation
	ImageNet	simCLR	MoCov2	CIFAR-10	CIFAR-100	SVHN	Fashion-MNIST	VisDA2017	Pascal VOC2012/2007	Pascal VOC 2012
# Epochs/Iters	10	10	10	182	182	182	182	20	50 Epochs/103K Iters	30K Iters
Batch Size	256	256	256	256	256	256	256	128	8	4
Learning Rate	0.0001	0.0001	0.0003	0.1	0.1	0.1	0.1	0.001	0.0001	0.01
	Fixed schedule			×0.1 at 91,136 epoch				×0.1 at 10 epoch	Cosine decay from 10^{-4} to 10^{-6}	Linear warmup 100 Iters ×0.1 at 18K, 22K Iters
Optimizer	SGD [70] with 0.9 momentum									
Weight Decay	1×10^{-4}	1×10^{-4}	1×10^{-4}	2×10^{-4}	2×10^{-4}	2×10^{-4}	2×10^{-4}	5×10^{-4}	5×10^{-4}	1×10^{-4}
Eval. Metric	Accuracy	Retrieval Accuracy		Accuracy	Accuracy	Accuracy	Accuracy	Accuracy	AP, AP_{50}, AP_{75}	mIOU

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 [35, 83, 69, 67, 9], generative adversarial networks [17, 8], 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 complicacy 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 Setups

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[1]. 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 $\theta \in \mathbb{R}^{d_1}$ (excluding the first convolution layer) and task-specific parameters $\gamma \in \mathbb{R}^{d_2}$ on an input image x.

Pre-training. For the supervised pre-training, we use the official pre-trained ResNet-50[2] on the ImageNet dataset [18]. For the self-supervised, we adopt the pre-trained ResNet-50 models with simCLR[3] [10] and MoCov2[4] [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,

[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.

[2]The official Pytorch model zoo at https://pytorch.org/docs/stable/torchvision/models.html

[3]The official simCLR model zoo at https://github.com/google-research/simclr

[4]The official MoCov2 model zoo at 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 γ, 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 $\mathcal{A}_t^{\mathcal{T}}(f(x;\theta,\gamma))$ be a training algorithm (e.g., SGD with certain hyperparameters) that trains a network $f(x;\theta,\gamma)$ on a task $\mathcal{T}$ (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 θ_{Img} is the supervised pre-trained weight, θ_{sim} and θ_{MoCo} are from the self-supervised pre-training by simCLR [10] and MoCov2 [16]. Let θ_0 be the random initialization, and θ_i be the network weights at the i_{th} epoch which is trained from θ_0. Let $\mathcal{E}^{\mathcal{T}}(f(x;\theta,\gamma))$ be the evaluation function of model f returned from $\mathcal{A}_t^{\mathcal{T}}$ on the corresponding task $\mathcal{T}$. 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:

$$\mathcal{E}^{\mathcal{T}}\left(\mathcal{A}_t^{\mathcal{T}}\left(f\left(x;m\odot\theta,\gamma\right)\right)\right)\geq\mathcal{E}^{\mathcal{T}}\left(\mathcal{A}_t^{\mathcal{T}}\left(f\left(x;\theta_p,\gamma\right)\right)\right)\quad(1)$$

That is, matching subnetworks perform no worse than the full dense models under the same training algorithm $\mathcal{A}_t^{\mathcal{T}}$ and evaluation metric $\mathcal{E}^{\mathcal{T}}$.

2. Winning ticket. If $f(x;m\odot\theta,\gamma)$ is a matching subnetwork with $\theta=\theta_p$ for $\mathcal{A}_t^{\mathcal{T}}$, it is a *winning ticket* for $\mathcal{A}_t^{\mathcal{T}}$.

3. Universal subnetwork. A subnetwork $f(x;m\odot\theta,\gamma_{\mathcal{T}_i})$ with task-specific configurations of $\gamma_{\mathcal{T}_i}$, is *universal* for tasks $\{\mathcal{T}_i\}_{i=1}^N$ if and only if it is matching for each $\mathcal{A}_{t_i}^{\mathcal{T}_i}$. The task set $\{\mathcal{T}_i\}_{i=1}^N$ 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 $\mathcal{T}$ (i.e., applying $\mathcal{A}_t^{\mathcal{T}}$) 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.

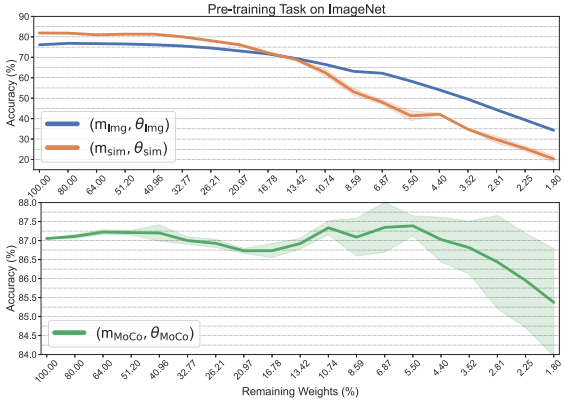

Figure 2. Performance of pre-training tasks on ImageNet. The masks (m_{Img}, m_{sim} and m_{MoCo}) of evaluated subnetworks are found on supervised, simCLR and MoCo pre-training tasks respectively by IMP. θ_{Img} = the pre-trained weights from the supervised ImageNet classification; θ_{sim} = the pre-trained weights of simCLR [10]; θ_{MoCo} = the pre-trained weights of MoCo [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 $f(x;m_{\mathcal{P}}\odot\theta_p,\cdot)$, found on the pre-training task $\mathcal{P}$, also winning tickets for other downstream tasks $\mathcal{T}$?

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 $f(x;m_{\mathcal{P}}\odot\theta_p,\cdot)$ outperform the subnetworks $f(x;m_{\mathcal{T}}\odot\theta_i,\cdot)$ ($\theta_i\in\{\theta_0,\theta_{5\%}{}^{[6]},\theta_p\}$), 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

[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 $1\%\sim5\%$ training epochs. We take 5% for default comparison.

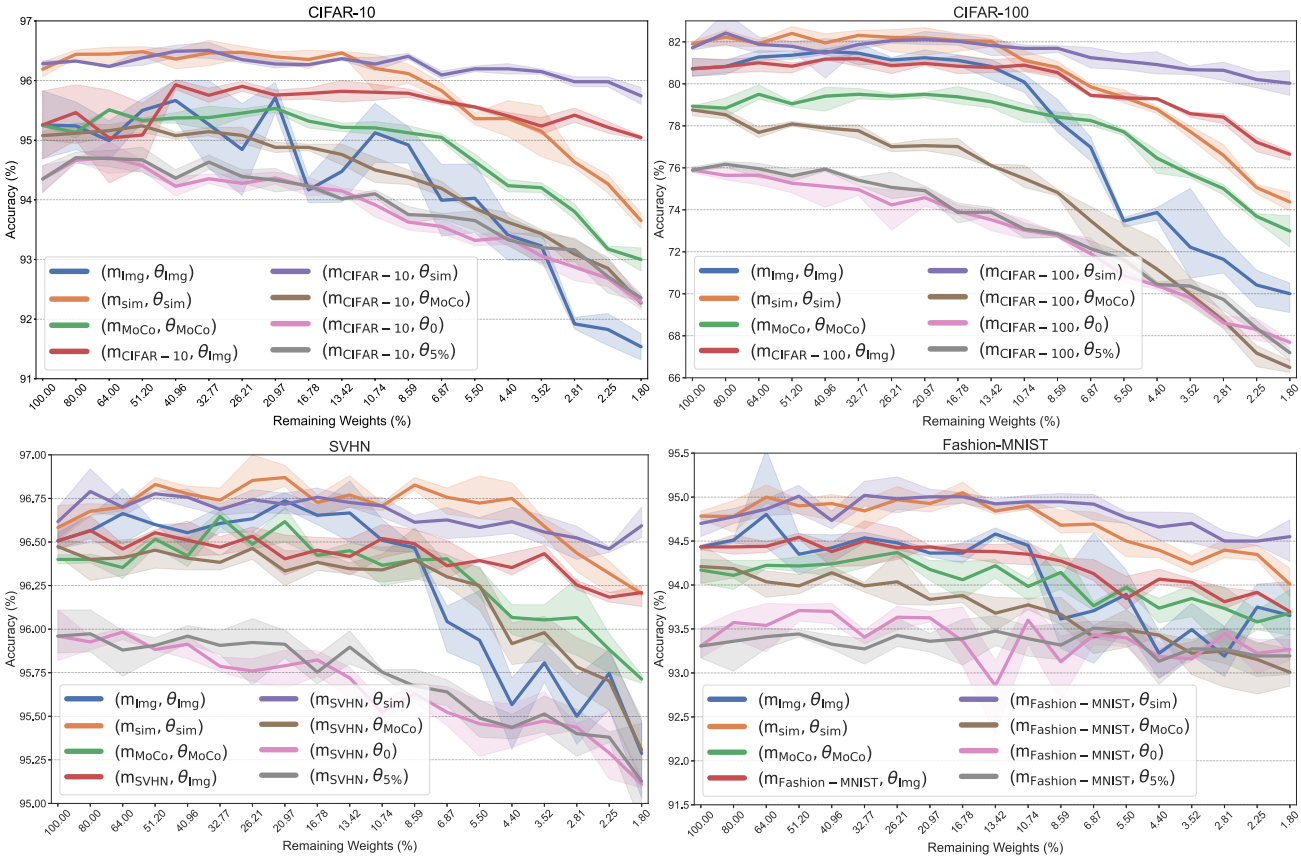

Figure 3. Performance of IMP subnetworks with a range of sparsity from 0.00% to 98.20% (i.e., remaining weight from 100% to 1.80%) on downstream classification tasks, including CIFAR-10, CIFAR-100, SVHN and Fashion-MNIST. $(m_{\text{Img}}, \theta_{\text{Img}})$, $(m_{\text{sim}}, \theta_{\text{sim}})$ and $(m_{\text{MoCo}}, \theta_{\text{MoCo}})$ denote transfer performance of subnetworks found at pre-training tasks. Subnetworks with $(m_{\mathcal{T}_i}, \theta_p)$, $\mathcal{T}_i \in$ {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 $\mathcal{T}_i$ with pre-trained weights θ_p. Subnetworks $(m_{\mathcal{T}_i}, \theta_0)$ and $(m_{\mathcal{T}_i}, \theta_{5\%})$ are found on the task $\mathcal{T}_i$ with the random initialization θ_0 [31] and an early rewinding weights $\theta_{5\%}$ [69]. Curves with errors (shadow regions) are the average across three independent runs, with the standard deviations: same hereinafter.

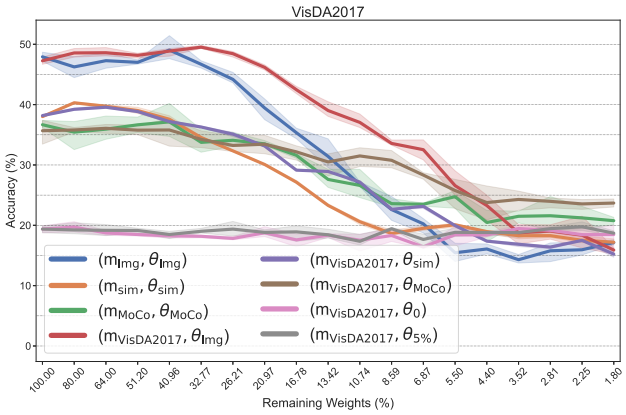

Figure 4. Performance of IMP subnetworks with a range of sparsity from 0.00% to 98.20% on the synthetic dataset, VisDA2017.

(m, θ): i) transferred subnetworks with $(m_{\mathcal{P}}, \theta_p)$, $\mathcal{P} \in \{\text{Img}, \text{sim}, \text{MoCo}\}$ 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 $(m_{\mathcal{T}}, \theta_p)$, $\mathcal{T} \in$ {CIFAR-10, CIFAR-100, SVHN, Fashion-MNIST, VisDA2017}; iii) subnetworks consists of $(m_{\mathcal{T}}, \theta_i)$, $\theta_i \in \{\theta_0, \theta_{5\%}\}$, identified with the original random initialization θ_0 or early rewinding weights $\theta_{5\%}$ on downstream tasks $\mathcal{T}$. Summarizing all comprehensive results, our main observations are:

A1: Subnetworks with $(m_{\mathcal{P}}, \theta_p)$ 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[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

[7]Practically, to account for random fluctuations, we consider a subnetwork to be a winning ticket as long as its performance is within one standard deviation of the unpruned dense model.

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 $f(x; m_{\mathcal{P}} \odot \theta_p, \cdot)$ 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, the 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 $f(x; m_{\mathcal{P}} \odot \theta_p, \cdot)$ perform the best until extreme sparsity. Subnetworks $f(x; m_{\mathcal{T}} \odot \theta_p, \cdot)$, found on a specific downstream task with pre-trained weights, can be considered as *"performance upbound"* for all our IMP subnetworks. $f(x; m_{\mathcal{T}} \odot \theta_p, \cdot)$ 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 $f(x; m_{\mathcal{T}} \odot \theta_p, \cdot)$ with at most 59.04% sparsity; ii) On the natural image datasets, $f(x; m_{\text{MoCo}} \odot \theta_{\text{MoCo}}, \cdot)$ steadily outperform $f(x; m_{\mathcal{T}} \odot \theta_p, \cdot)$ 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 θ_0 and $\theta_{5\%}$ are inferior on all downstream tasks, compared to subnetworks with pre-trained initialization θ_p.

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 $(m_{\mathcal{P}}, \theta_p)$ versus the downstream task subnetworks with $(m_{\mathcal{T}}, \theta_p)$, as shown in Figure 5. Observations are organized as follows:

A1: Subnetworks $f(x; m_{\mathcal{P}} \odot \theta_p, \cdot)$ 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: $\mathcal{E}^{\mathcal{T}}(f(x; m_{\text{MoCo}} \odot \theta_{\text{MoCo}}, \cdot)) > \mathcal{E}^{\mathcal{T}}(f(x; m_{\text{Img}} \odot \theta_{\text{Img}}, \cdot)) > \mathcal{E}^{\mathcal{T}}(f(x; m_{\text{sim}} \odot \theta_{\text{sim}}, \cdot))$, $\mathcal{T} \in \{\text{detection}, \text{segmentation}\}$. It suggests that MoCo pre-trained weights are most favorable for transferring to detection and segmentation tasks [16].

A3: Subnetworks $f(x; m_{\mathcal{T}} \odot \theta_p, \cdot)$ surpass subnetworks $f(x; m_{\mathcal{P}} \odot \theta_p, \cdot)$ by a non-negligible margin. As shown in Figure 5, with the assistance from the pre-trained initialization (θ_{Img}, θ_{sim}, θ_{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 $(m_{\mathcal{P}}, \theta_p)$.

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., $\frac{m_i \cap m_j}{m_i \cup m_j}$ in [9]) are reported, which reflects the overlap degree between hamming masks m_i and m_j, where $i, j \in \{\text{Img}, \text{sim}, \text{MoCo}\}$. 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 struc-

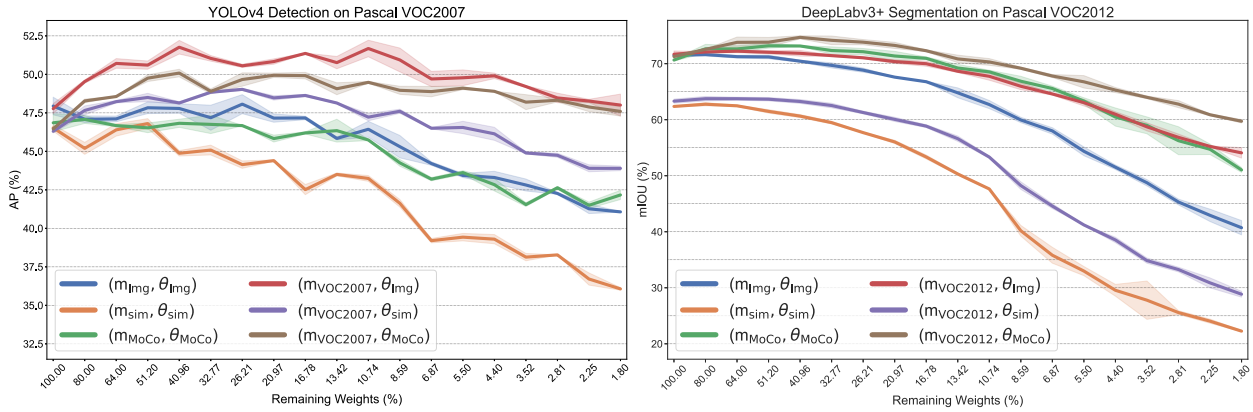

Figure 5. Performance of IMP subnetworks with a range of sparsity from 0.00% to 98.20% on the downstream detection and segmentation tasks. Subnetworks with $(m_{\text{VOC2007}}, \theta_p)$ and $(m_{\text{VOC2012}}, \theta_p)$, $\theta_p \in \{\theta_{\text{Img}}, \theta_{\text{sim}}, \theta_{\text{MoCo}}\}$ are identified on the downstream detection and segmentation tasks with pre-trained weights θ_p, respectively. The standard deviations are around $0.1\% \sim 2.5\%$ AP/mIOU.

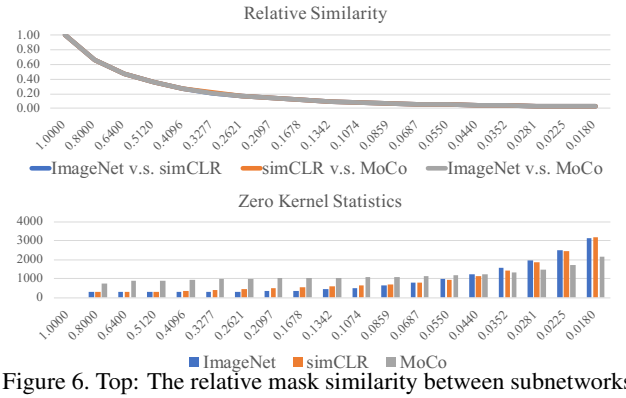

Figure 6. Top: The relative mask similarity between subnetworks which identified on supervised ImageNet, simCLR and MoCo pre-training tasks. Bottom: The number of completely pruned (zero) kernels in subnetworks found on different pre-training tasks.

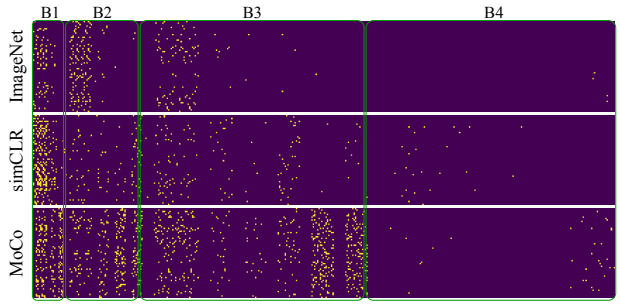

Figure 7. Kernel-wise heatmap visualizations of subnetworks with 79.03% sparsity found on supervised ImageNet, simCLR and MoCo pre-training tasks. From left to right, we visualization all kernels of subnetworks from the input to the output layers. The bright dots (●) represent the completely pruned (zero) kernels and the dark dots (●) the kernels having at least one unpruned weight. B1∼B4 donate four residual blocks in the ResNet-50 backbone.

tured sparse subnetworks (i.e., more all-zero kernels) may have a stronger potential for hardware speedup [26].

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 θ_0, early rewinding weights $\theta_{5\%}$, and the pre-trained weights θ_{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 $m^c_{\text{CIFAR}-100}$, 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 θ_0 or $\theta_{5\%}$, identified subnetworks are resilient to structure perturbations. In other words, there only exist marginal performance differences across subnetworks with masks $m_{\text{CIFAR}-100}$, $m^c_{\text{CIFAR}-100}$, 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 $(m^c_{\text{CIFAR}-100}, \theta_{\text{Img}})$ 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.

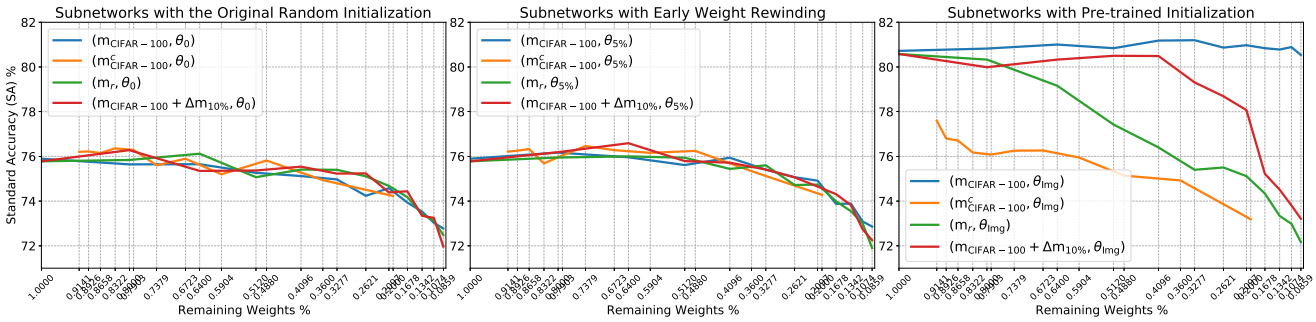

Figure 8. Performance comparison across subnetworks found on CIFAR-100 with the original random initialization θ_0, early rewinding weight $\theta_{5\%}$, and the pre-trained weights θ_{Img}. $m_{\text{CIFAR}-100}$ = masks found by IMP; $m^c_{\text{CIFAR}-100}$ = the complementary masks of $m_{\text{CIFAR}-100}$, where $m \cap m^c = \emptyset$ and $m \cup m^c = \mathbf{1} \in \mathbb{R}^{d_1}$; m_r = random pruned mask; $\Delta m_{10\%}$ = mask perturbations by randomly flipping 10% "1" and 10% "0" in the mask $m \in \{0,1\}^{d_1}$ 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.

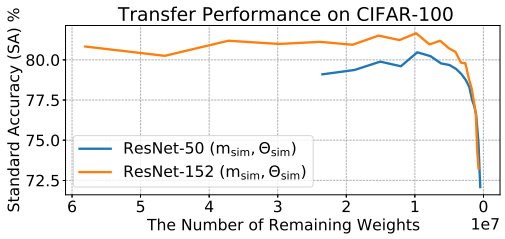

Figure 9. Transfer performance on CIFAR-100 over the number of remaining weights. Subnetworks are found on the simCLR pre-training task with pre-trained ResNet-50 and ResNet-152 weights.

[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.

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 $(m_{\text{sim}}, \theta_{\text{sim}})$ with 73.79% sparsity on CIFAR-100 downstream task.

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

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 complicacy 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 fpr 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

[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] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020. 3

[4] 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

[5] 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

[6] 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*, pages 2959–2968, 2019. 3

[7] 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

[8] Tianlong Chen, Yu Cheng, Zhe Gan, Jingjing Liu, and Zhangyang Wang. Ultra-data-efficient gan training: Drawing a lottery ticket first, then training it toughly. *arXiv preprint arXiv:2103.00397*, 2021. 3

[9] 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, 6, 12

[10] 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, 3, 4, 8, 12

[11] 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

[12] Ting Chen and Lala Li. Intriguing properties of contrastive losses, 2020. 8

[13] 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

[14] Tianlong Chen, Yongduo Sui, Xuxi Chen, Aston Zhang, and Zhangyang Wang. A unified lottery ticket hypothesis for graph neural networks, 2021. 3

[15] Tianlong Chen, Zhenyu Zhang, Sijia Liu, Shiyu Chang, and Zhangyang Wang. Long live the lottery: The existence of winning tickets in lifelong learning. In *ICLR*, 2021. 3

[16] 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

[17] Xuxi Chen, Zhenyu Zhang, Yongduo Sui, and Tianlong Chen. Gans can play lottery tickets too. In *ICLR*, 2021. 3

[18] 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, 3

[19] Shrey Desai, Hongyuan Zhan, and Ahmed Aly. Evaluating lottery tickets under distributional shifts. *arXiv preprint arXiv:1910.12708*, 2019. 3

[20] Tim Dettmers and Luke Zettlemoyer. Sparse networks from scratch: Faster training without losing performance. *arXiv preprint arXiv:1907.04840*, 2019. 2

[21] 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. 3

[22] 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. 1, 3

[23] 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

[24] 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

[25] 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. 1, 3

[26] 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

[27] 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

[28] Utku Evci, Fabian Pedregosa, Aidan Gomez, and Erich Elsen. The difficulty of training sparse neural networks. *arXiv preprint arXiv:1906.10732*, 2019. 3

[29] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015. 4

[30] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object

classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 4

[31] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations*, 2019. 1, 2, 3, 4, 5, 7, 12

[32] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. *arXiv preprint arXiv:1912.05671*, 2019. 2, 4, 7

[33] Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M Roy, and Michael Carbin. The lottery ticket hypothesis at scale. *arXiv preprint arXiv:1903.01611*, 2019. 2

[34] Jonathan Frankle, David J. Schwab, and Ari S. Morcos. The early phase of neural network training. In *International Conference on Learning Representations*, 2020. 3

[35] Trevor Gale, Erich Elsen, and Sara Hooker. The state of sparsity in deep neural networks. *arXiv preprint arXiv:1902.09574*, 2019. 2, 3

[36] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 1, 3

[37] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020. 13

[38] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015. 2, 3, 4

[39] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1135–1143. Curran Associates, Inc., 2015. 2

[40] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. 1, 3, 4, 12

[41] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *Proceedings of the IEEE international conference on computer vision*, pages 4918–4927, 2019. 1, 3, 6

[42] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[43] Yihui He, Xiangyu Zhang, and Jian Sun. Channel pruning for accelerating very deep neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1389–1397, 2017. 2

[44] Olivier J Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272*, 2019. 3

[45] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*, 2018. 3

[46] Minyoung Huh, Pulkit Agrawal, and Alexei A Efros. What makes imagenet good for transfer learning? *arXiv preprint arXiv:1608.08614*, 2016. 1, 3

[47] Ziyu Jiang, Tianlong Chen, Ting Chen, and Zhangyang Wang. Robust pre-training by adversarial contrastive learning. *Advances in Neural Information Processing Systems*, 33, 2020. 3

[48] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009. 3

[49] Yann LeCun, John S Denker, and Sara A Solla. Optimal brain damage. In *Advances in neural information processing systems*, pages 598–605, 1990. 2

[50] Namhoon Lee, Thalaiyasingam Ajanthan, and Philip Torr. Snip: Single-shot network pruning based on connection sensitivity. In *International Conference on Learning Representations*, 2019. 2

[51] Seungmin Lee, Dongwan Kim, Namil Kim, and Seong-Gyun Jeong. Drop to adapt: Learning discriminative features for unsupervised domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 91–100, 2019. 3

[52] Zhuohan Li, Eric Wallace, Sheng Shen, Kevin Lin, Kurt Keutzer, Dan Klein, and Joseph E Gonzalez. Train large, then compress: Rethinking model size for efficient training and inference of transformers. *arXiv preprint arXiv:2002.11794*, 2020. 8, 14

[53] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 3, 12

[54] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2736–2744, 2017. 2

[55] Zhuang Liu, Mingjie Sun, Tinghui Zhou, Gao Huang, and Trevor Darrell. Rethinking the value of network pruning. *arXiv preprint arXiv:1810.05270*, 2018. 2, 3

[56] Haoyu Ma, Tianlong Chen, Ting-Kuei Hu, Chenyu You, Xiaohui Xie, and Zhangyang Wang. Good students play big lottery better. *arXiv*, abs/2101.03255, 2021. 3

[57] Marco Manfredi and Yu Wang. Shift equivariance in object detection. *arXiv preprint arXiv:2008.05787*, 2020. 3

[58] Rahul Mehta. Sparse transfer learning via winning lottery tickets. *arXiv preprint arXiv:1905.07785*, 2019. 2, 3

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

initializations across datasets and optimizers. In *Advances in Neural Information Processing Systems*, pages 4932–4942, 2019. 2, 3

[60] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. 3

[61] 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

[62] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 3

[63] 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

[64] 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

[65] 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

[66] 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. 3

[67] Sai Prasanna, Anna Rogers, and Anna Rumshisky. When bert plays the lottery, all tickets are winning. *arXiv preprint arXiv:2005.00561*, 2020. 3

[68] 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, 12

[69] Alex Renda, Jonathan Frankle, and Michael Carbin. Comparing rewinding and fine-tuning in neural network pruning. In *International Conference on Learning Representations*, 2020. 2, 3, 4, 5

[70] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016. 3

[71] 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

[72] Pedro Savarese, Hugo Silva, and Michael Maire. Winning the lottery with continuous sparsification, 2020. 3

[73] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019. 3

[74] 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

[75] 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

[76] 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

[77] 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

[78] 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

[79] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 3

[80] Shihui Yin, Kyu-Hyoun 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

[81] 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

[82] 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

[83] Haonan 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

[84] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016. 3

[85] 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

[86] 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. 2

[87] 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