SuperTickets: Drawing Task-Agnostic Lottery Tickets from Supernets via Jointly Architecture Searching and Parameter Pruning

Haoran You[1⋆], Baopu Li[2,3], Zhanyi Sun[1], Xu Ouyang[1], and Yingyan Lin[1]

[1]Rice University [2]Baidu USA [3]Oracle Corporation
{haoran.you,zs19,xo2,yingyan.lin}@rice.edu, baopu.li@oracle.com

Abstract. Neural architecture search (NAS) has demonstrated amazing success in searching for efficient deep neural networks (DNNs) from a given supernet. In parallel, lottery ticket hypothesis has shown that DNNs contain small subnetworks that can be trained from scratch to achieve a comparable or even higher accuracy than the original DNNs. As such, it is currently a common practice to develop efficient DNNs via a pipeline of first search and then prune. Nevertheless, doing so often requires a tedious and costly process of search-train-prune-retrain and thus prohibitive computational cost. In this paper, we discover for the first time that both efficient DNNs and their lottery subnetworks (i.e., lottery tickets) can be directly identified from a supernet, which we term as **SuperTickets**, via a two-in-one training scheme with jointly architecture searching and parameter pruning. Moreover, we develop a progressive and unified SuperTickets identification strategy that allows the connectivity of subnetworks to change during supernet training, achieving better accuracy and efficiency trade-offs than conventional sparse training. Finally, we evaluate whether such identified SuperTickets drawn from one task can transfer well to other tasks, validating their potential of simultaneously handling multiple tasks. Extensive experiments and ablation studies on three tasks and four benchmark datasets validate that our proposed SuperTickets achieve boosted accuracy and efficiency trade-offs than both typical NAS and pruning pipelines, regardless of having retraining or not. Codes and pretrained models are available at https://github.com/RICE-EIC/SuperTickets.

Keywords: Lottery Ticket Hypothesis, Efficient Training/Inference, Neural Architecture Search, Task-agnostic DNNs

1 Introduction

While deep neural networks (DNNs) have achieved unprecedented performance in various tasks and applications like classification, segmentation, and detection [9], their prohibitive training and inference costs limit their deployment on resource-constrained devices for more pervasive intelligence. For example, one

⋆ Work done while interning at Baidu USA

forward pass of the ResNet50 [18] requires 4 GFLOPs (FLOPs: floating point operations) and its training requires 10^{18} FLOPs [50]. To close the aforementioned gap, extensive attempts have been made to compress DNNs from either macro-architecture (e.g., NAS [38, 45, 9]) or fine-grained parameter (e.g., network pruning [17, 12]) levels. A commonly adopted DNN compression pipeline following a coarse-to-fine principle is to first automatically search efficient and powerful DNN architectures from a larger supernet and then prune the searched DNNs via costly train-prune-retrain process [11, 10, 23] to derive smaller and sparser subnetworks with a comparable or degraded accuracy but largely reduced inference costs. However, such pipeline requires a tedious search-train-prune-retrain process and thus still prohibitive training costs.

To address the above limitation for simplifying the pipeline and further improve the accuracy-efficiency trade-offs of the identified networks, we advocate a **two-in-one training** framework for simultaneously identifying both efficient DNNs and their lottery subnetworks via jointly architecture searching and parameter pruning. We term the identified small subnetworks as **SuperTickets** if they achieve comparable or even superior accuracy-efficiency trade-offs than previously adopted search-then-prune baselines, because they are drawn from supernets and represent both coarse-grained DNN architectures and fine-grained DNN subnetworks. We make non-trivial efforts to explore and validate the potential of SuperTickets by answering three key questions: *(1) whether such SuperTickets can be directly found from a supernet via two-in-one training? If yes, then (2) how to effectively identify such SuperTickets? and (3) can SuperTickets found from one task/dataset transfer to another, i.e., have the potential to handle different tasks/datasets?* To the best of our knowledge, this is the first attempt taken towards identifying both DNN architectures and their corresponding lottery ticket subnetworks through a unified two-in-one training scheme. Our contributions can be summarized as follows:

- We **for the first time** discover that efficient DNN architectures and their lottery subnetworks, i.e., SuperTickets, can be simultaneously identified from a supernet leading to superior accuracy-efficiency trade-offs.
- We develop an unified progressive identification strategy to effectively find the SuperTickets via a two-in-one training scheme which allows the subnetworks to iteratively reactivate the pruned connections during training, offering better performance than conventional sparse training. Notably, our identified SuperTickets *without retraining* already outperform previously adopted first-search-then-prune baselines, and thus can be directly deployed.
- We validate the transferability of identified SuperTickets across different tasks/datasets, and conduct extensive experiments to compare the proposed SuperTickets with those from existing search-then-prune baselines, typical NAS techniques, and pruning works. Results on three tasks and four datasets demonstrate the consistently superior accuracy-efficiency trade-offs and the promising transferability for handling different tasks offered by SuperTickets.

2 Related Works

Neural Architecture Search (NAS). NAS has achieved an amazing success in automating the design of efficient DNN architectures and boosting accuracy-efficiency trade-offs [57, 39, 19]. To search for task-specific DNNs, early works [39, 38, 19] adopt reinforcement learning based methods that require a prohibitive search time and computing resources, while recent works [26, 45, 41, 48] update both the weights and architectures during supernet training via differentiable search that can greatly improve the search efficiency as compared to prior NAS works. More recently, some works adopt one-shot NAS [16, 3, 53, 42] to decouple the architecture search from supernet training and then evaluate the performance of subnets (i.e., searched DNNs) whose weights are directly inherited from the pretrained supernet. Such methods are generally applicable to search for efficient CNNs [16, 2] or Transformers [43, 4, 37] for solving both vision and language tasks. To search for multi-task DNNs, recently emerging works like HR-NAS [9] and FBNetv5 [46] advocate supernet designs with multi-resolution branches so as to accommodate both image classification and other dense prediction tasks that require high-resolution representations. In this work, we propose to directly search for not only efficient DNNs but also their lottery subnetworks from supernets to achieve better accuracy-efficiency trade-offs while being able to handle different tasks.

Lottery Ticket Hypothesis (LTH). Frankle et al. [12, 13] showed that winning tickets (i.e., small subnetworks) exist in randomly initialized dense networks, which can be retrained to restore a comparable or even better accuracy than their dense network counterparts. This finding has inspired lots of research directions as it implies the potential of sparse subnetworks. For efficient training, You et al. [50] consistently find winning tickets at early training stages, largely reducing DNNs' training costs. Such finding has been extended to language models (e.g., BERT) [6], generative models (e.g., GAN) [33], and graph neural networks [51]; Zhang et al. [55] recognize winning tickets more efficiently by training with only a specially selected subset of data; and Ramanujan et al. [34] further identify winning tickets directly from random initialization that perform well even without retraining. For robust and efficient DNNs, Fu et al. [14] discover subnetworks with inborn robustness from random initializated DNNs, called robust scratch tickets; Chen et al. [5] leverage sparse training to reduce robust generalization and over-fitting. In contrast, our goal is to simultaneously find both efficient DNNs and their lottery subnetworks from supernets, beyond the scope of sparse training or drawing winning tickets from dense DNN models.

Task-Agnostic DNNs Design. To facilitate designing DNNs for different tasks, recent works [27, 19, 44] propose to design general architecture backbones for various computer vision tasks. For example, HR-Net [44] maintains high-resolution representations through the whole network for supporting dense prediction tasks, instead of connecting high-to-low resolution convolutions in series like ResNet or VGGNet; Swin-Transformer [27] adopts hierarchical vision transformers to serve as a general-purpose backbone that is compatible with a broad range of vision tasks; ViLBERT [29, 30] proposes a multi-modal two-

stream model to learn task-agnostic joint representations of both image and language; Data2vec [1] designs a general framework for self-supervised learning in speech, vision and language. Moreover, recent works [9, 46, 47] also leverage NAS to automatically search for task-agnostic and efficient DNNs from hand-crafted supernets. In this work, we aim to identify task-agnostic SuperTickets that achieve better accuracy-efficiency trade-offs.

3 The Proposed SuperTickets Method

In this section, we address the three key questions of SuperTickets. First, we develop a two-in-one training scheme to validate our hypothesis that SuperTickets exist and can be found directly from a supernet. Second, we further explore more effective SuperTickets identification strategies via iterative neuron reactivation and progressive pruning, largely boosting the accuracy-efficiency trade-offs. Third, we evaluate the transferability of the identified SuperTickets across different datasets or tasks, validating their potential of being task-agnostic.

3.1 Do SuperTickets Exist in Supernets?

SuperTickets Hypothesis. We hypothesize that both efficient DNN architectures and their lottery subnetworks can be directly identified from a supernet, and term these subnetworks as **SuperTickets** if they achieve on par or even better accuracy-efficiency trade-offs than those from first-search-then-prune counterparts. Considering a supernet $f(x;\theta_S)$, various DNN architectures a are sampled from it whose weights are represented by $\theta_S(a)$, then we can define SuperTickets as $f(x; m \odot \theta_S(a))$, where $m \in \{0,1\}$ is a mask to indicate the pruned and unpruned connections in searched DNNs. The SuperTickets Hypothesis implies that jointly optimizing DNN architectures a and corresponding sparse masks m works better, i.e., resulting in superior accuracy-efficiency trade-offs, than sequentially optimizing them.

Experiment Settings. To perform experiments for exploring whether SuperTickets generally exist, we need (1) a suitable supernet taking both classical efficient building blocks and task-agnostic DNN design principles into consideration and (2) corresponding tasks, datasets, and metrics. We elaborate our settings below. NAS and Supernets: We consider a multi-branch search space containing both efficient convolution and attention building blocks following one state-of-the-art (SOTA) work of HR-NAS [9], whose unique hierarchical multi-resolution search space for handling multiple vision tasks stands out compared to others. In general, it contains two paths: MixConv [40] and lightweight Transformer for extracting both local and global context information. Both the number of convolutional channels with various kernel sizes and the number of tokens in the Transformer are searchable parameters. Tasks, Datasets, and Metrics: We consider semantic segmentation on Cityscapes [7] and human pose estimation on COCO keypoint [24] as two representative tasks for illustrative purposes. For Cityscapes, we conduct experiments with 512×1024 input size, an initial learning rate of 0.04, a batch size of 32, and 430 training epochs. The mean Intersection

Algorithm 1: Two-in-One Framework for Identifying SuperTickets.

Input: The supernet weights θ_S, drop threshold ϵ, and pruning ratio p;
Output: Efficient DNNs and their lottery subnetworks $f(x; m \odot \theta_S(a))$.

while t *(epoch)* $< t_{max}$ **do**
  $t = t + 1$;
  Update weights θ_S and importance factor r using SGD training;
  if $t \mod t_s = 0$ **then** ▷ `Search for DNNs`
    Remove search units whose importance factors $r < \epsilon$;
    Recalibrate the running statistics of BN layers to obtain subnet a;
    // If enabling the iterative reactivation technique
    Reactivate the gradients of pruned weights;
  else if $t \mod t_p = 0$ **then** ▷ `Prune for subnetworks`
    // If enabling the progressive pruning technique
    Redefine the pruning ratio as $\min\{p, 10\% \times \lfloor t/t_p \rfloor\}$;
    Perform magnitude-based pruning towards the target ratio;
    Keep the sparse mask m_t and disable pruned weights' gradients;
  end
end
return $f(x; m_t \odot \theta_S(a))$; ▷ SuperTickets

over Union (mIoU), mean Accuracy (mAcc), and overall Accuracy (aAcc) are evaluation metrics. For COCO keypoint, we train the model using input size 256×192, an initial learning rate of 1e-3, a batch size of 384 for 210 epochs. The average precision (AP), recall scores (AR), AP^M and AP^L for medium or large objects are evaluation metrics. All experiments are run on Tesla V100*8 GPUs.

Two-in-One Training. To validate the SuperTickets hypothesis, we propose a two-in-one training algorithm that simultaneously searches and prunes during supernet training of NAS. As shown in Alg. 1 and Fig. 1, for searching for efficient DNNs, we adopt a progressive shrinking NAS by gradually removing unimportant search units that can be either convolutional channels or Transformer tokens. After every p_s training epochs, we will detect and remove the unimportant search units once their corresponding importance factors r (i.e., the scales in Batch Normalization (BN) layers) are less than a predefined drop threshold ϵ. Note that r can be jointly learned with supernet weights, such removing will not affect the remaining search units since channels in depth-wise convolutions are independent among each other, as also validated by [9, 32]. In addition, we follow network slimming [28] to add a l_1 penalty as a regularization term for polariz-

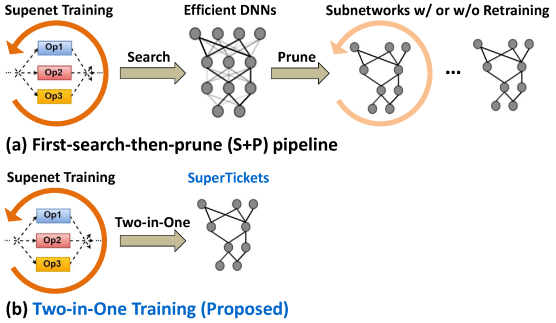


Fig. 1. Illustrating first-search-then-prune (S+P) vs. our two-in-One training.

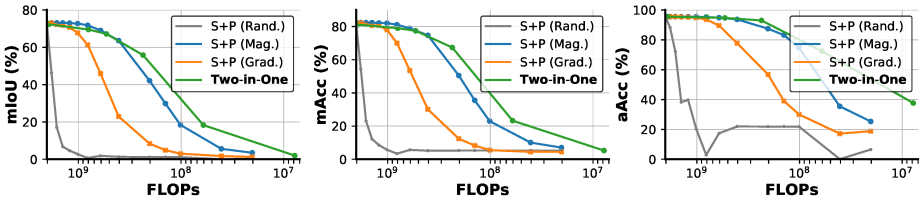


Fig. 2. Comparing the mIoU, mAcc, aAcc and inference FLOPs of the resulting networks from the proposed two-in-one training and first-search-then-prune (S+P) baselines on semantic segmentation task and Cityscapes dataset, where Rand., Mag., and Grad. represent random, magnitude, and graident-based pruning, respectively. Note that each method has a series of points for representing different pruning ratios ranging from 10% to 98%. All accuracies are averaged over three runs.

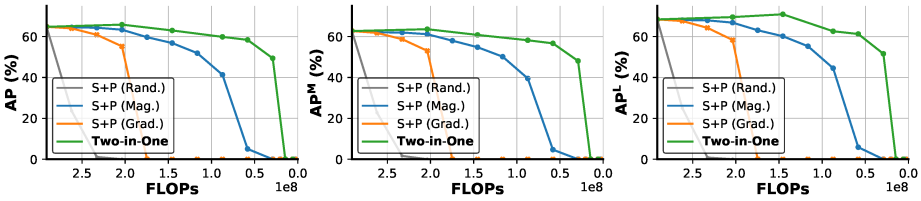


Fig. 3. Comparing the AP, AP^M, AP^L and inference FLOPs of the resulting networks from the proposed two-in-one training and baselines on human pose estimation task and COCO keypoint dataset. Each method has a series of points for representing different pruning ratios ranging from 10% to 98%. All accuracies are averaged over three runs.

ing the importance factors to ease the detection of unimportant units. After removing them, the running statistics in BN layers are recalibrated in order to match the searched DNN architecture a for avoiding covariate shift [21, 49]. For pruning of searched DNNs, we perform magnitude-based pruning towards the given pruning ratio per t_p epochs, the generated spare mask m_t will be kept so as to disable the gradients flow of the pruned weights during the following training. Note that we do not incorporate the iterative reactivation and progressive pruning techniques (highlighted with colors/shadows in Alg. 1, which will be elaborated later) as for now. Such vanilla two-in-one training algorithm can be regarded as the first step towards answering the puzzle whether SuperTickets generally exist.

Existence of SuperTickets. We compare the proposed two-in-one training with first-search-then-prune (S+P) baselines and report the results on Cityscapes and COCO keypoint at Fig. 2 and Fig. 3, respectively. We see that the proposed two-in-one training consistently generates comparable or even better accuracy-efficiency trade-offs as compared to S+P with various pruning criteria (random, magnitude, and gradient) since our methods demonstrate much better performance of segmentation or human pose estimation under different FLOPs reductions as shown in the above two figures, indicating that SuperTickets generally exist in a supernet and have great potential to outperform the commonly adopted approaches, i.e., sequentially optimizing DNN architectures and sparse masks.

Table 1. Breakdown analysis of the proposed SuperTickets identification strategy. We report the performance of found subnetworks under 90%/80% sparsity on two datasets.

Methods	2-in-1	PP	IR-P	IR-S	Retrain	Cityscapes			COCO Keypoint			
						mIoU	mAcc	aAcc	AP	AP^M	AP^L	AR
S+P (Mag.)						42.12	50.49	87.45	5.04	4.69	5.89	10.67
S+P (Mag.)					✓	51.03	59.61	90.88	48.63	46.82	51.74	53.38
Ours	✓					55.84	67.38	92.97	58.38	56.68	61.26	62.23
Ours	✓	✓				63.89	73.56	94.17	60.14	57.93	63.70	63.79
Ours	✓	✓	✓			45.73	55.52	89.36	5.48	7.43	4.36	10.85
Ours	✓	✓		✓		**66.61**	**76.30**	**94.63**	**61.02**	**58.80**	**64.64**	**64.78**
Ours	✓	✓		✓	✓	**67.17**	**77.03**	**94.73**	**61.48**	**59.30**	**65.19**	**65.20**

3.2 How to More Effectively Identify SuperTickets?

We have validated the existence of SuperTickets, the natural next question is how to more effectively identify them. To this end, we propose two techniques that can be seamlessly incorporated into the two-in-one training framework to more effectively identify SuperTickets and further boost their achievable performance.

Progressive Pruning (PP). Although simultaneously searching and pruning during supernet training enables the opportunity of cooperation between coarse-grained search units removal and fine-grained weights pruning, i.e., NAS helps to refine the pruned networks as a compensation by removing over-pruned units for avoiding bottlenecked layers, we find that over-pruning at the early training stages inevitably hurts the networks' generalizability, and further propose a progressive pruning (PP) techniques to overcome this shortcoming. As highlighted in the **cyan** part of Alg. 1, the pruning ratio is defined as $\min\{p, 10\% \times \lfloor t/t_p \rfloor\}$, which means that the network sparsity will gradually increase from 10% to the target ratio p, by 10% per t_p epochs. The PP technique helps to effectively avoid over-pruning at early training stages and thus largely boosts the final performance. As demonstrated in Table 1, two-in-one training with PP achieves 8.05%/6.18%/1.2% mIoU/mAcc/aAcc and 1.76%/1.25%/2.44%/1.56% AP/AP^M/AP^L/AR improvements on Cityscapes and COCO keypoint datasets, respectively, as compared to the vanilla two-in-one training under 90% sparsity.

Iterative Reactivation (IR). Another problem in the two-in-one framework is that the pruned weights will never get gradients updates throughout the remaining training. To further boost the performance, we design an iterative reactivation (IR) strategy to facilitate the effective SuperTickets identification by allowing the connectivity of subnetworks to change during supernet training. Specifically, we reactivate the gradients of pruned weights as highlighted in the **orange** part of Alg. 1. Note that we reactivate during searching instead of right after pruning, based on a hypothesis that sparse training is also essential to the two-in-one training framework. In practice, the pruning interval p_t is different from the searching interval p_s in order to allow a period of sparse training. To validate the hypothesis, we design two variants: IR-S and IR-P that reactivate pruned weights' gradients during searching and pruning, respectively, and show the comparisons in Table 1. We observe that: (1) IR-P leads to even worse accuracy than vanilla two-in-one training, validating that sparse training is essential; (2) IR-S further leads to 2.72%/2.74%/0.46% mIoU/mAcc/aAcc and

0.88%/0.87%/0.94%/0.99% AP/AP^M/ AP^L/AR improvements on Cityscapes and COCO keypoint, respectively, on top of two-in-one training with PP.

SuperTickets w/ or w/o Retraining. Since the supernet training, architecture search, and weight pruning are conducted in an unified end-to-end manner, the resulting SuperTickets can be deployed directly without retraining, achieving better accuracy-efficiency trade-offs than S+P baselines (even with retraining) as indicated by Table 1. To investigate whether retraining can further boost the performance, we retrain the found SuperTickets for another 50 epochs and report the results at Table 1. We see that retraining further leads to 0.56%/0.73%/0.10% mIoU/mAcc/aAcc and 0.46%/0.50%/0.55%/0.42% AP/AP^M/ AP^L/AR improvements on Cityscapes and COCO keypoint datasets, respectively.

3.3 Can the Identified SuperTickets Transfer?

To validate the potential of identified SuperTickets for handling different tasks and datasets, we provide empirical experiments and analysis as follows. Note that we adjust the final classifier to match target datasets during transfer learning.

SuperTickets Transferring Among Datasets. We first test the transferability of the identified SuperTickets among different datasets within the same task, i.e., Cityscapes and ADE20K as two representatives in the semantic segmentation task. Table 2 shows that SuperTickets identified from one dataset can transfer to another dataset while leading to comparable or even better performance than S+P baselines with (denoted as "w/ RT") or without retraining (by default). For example, when tested on Cityscapes, SuperTickets identified from ADE20K after fine-tuning lead to 2.2% and 20.8% higher mIoU than S+P (Mag.) w/ and w/o RT baselines which are directly trained on target Cityscapes dataset. Likewise, the SuperTickets transferred from Cityscapes to ADE20K also outperform baselines on target dataset.

Table 2. Supertickets transfer validation tests under 90% sparsity.

Methods	Params	FLOPs	Cityscapes		
			mIoU	mAcc	aAcc
S+P (Grad.)	0.13M	203M	8.41	12.39	56.77
S+P (Mag.)	0.13M	203M	42.12	50.49	87.45
S+P (Mag.) w/ RT	0.13M	203M	60.76	70.40	93.38
ADE20K Tickets	0.20M	247M	**62.91**	**73.32**	**93.82**
ImageNet Tickets	0.18M	294M	61.64	71.78	93.75
Methods	**Params**	**FLOPs**	**ADE20K**		
			mIoU	mAcc	aAcc
S+P (Grad.)	0.11M	154M	0.79	1.50	25.58
S+P (Mag.)	0.11M	154M	3.37	4.70	39.47
Cityscapes Tickets	0.13M	119M	20.83	29.95	69.00
ImageNet Tickets	0.21M	189M	**22.42**	**31.87**	**70.21**

SuperTickets Transferring Among Tasks. To further investigate whether the identified SuperTickets can transfer among different tasks. We consider to transfer SuperTickets's feature extraction modules identified from ImageNet on classification task to Cityscapes and ADE20K on segmentation tasks, where the dense prediction heads and final classifier are still inherited from the target datasets. The results are presented in the last row of the two sub-tables in Table 2. We observe that such transferred networks still perform well on downstream tasks. Sometimes, it even achieves better performance than transferring within one task, e.g., ImageNet $\rightarrow$ ADE20K works better (1.6% higher mIoU) than Cityscapes $\rightarrow$ ADE20K. We supply more experiments on various pruning ratios in Sec. 4.3.2.

4 Experiment Results

In this section, we first describe our experiment settings, and then benchmark the identified SuperTickets over both typical NAS or pruning methods' resulting DNNs and first-search-then-prune baselines on three commonly used vision tasks(classification, semantic segmentation, human pose key point detection) and four datasets. After that, we conduct ablation studies regarding the proposed SuperTickets' identifier and transferability.

4.1 Experiment Setting

Tasks, Datasets, and Supernets. Tasks and Datasets. We consider four benchmark datasets and three representative vision tasks to demonstrate the effectiveness of SuperTickets, including image classification on ImageNet [8] dataset with 1.2 million training images and 50K validation images; semantic segmentation on Cityscapes [7] and ADE20K [56] datasets with 2975/500/1525 and 20K/2K/3K images for training, validation, and testing, respectively; human pose estimation on COCO keypoint [24] dataset with 57K images and 150K person instances for training, and 5K images for validation. These selected datasets require different receptive fields and global/local contexts, manifesting themselves as proper test-beds for SuperTickets on multiple tasks. Supernets. For all experiments, we adopt the same supernet as HR-NAS [9] thanks to the task-agnostic multi-resolution supernet design. It begins with two 3×3 convolutions with stride 2, which is followed by five parallel modules to gradually divide it into four branches of decreasing resolutions, the learned features from all branches are then merged together for classification or dense prediction.

Search and Training Settings. For training supernets on ImageNet, we adopt a RMSProp optimizer with 0.9 momentum and 1e-5 weight decay, exponential moving average (EMA) with 0.9999 decay, and exponential learning rate decay with an initial learning rate of 0.016 and 256 batch size for 350 epochs. For Cityscapes and ADE20K, we use an AdamW optimizer, an initial learning rate of 0.04 with batch size 32 due to larger input image sizes, and train for 430 and 200 epochs, respectively, following [9]. For COCO keypoint, we follow [44] to use an Adam optimizer for 210 epochs, the initial learning rate is set to 1e-3, and is divided by 10 at the 170th and 200th epochs, respectively. In addition, we perform architecture search during supernet training. For all search units, we use the scales from their attached BN layers as importance factors r; search units with $r < 0.001$ are regarded as unimportant and removed every 10 epochs (i.e., $t_s = 10$); Correspondingly, magnitude-based pruning will be performed per 25 epochs for ImageNet and Cityscapes, or per 15 epochs for ADE20K and COCO keypoint (i.e., $t_p = 25/15$), resulting intervals for sparse training as in Sec. 3.2.

Baselines and Evaluation Metrics. Baselines. For all experiments, we consider the S+P pipeline as one of our baselines, where the search method follows [9]; the pruning methods can be chosen from random pruning, magnitude pruning [17, 12], and gradient pruning [22]. In addition, we also benchmark with hand-crafted DNNs, e.g., ShuffleNet [54, 31] and MobiletNetV2 [35], and prior

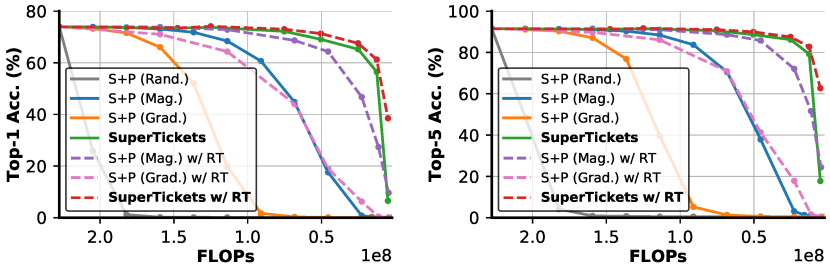


Fig. 4. Comparing the top-1/5 accuracy and FLOPs of the proposed SuperTickets and S+P baselines on ImageNet. Each method has a series of points to represent different pruning ratios ranging from 10% to 98%. All accuracies are averaged over three runs. We also benchmark all methods with retraining (denoted as w/ RT).

typical NAS resulting task-specific DNNs, e.g., MobileNetV3 [19] and Auto-DeepLab [25]. We do not compare with NAS/tickets works with SOTA accuracy due to different goals and experimental settings. All baselines are benchmarked under similar FLOPs or accuracy for fair comparisons. Evaluation Metrics. We evaluate the SuperTickets and all baselines in terms of accuracy-efficiency trade-offs. Specifically, the accuracy metrics refer to top-1/5 accuracy for classification tasks; mIoU, mAcc, and aAcc for segmentation tasks; AP, AR, AP^M, and AP^L for human pose estimation tasks. For efficiency metrics, we evaluate and compare both the number of parameters and inference FLOPs.

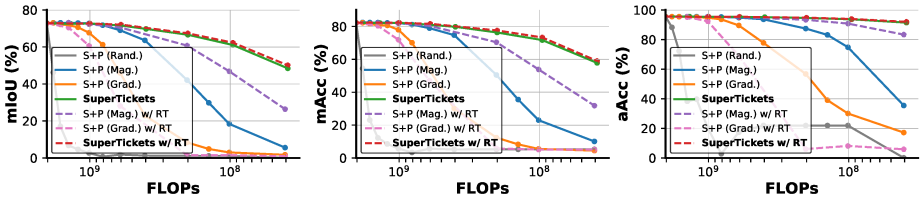
4.2 Evaluating SuperTickets over Typical Baselines

4.2.1 SuperTickets on the Classification Task

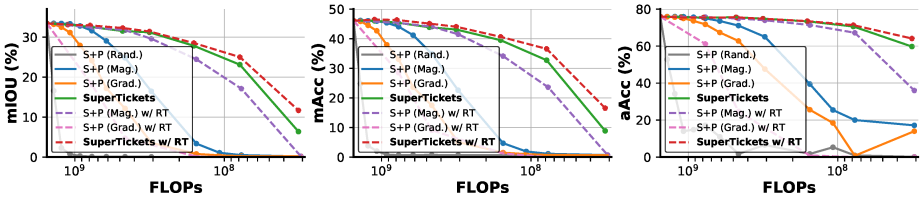
We show the overall comparisons between SuperTickets and some typical baselines in terms of accuracy-efficiency trade-offs in Fig. 4 and Table. 3, from which we have **two observations**. First, SuperTickets consistently outperform all baselines by reducing the inference FLOPs while achieving a comparable or even better accuracy. Specifically, SuperTickets reduce 61.4% $\sim$ 81.5% FLOPs while offering a comparable or better accuracy (+0.1% $\sim$ +4.6%) as compared to both S+P and some task-specific DNNs; Likewise, when comparing under comparable number of parameters or FLOPs, SuperTickets lead to on average 26.5% (up to 64.5%) and on average 41.3% (up to 71.9%) top-1 accuracy improvements as compared to S+P (Mag.) and S+P (Grad.) across various pruning ratios, e.g., under 50% pruning ratios, SuperTickets achieve 74.2% top-1 accuracy, +1.4% and +9.9% over S+P (Mag.) and S+P (Grad.), respectively. Second, SuperTickets w/o retraining even surpass S+P baselines with retraining as demonstrated in Fig. 4, leading to on average 6.7%

Table 3. SuperTickets vs. some typical methods on ImageNet. FLOPs is measured with the input size of 224×224.

Model	Params	FLOPs	Top-1 Acc.
CondenseNet [20]	2.9M	274M	71.0%
ShuffleNetV1 [54]	3.4M	292M	71.5%
ShuffleNetV2 [31]	3.5M	299M	72.6%
MobileNetV2 [35]	3.4M	300M	72.0%
FBNet [45]	4.5M	295M	74.1%
S+P (Grad.)	2.7M	114M	64.3%
S+P (Mag.)	2.7M	114M	72.8%
SuperTickets	2.7M	125M	**74.2%**



(a) Comparing SuperTickets with S+P baselines on Cityscapes.



(b) Comparing SuperTickets with S+P baselines on ADE20K.

Fig. 5. Comparing the mIoU, mAcc, aAcc and inference FLOPs of the proposed SuperTickets and S+P baselines on Cityscapes and ADE20K datasets. Each method has a series of points to represent different pruning ratios ranging from 10% to 98%.

(up to 29.2%) higher top-1 accuracy under comparable FLOPs across various pruning ratios (10% $\sim$ 98%). Furthermore, SuperTickets w/ retraining achieve 0.1% $\sim$ 31.9% (on average 5.3%) higher accuracy than the counterparts w/o retraining, pushing forward the frontier of accuracy-efficiency trade-offs.

4.2.2 SuperTickets on the Segmentation Task

Experiments on Cityscapes. We compare SuperTickets with typical baselines on Cityscapes as shown in Fig. 5 (a) and Table 4. We see that SuperTickets consistently outperform all baselines in terms of mIoU/mAcc/aAcc and FLOPs. Specifically, SuperTickets reduce 60% $\sim$ 80.86% FLOPs while offering a comparable or better mIoU (0.28 % $\sim$ 43.26%) as compared to both S+P and task-specific DNNs; Likewise, when comparing under comparable number of parameters or FLOPs, SuperTickets lead to on average 17.70% (up to 42.86%) and 33.36% (up to 58.05%) mIoU improvements as compared to S+P (Mag.) and S+P (Grad.) across various pruning ratios, e.g., under 50% pruning ratios, SuperTickets achieve 72.68% mIoU, +0.37% and +12% over S+P (Mag.) and S+P (Grad.), respectively. We also report the comparison among methods after retraining at Fig. 5, as denoted by "w/ RT". We find that S+P (Grad.) w/ RT suffers from overfitting and even leads to worse performance; In contrast, SuperTickets w/ retraining

Table 4. SuperTickets vs. some typical methods on Cityscapes. FLOPs is measured with the input size of 512×1024.

Model	Params	FLOPs	mIoU
BiSeNet [52]	5.8M	6.6G	69.00%
MobileNetV2 [35]	2.1M	5.3G	70.71%
MobileNetV3 [19]	1.5M	2.5G	72.36%
ShuffleNetV2 [31]	3.0M	6.9G	71.30%
Auto-DeepLab [25]	3.2M	27.3G	71.21%
SqueezeNAS [36]	0.73M	8.4G	72.40%
S+P (Grad.) w/ RT	0.63M	1.0G	60.66%
S+P (Mag.) w/ RT	0.63M	1.0G	72.31%
SuperTickets	0.63M	1.0G	**72.68%**

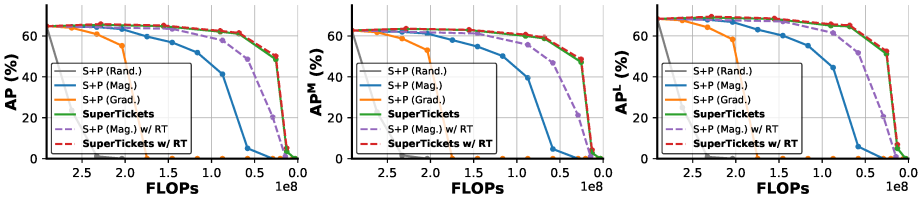


Fig. 6. Comparing the AP, AP^M, AP^L and inference FLOPs of the proposed SuperTickets and baselines on human pose estimation task and COCO keypoint dataset. Each method has a series of points for representing different pruning ratios ranging from 10% to 98%. All accuracies are averaged over three runs.

further achieve 0.51% $\sim$ 1.64% higher accuracy than the counterparts w/o retraining, pushing forward the frontier of accuracy-efficiency trade-offs.

Experiments on ADE20K. Similarly, we test the superiority of SuperTickets on ADE20K as shown in Fig. 5 (b) and Table 5. The proposed SuperTickets consistently outperform all baselines in terms of accuracy-efficiency trade-offs, reducing 38.46% $\sim$ 48.53% FLOPs when comparing under similar mIoU. When compared under comparable number of parameters or FLOPs, SuperTickets lead to an average of 9.43% (up to 22.6%) and 14.17% (up to 27.61%) mIoU improvements as compared to S+P (Mag.) and S+P (Grad.), respectively, across various pruning ratios. In addition, SuperTickets w/ retraining further achieve 0.01% $\sim$ 5.3% higher accuracy than the counterparts w/o retraining on ADE20K.

Table 5. SuperTickets vs. typical methods on ADE20K. FLOPs is measured with the input size of 512×512.

Model	Params	FLOPs	mIoU
MobileNetV2 [35]	2.2M	2.8G	32.04%
MobileNetV3 [19]	1.6M	1.3G	32.31%
S+P (Grad.)	1.0M	0.8G	24.14%
S+P (Mag.)	1.0M	0.8G	31.59%
SuperTickets	1.0M	0.8G	**32.54%**

4.2.3 SuperTickets on the Human Pose Estimation Task

We compare SuperTickets with a few typical baselines on COCO keypoint as shown in Fig. 6 and Table 6. We see that SuperTickets consistently outperform all the related baselines in terms of AP/AP^M/AP^L/AR and FLOPs. Specifically, SuperTickets reduce 30.3% $\sim$ 78.1% FLOPs while offering a comparable or better AP (+0.8% $\sim$ 11.79%) as compared to both S+P and task-specific DNNs; Likewise, when comparing under comparable number of parameters or FLOPs, SuperTickets lead to on average 17.4% (up to 55.9%) AP improvements. In addition, SuperTickets w/ retraining further achieve on average 1.1% higher accuracy than the counterparts w/o retraining on COCO keypoint.

Table 6. SuperTickets vs. typical algorithms on COCO. FLOPs is measured with the input size of 256×192.

Model	Params	FLOPs	AP	AP^M	AP^L	AR
ShuffleNetV1 [54]	1.0M	0.16G	58.5	55.2	64.6	65.1
ShuffleNetV2 [31]	1.3M	0.17G	59.8	56.5	66.2	66.4
MobileNetV2 [35]	2.3M	0.33G	64.6	61.0	71.1	70.7
S+P (Mag.)	0.6M	0.23G	63.4	61.2	66.8	67.3
SuperTickets	0.6M	0.23G	**65.4**	**63.4**	69.0	68.9

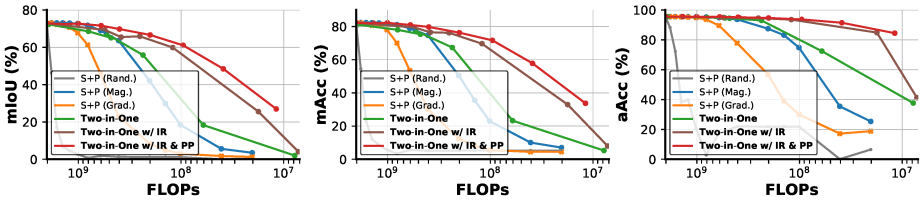


Fig. 7. Ablation studies of the SuperTickets identified from two-in-one framework w/ or w/o the proposed iterative activation (IR) and progressive pruning (PP) techniques.

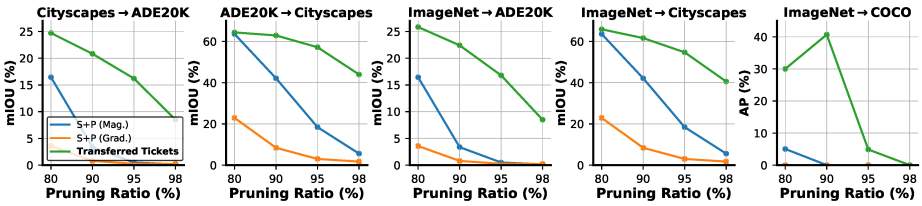


Fig. 8. Ablation studies of transferring identified SuperTickets from one dataset/task to another dataset/task under various pruning ratios ranging from 80% to 98%.

4.3 Ablation Studies of the Proposed SuperTickets

4.3.1 Ablation Studies of SuperTickets' Identification

We provide comprehensive ablation studies to show the benefit breakdown of the proposed two-in-one training framework and more effective identification techniques, i.e., progressive pruning (PP) and iterative reactivation (IR). As shown in Fig. 7, we report the complete mIoU-FLOPs trade-offs with various pruning ratios ranging from 10% to 99% when testing on Cityscapes dataset, where x axis is represented by log-scale for emphasizing the improvements when pruning ratio reaches high. As compared to S+P (Mag.), SuperTickets identified from vanilla two-in-one framework achieve up to 40.17% FLOPs reductions when comparing under similar mIoU, or up to 13.72% accuracy improvements when comparing under similar FLOPs; Adopting IR during two-in-one training further leads to up to 68.32% FLOPs reductions or up to 39.12% mIoU improvements; On top of the above, adopting both IR and PP during two-in-one training offers up to 80.86% FLOPs reductions or up to 43.26% mIoU improvements. This set of experiments validate the effectiveness of the general two-in-one framework and each of the proposed techniques.

4.3.2 Ablation Studies of SuperTickets' Transferability

We previously use one set of experiments under 90% sparsity in Sec. 3.3 to validate that the identified SuperTickets can transfer well. In this section, we supply more comprehensive ablation experiments under various pruning ratios and among several datasets/tasks. As shown in Fig. 8, the left two subplots indicate the transfer between different datasets (Cityscapes $\leftrightarrow$ ADE20K) generally works across four pruning ratios. In particular, transferred SuperTickets lead to

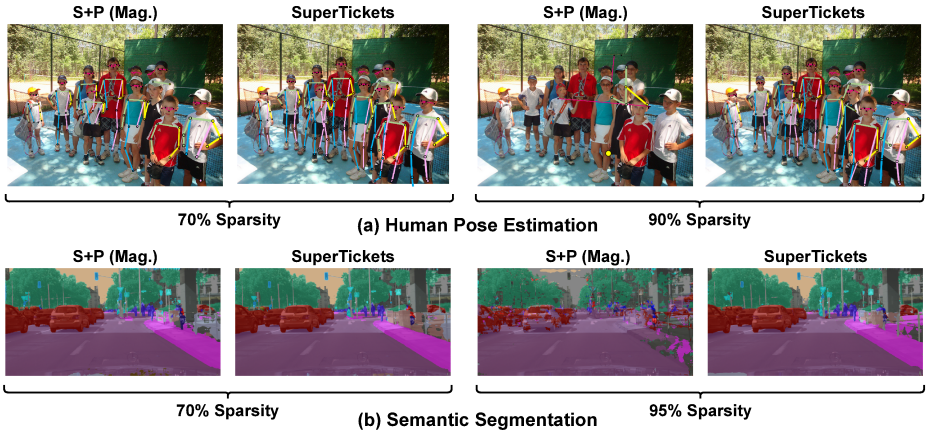


Fig. 9. Visualization of the human pose estimation on COCO keypoint dataset and the streetview/semantic labels on Cityscapes dataset under different pruning ratios.

76.14% $\sim$ 81.35% FLOPs reductions as compared to the most competitive S+P baseline, while offering comparable mIoU (0.27% $\sim$ 1.85%). Furthermore, the right three subplots validate that the identified SuperTickets from classification task can transfer well to other tasks (i.e., segmentation and human pose estimation). Specifically, it leads to 68.67% $\sim$ 69.43% FLOPs reductions as compared to the S+P (Mag.) baseline, when achieving comparable mIoU or AP.

5 Discussions

Limitations of Transferred SuperTickets. Although identified SuperTickets can transfer with only classifiers as task-specific, there is still a limitation in the transferred SuperTickets. That is, transferred SuperTickets cannot surpass those SuperTickets directly found on the target datasets/tasks. Moreover, when the sparsity is low (e.g., 30%), the transferred SuperTickets will underperform both SuperTickets and S+P. This is counterintuitive and opposite to the observation in compressing pretrained models [15], where low pruning ratios do not hurt the accuracy after transferring while overpruning leads to under-fitting. It implies that the dedicated search is necessary when pruning ratio is relatively low; while for high sparsity, the impacts of neural architectures will be less.

Visualization and Discussion. We visualize the results of SuperTickets and S+P baselines on COCO keypoint and Cityscapes datasets under different pruning ratios, as shown in Fig. 9. We observe that S+P baselines work but miss some keypoints or semantic understandings under medium sparsity (e.g., 70%) while collapse under high pruning ratios (e.g, 90/95%); In contrast, our identified SuperTickets consistently work well among a wide range of pruning ratios, validating the effectiveness of our proposed SuperTickets.

6 Conclusion

In this paper, we advocate a two-in-one framework where both efficient DNN architectures and their lottery subnetworks (i.e., SuperTickets) can be identified from a supernet simultaneously, resulting in better performance than first-search-then-prune baselines. Also, we develop two techniques during supernet training to more effectively identify such SuperTickets, pushing forward the frontier of accuracy-efficiency trade-offs. Moreover, we test the transferability of SuperTickets to reveal their potential for being task-agnostic. Results on three tasks and four datasets consistently demonstrate the superiority of our proposed two-in-one framework and the resulting SuperTickets, opening up a new perspective in searching and pruning for more accurate and efficient networks.

References

1. Baevski, A., Hsu, W.N., Xu, Q., Babu, A., Gu, J., Auli, M.: data2vec: A general framework for self-supervised learning in speech, vision and language. arXiv preprint arXiv:2202.03555 (2022)
2. Bender, G., Kindermans, P.J., Zoph, B., Vasudevan, V., Le, Q.: Understanding and simplifying one-shot architecture search. In: International Conference on Machine Learning. pp. 550–559. PMLR (2018)
3. Cai, H., Gan, C., Wang, T., Zhang, Z., Han, S.: Once-for-all: Train one network and specialize it for efficient deployment. arXiv preprint arXiv:1908.09791 (2019)
4. Chen, M., Peng, H., Fu, J., Ling, H.: Autoformer: Searching transformers for visual recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
5. Chen, T., Zhang, Z., pengjun wang, Balachandra, S., Ma, H., Wang, Z., Wang, Z.: Sparsity winning twice: Better robust generalization from more efficient training. In: International Conference on Learning Representations (2022), https://openreview.net/forum?id=SYuJXrXq8tw
6. Chen, X., Cheng, Y., Wang, S., Gan, Z., Wang, Z., Liu, J.: Earlybert: Efficient bert training via early-bird lottery tickets. arXiv preprint arXiv:2101.00063 (2020)
7. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3213–3223 (2016)
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
9. Ding, M., Lian, X., Yang, L., Wang, P., Jin, X., Lu, Z., Luo, P.: Hr-nas: Searching efficient high-resolution neural architectures with lightweight transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2982–2992 (2021)
10. Ding, Y., Wu, Y., Huang, C., Tang, S., Wu, F., Yang, Y., Zhu, W., Zhuang, Y.: Nap: Neural architecture search with pruning. Neurocomputing (2022)
11. Feng, Q., Xu, K., Li, Y., Sun, Y., Wang, D.: Edge-wise one-level global pruning on nas generated networks. In: Chinese Conference on Pattern Recognition and Computer Vision (PRCV). pp. 3–15. Springer (2021)
12. Frankle, J., Carbin, M.: The lottery ticket hypothesis: Finding sparse, trainable neural networks. In: International Conference on Learning Representations (2019), https://openreview.net/forum?id=rJl-b3RcF7
13. Frankle, J., Dziugaite, G.K., Roy, D., Carbin, M.: Linear mode connectivity and the lottery ticket hypothesis. In: International Conference on Machine Learning. pp. 3259–3269. PMLR (2020)
14. Fu, Y., Yu, Q., Zhang, Y., Wu, S., Ouyang, X., Cox, D., Lin, Y.: Drawing robust scratch tickets: Subnetworks with inborn robustness are found within randomly initialized networks. Advances in Neural Information Processing Systems **34** (2021)
15. Gordon, M.A., Duh, K., Andrews, N.: Compressing bert: Studying the effects of weight pruning on transfer learning. arXiv preprint arXiv:2002.08307 (2020)
16. Guo, Z., Zhang, X., Mu, H., Heng, W., Liu, Z., Wei, Y., Sun, J.: Single path one-shot neural architecture search with uniform sampling. In: European Conference on Computer Vision. pp. 544–560. Springer (2020)

17. Han, S., Mao, H., Dally, W.J.: Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. arXiv preprint arXiv:1510.00149 (2015)
18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016), https://github.com/facebookarchive/fb.resnet.torch
19. Howard, A., Sandler, M., Chu, G., Chen, L.C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., et al.: Searching for mobilenetv3. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1314–1324 (2019)
20. Huang, G., Liu, S., Van der Maaten, L., Weinberger, K.Q.: Condensenet: An efficient densenet using learned group convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2752–2761 (2018)
21. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. pp. 448–456. PMLR (2015)
22. Lee, N., Ajanthan, T., Torr, P.H.: Snip: Single-shot network pruning based on connection sensitivity. arXiv preprint arXiv:1810.02340 (2018)
23. Li, Z., Yuan, G., Niu, W., Zhao, P., Li, Y., Cai, Y., Shen, X., Zhan, Z., Kong, Z., Jin, Q., et al.: Npas: A compiler-aware framework of unified network pruning and architecture search for beyond real-time mobile acceleration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14255–14266 (2021)
24. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
25. Liu, C., Chen, L.C., Schroff, F., Adam, H., Hua, W., Yuille, A.L., Fei-Fei, L.: Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 82–92 (2019)
26. Liu, H., Simonyan, K., Yang, Y.: Darts: Differentiable architecture search. arXiv preprint arXiv:1806.09055 (2018)
27. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10012–10022 (2021)
28. Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., Zhang, C.: Learning efficient convolutional networks through network slimming. In: Proceedings of the IEEE international conference on computer vision. pp. 2736–2744 (2017)
29. Lu, J., Batra, D., Parikh, D., Lee, S.: Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. Advances in neural information processing systems **32** (2019)
30. Lu, J., Goswami, V., Rohrbach, M., Parikh, D., Lee, S.: 12-in-1: Multi-task vision and language representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10437–10446 (2020)
31. Ma, N., Zhang, X., Zheng, H.T., Sun, J.: Shufflenet v2: Practical guidelines for efficient cnn architecture design. In: Proceedings of the European conference on computer vision (ECCV). pp. 116–131 (2018)
32. Mei, J., Li, Y., Lian, X., Jin, X., Yang, L., Yuille, A., Yang, J.: Atomnas: Fine-grained end-to-end neural architecture search. arXiv preprint arXiv:1912.09640 (2019)

33. Mukund Kalibhat, N., Balaji, Y., Feizi, S.: Winning lottery tickets in deep generative models. arXiv e-prints pp. arXiv–2010 (2020)
34. Ramanujan, V., Wortsman, M., Kembhavi, A., Farhadi, A., Rastegari, M.: What's hidden in a randomly weighted neural network? In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11893–11902 (2020)
35. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4510–4520 (2018)
36. Shaw, A., Hunter, D., Landola, F., Sidhu, S.: Squeezenas: Fast neural architecture search for faster semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. pp. 0–0 (2019)
37. Su, X., You, S., Xie, J., Zheng, M., Wang, F., Qian, C., Zhang, C., Wang, X., Xu, C.: Vision transformer architecture search. arXiv preprint arXiv:2106.13700 (2021)
38. Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., Le, Q.V.: Mnasnet: Platform-aware neural architecture search for mobile. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2820–2828 (2019)
39. Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning. pp. 6105–6114. PMLR (2019)
40. Tan, M., Le, Q.V.: Mixconv: Mixed depthwise convolutional kernels. arXiv preprint arXiv:1907.09595 (2019)
41. Wan, A., Dai, X., Zhang, P., He, Z., Tian, Y., Xie, S., Wu, B., Yu, M., Xu, T., Chen, K., et al.: Fbnetv2: Differentiable neural architecture search for spatial and channel dimensions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12965–12974 (2020)
42. Wang, D., Gong, C., Li, M., Liu, Q., Chandra, V.: Alphanet: Improved training of supernet with alpha-divergence. arXiv preprint arXiv:2102.07954 (2021)
43. Wang, H., Wu, Z., Liu, Z., Cai, H., Zhu, L., Gan, C., Han, S.: Hat: Hardware-aware transformers for efficient natural language processing. arXiv preprint arXiv:2005.14187 (2020)
44. Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., et al.: Deep high-resolution representation learning for visual recognition. IEEE transactions on pattern analysis and machine intelligence **43**(10), 3349–3364 (2020)
45. Wu, B., Dai, X., Zhang, P., Wang, Y., Sun, F., Wu, Y., Tian, Y., Vajda, P., Jia, Y., Keutzer, K.: Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10734–10742 (2019)
46. Wu, B., Li, C., Zhang, H., Dai, X., Zhang, P., Yu, M., Wang, J., Lin, Y., Vajda, P.: Fbnetv5: Neural architecture search for multiple tasks in one run. arXiv preprint arXiv:2111.10007 (2021)
47. Xu, J., Tan, X., Luo, R., Song, K., Li, J., Qin, T., Liu, T.Y.: Nas-bert: task-agnostic and adaptive-size bert compression with neural architecture search. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. pp. 1933–1943 (2021)
48. Yang, Y., You, S., Li, H., Wang, F., Qian, C., Lin, Z.: Towards improving the consistency, efficiency, and flexibility of differentiable neural architecture search. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6667–6676 (June 2021)

49. You, F., Li, J., Zhao, Z.: Test-time batch statistics calibration for covariate shift. arXiv preprint arXiv:2110.04065 (2021)
50. You, H., Li, C., Xu, P., Fu, Y., Wang, Y., Chen, X., Baraniuk, R.G., Wang, Z., Lin, Y.: Drawing early-bird tickets: Toward more efficient training of deep networks. In: International Conference on Learning Representations (2020), https://openreview.net/forum?id=BJxsrgStvr
51. You, H., Lu, Z., Zhou, Z., Fu, Y., Lin, Y.: Early-bird gcns: Graph-network co-optimization towards more efficient gcn training and inference via drawing early-bird lottery tickets. In: Association for the Advancement of Artificial Intelligence (2022)
52. Yu, C., Wang, J., Peng, C., Gao, C., Yu, G., Sang, N.: Bisenet: Bilateral segmentation network for real-time semantic segmentation. In: Proceedings of the European conference on computer vision (ECCV). pp. 325–341 (2018)
53. Yu, J., Jin, P., Liu, H., Bender, G., Kindermans, P.J., Tan, M., Huang, T., Song, X., Pang, R., Le, Q.: Bignas: Scaling up neural architecture search with big single-stage models. In: European Conference on Computer Vision. pp. 702–717. Springer (2020)
54. Zhang, X., Zhou, X., Lin, M., Sun, J.: Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 6848–6856 (2018)
55. Zhang, Z., Chen, X., Chen, T., Wang, Z.: Efficient lottery ticket finding: Less data is more. In: International Conference on Machine Learning. pp. 12380–12390. PMLR (2021)
56. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 633–641 (2017)
57. Zoph, B., Vasudevan, V., Shlens, J., Le, Q.V.: Learning transferable architectures for scalable image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8697–8710 (2018)

A Visualization of The Adopted Supernet Architecture

We visualize the adopted supernet following [9] in Fig. 10. It begins with two 3×3 convolutions with stride 2, which are followed by five fusion modules and five parallel modules to gradually divide it into four branches of decreasing resolutions, the learned features from all branches are then merged together for classification or dense prediction.

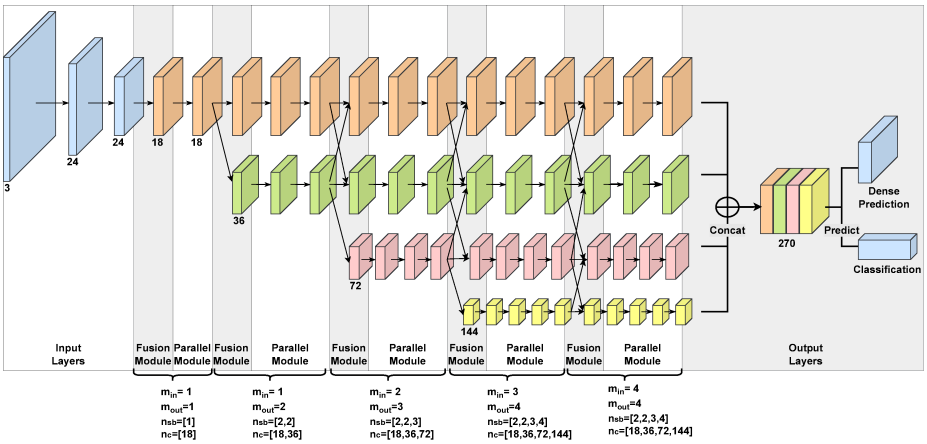


Fig. 10. Visualization of the adopted supernet architecture, where m_{in} and m_{out} denote the number of input and output branches in the fusion module; n_{sb} and n_c represent the number of searching blocks and channels in the parallel module, respectively.

B More Visualization of Visual Recognition Results

We further visualize the results of SuperTickets and S+P baselines on COCO keypoint and Cityscapes datasets under different pruning ratios, as shown in Fig. 11 and Fig. 12, respectively. We observe that S+P baselines work but miss some keypoints or semantic understandings under medium sparsity (e.g., 70/80%) while collapse under high pruning ratios (e.g, 90/95%); In contrast, our identified SuperTickets consistently work well among a wide range of pruning ratios, validating the effectiveness of our proposed SuperTickets.

Fig. 11. Visualization of the human pose estimation on COCO keypoint dataset under various pruning ratios.

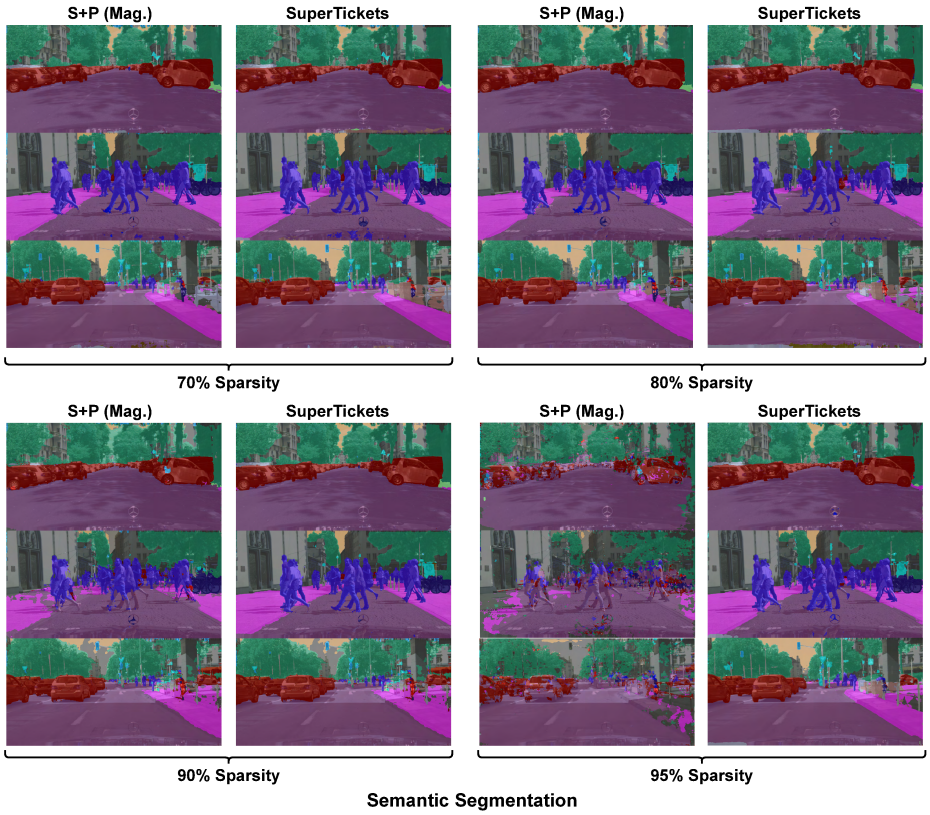


Fig. 12. Visualization of the streetview/semantic labels on Cityscapes dataset under various pruning ratios.