This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

DATA: Domain-Aware and Task-Aware Self-supervised Learning

Qing Chang [1,3,4,5] Junran Peng [2] Lingxi Xie [2] Jiajun Sun [1,3,4,5] Haoran Yin [1,3,4,5]
Qi Tian [2] Zhaoxiang Zhang*[1,3,4,5,6]
[1]University of Chinese Academy of Sciences, [2]Huawei Inc.
[3]Institute of Automation, Chinese Academy of Sciences
[4]National Laboratory of Pattern Recognition
[5]Center for Research on Intelligent Perception and Computing
[6]Centre for Artificial Intelligence and Robotics, HKISI_CAS

changqing2020@ia.ac.cn, jrpeng4ever@126.com
198808xc@gmail.com, {sunjiajun211,yinhaoran19}@mails.ucas.ac.cn
tian.qi1@huawei.com, zhaoxiang.zhang@ia.ac.cn

Abstract

*The paradigm of training models on massive data without label through self-supervised learning (SSL) and fine-tuning on many downstream tasks has become a trend recently. However, due to the high training costs and the unconsciousness of downstream usages, most self-supervised learning methods lack the capability to correspond to the diversities of downstream scenarios, as there are various data domains, different vision tasks and latency constraints on models. Neural architecture search (NAS) is one universally acknowledged fashion to conquer the issues above, but applying NAS on SSL seems impossible as there is no label or metric provided for judging model selection. In this paper, we present DATA, a simple yet effective NAS approach specialized for SSL that provides **D**omain-**A**ware and **T**ask-**A**ware pre-training. Specifically, we (i) train a supernet which could be deemed as a set of millions of networks covering a wide range of model scales without any label, (ii) propose a flexible searching mechanism compatible with SSL that enables finding networks of different computation costs, for various downstream vision tasks and data domains without explicit metric provided. Instantiated With MoCo v2, our method achieves promising results across a wide range of computation costs on downstream tasks, including image classification, object detection and semantic segmentation. DATA is orthogonal to most existing SSL methods and endows them the ability of customization on downstream needs. Extensive experiments on other SSL methods demonstrate the generalizability of the proposed method. Code is released at https://github.com/GAIA-vision/GAIA-ssl.*

*Corresponding author.

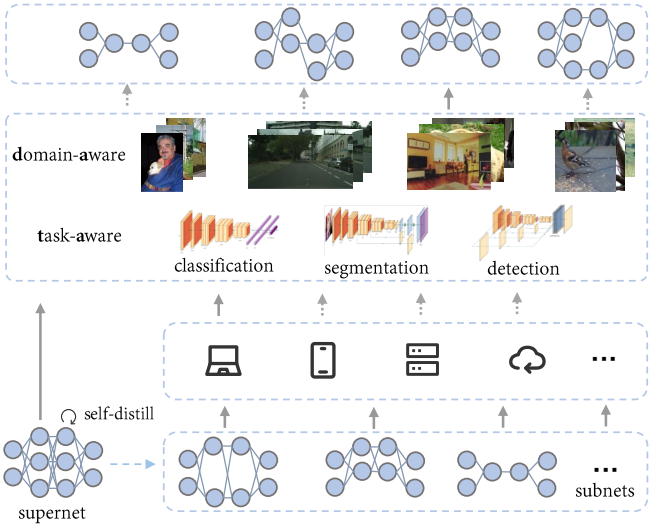

Figure 1. Illustration of how DATA works. We first build a supernet which is a set of many subnets, and train massive models simultaneously in the regime of self-supervised learning. Then, we propose a unsupervised searching method that enables domain-aware and task-aware model selection without any label. The mechanism enables self-supervised models to fit various scenarios including point,edge and cloud covering different vision tasks like image classification, object detection, and segmentation. Network architectures in figure are ploted by software PlotNeuralNet [25].

1. Introduction

As is universally agreed that deep learning algorithms are data-hungry, how to leverage the exponentially growing unlabelled data from open-source has become a huge challenge. Self-supervised learning (SSL), which utilizes the

inherent relationships of data to formulate supervision without manual annotation, has made remarkable progresses in both natural language processing (NLP) [14, 18, 34, 35] and computer vision (CV) [10, 19, 20] area. Despite its great success, to unleash the true power of SSL requires gigantic scale of data and unimaginable training budgets. This brings about a side-effect that it is extremely expensive to train models of various architectures to cover the heterogeneous downstream needs, considering most scenarios are requesting for models of different scales and different downstream vision tasks may desire different model architectures. It is usually believed that neural architecture search (NAS) is designed for solving the issues above. However, labels are so indispensable for existing NAS methods that it seems impossible to apply NAS in SSL, because there is no clue for model selection if no label or metric is provided. These reflections leave us two problems:

(1) **Is it possible to train networks of distinctive architectures simultaneously in SSL?** Gladly, previous methods [4, 5, 8, 44] have proved that training a supernet that comprises of millions of weight-sharing subnets is possible in the regime of supervised learning. Thus the hardship only lies in how to prevent the co-training of different networks from diverging when there is no strong and stable supervision. Some recent studies [8, 49] interpret the process of SSL of siamese based methods as a form of self-distillation that the query-branch works as a student and the teacher-branch works as a teacher. Thus if we stabilize the behaviour of the teacher, we could provide a relatively steady knowledge source for the heterogeneous students. In this work, we build a supernet training mechanism for siamese based SSL that we fix the *key-branch* with the maximum architecture of supernet as a teacher and vary only the architectures of the *query-branch*. Experiments show that this ensures the efficiency of convergence and greatly improves the capability of feature representation of small subnets. More importantly, this design of training supernet in SSL brings us the answer to the critical question below.

(2) How to judge the quality of a network if no label or metric is provided? It is generally agreed that *the bigger the better* works for deep neural networks when data is sufficient. Given a supernet that covers subnets of different sizes and the knowledge distillation behaviour of SSL, the distance between subnets and the maximum network naturally becomes a self-supervised metric for judging the quality of networks. This metric works well especially when there is a budget constraint for subnets. More discussions about this assumption are placed in Sec 6

We further extend our exploration to enable the searching process to be aware of the type of downstream tasks that different tasks adopt different types of features for measuring the distance of student and teacher. This greatly minimizes the gap transferring to downstream tasks while keeping the searching strategy plug-and-play.

As shown in Figure 1, our approach enables training models of various sizes all in one go and searching appropriate models specialized for specific downstream tasks, computation constraints and data domains. This entire pipeline does not require any label for training or model selection. Instantiated with MoCo v2 [11] , we validate our contributions on evaluating our models on several standard of self-supervised benchmarks. We also combine our approach with other existing SSL methods [19, 38, 49] to demonstrate the generalizability.

2. Related Work

2.1. Self-supervised learning

Self-supervised learning has become the main paradigm of unsupervised learning. It aims at building a good pretext to learn fruitful feature representation from data itself. These pretexts can be mainly separated by two categories: reconstruction-based which concludes colorization [48], spatial jigsaw puzzles [39], inpainting [45] and discriminant-based which contains rotation predict [26], instance-level contrastive [10, 19, 20], and fine-grained contrastive [38, 42].

Contrastive learning. For the first time, contrastive methods [10, 20] makes self-supervised training become comparable with supervised counterpart. Their method mainly focus on pulling representations of different views of the same image (positive pairs) closer and pushing representations of different images (negative pairs) away in the same time. Further, [7, 19] just use positive pairs to make network learn fruitful feature. These aforementioned methods pretext are sub-optimal in some extent for dense predict task (object detection, semantic segmentation, etc.). To relive this issue, fine-grained pretext [2, 24, 31, 38, 42] are proposed. Most of these methods already outperform supervised counterpart in some dense predict downstream tasks.

2.2. Neural Architecture Search

Neural architecture search aims at automating the architecture design process under certain constraints. [50, 51] proposes to use reinforcement learning with the metric on proxy datasets as the reward for solving this problem. But due to the unaffordable cost, one-stage NAS [1, 3, 6, 15, 30] are proposed, which train and search candidate architectures inside a single supernet. Though getting a specific architecture easily with those methods, we still need to train and search from scratch to get a new architecture once the constraints (such as latency, memory cost) changed. Further, researchers propose methods [4, 5, 9, 44, 46] to achieve training one supernet, which can contain a series of subnets

that cover a wide range of scenarios. Their effective supervised training methodology inspired us the most. In [29], it firstly shows that network architectures performance on self-supervised tasks like rotation prediction is linearly correlated with the performance on the supervised task which inspired the design of searching mechanism in our method.

3. Method

3.1. Preliminaries

We formulate the process of common siamese based SSL as a process of dynamic knowledge distillation [17, 49], which shares the same notion with [8]. To be convenient, we alias the *query-branch* as *student branch*, and the *key-branch* as *teacher branch* in our paper. Given N unlabeled samples $x_1, x_2, ..., x_N$, two views (x_i^s and x_i^t) are obtained on each sample through composition of different augmentations T and fed into a student $g(\cdot, \theta^s)$ and a teacher network $g(\cdot, \theta^t)$, parameterized by θ^s and θ^t respectively. In most cases, the teacher shares the exponential moving averaged (EMA) weights of student, namely, $\theta^t \leftarrow \lambda\theta^s + (1-\lambda)\theta^s$. We use $z_i^s = g(x^s, \theta^s)$ and $z_i^t = g(x^t, \theta^t)$ to denote the encoded features from student and teacher models, respectively. $H(z^s, z^t)$ is used to represent the similarity function. Taking MoCo [20] as an example, the InfoNCE loss [33] (Eq. 1) is adopted for training model:

$$\mathcal{L}_i = -\log \frac{\exp\left(H(g(x_i^s, \theta^s) \cdot g(x_i^t, \theta^t)/\tau\right)}{\sum_j^N \exp\left(H(g(x_i^s, \theta^s) \cdot g(x_j^t, \theta^t))/\tau\right)} \quad (1)$$

where τ is a temperature hyper-parameter [41].

3.2. Self-supervised Supernet Training

No single model could perfectly match the needs of heterogeneous downstream applications, as there might be different latency constraints, data domains and task gaps. Thus we aim to train a great deal of models together instead of training a single one in the regime of SSL, and we hope they cover a wide range of model scale. We extend the definition of a network from $g(x, \theta)$ to $g(x, \theta, \mathcal{A})$, with a new dimension meaning model architecture. Concretely, we build up a supernet Φ which contains numerous weight-sharing [46] subnets $g^{(k)}$ of various architectures $\mathcal{A}^{(k)}$, formulated as:

$$\begin{cases} \mathcal{A} = \left(\mathcal{A}^{(1)}, \dots \mathcal{A}^{(k)}, \dots \mathcal{A}^{(K)}\right) \\ \theta = \left(\theta^{(1)}, \dots \theta^{(k)}, \dots \theta^{(K)}\right) \end{cases} \quad (2)$$

where K is the total number of subnets. Particularly, we mark the largest model in supernet Φ as $g(\cdot, \theta^{(K)}, \mathcal{A}^{(K)})$, because weights of all the subnets $g(\cdot, \theta^{(k)}, \mathcal{A}^{(k)})$ are completely included by it. During training, we fix the architecture of *teacher-branch* as the model-K which means $\mathcal{A}^t = \mathcal{A}^{(K)}$, and vary the architecture of *student-branch* as $\mathcal{A}^s \in \mathcal{A}$. The weights of *teacher-branch* use the EMA version of the model-K, namely, $\theta^t \leftarrow \lambda\theta^{(K)} + (1-\lambda)\theta^{(K)}$.

In each training iteration, we random sample two network architectures $\mathcal{A}^{(m)}, \mathcal{A}^{(n)}$ from Φ together with the maximum one $\mathcal{A}^{(K)}$ to form a architecture set $\Omega = \{\mathcal{A}^{(m)}, \mathcal{A}^{(n)}, \mathcal{A}^{(K)}\}$. Following the conventional training regime in siamese-based SSL, we feed x_i^t to *teacher-branch* and generate the embedded feature of teacher $z_i^t = g(x_i^t, \theta^t, \mathcal{A}^t)$, and we feed x_i^s to the models in *student-branch* from Ω to get student features $\mathcal{Z}_i^s = \{z_i^{s(m)}, z_i^{s(n)}, z_i^{s(K)}\}$ where $z_i^{s(m)} = g(x_i^s, \theta^{s(m)}, \mathcal{A}^{(m)})$. With similarity measured by dot product, we apply InfoNCE loss on $\{(z_i^t, z_i^s) | z_i^s \in \mathcal{Z}_i^s\}$, back-propagate the gradients and update all the involved parameters. The whole training process is shown as pseudo-code in Algorithm 1

Algorithm 1 Self-Supervised Supernet Training

Require: Define supernet Φ with largest architecture $\mathcal{A}^{(K)}$. Choose the specific contrastive learning method to determine $criterion$. Initialize the neural network $g(\cdot, \theta^s, \mathcal{A}^{(K)})$ and $g(\cdot, \theta^t, \mathcal{A}^{(K)})$

for $i = 1, ..., T_{iters}$. **do**
  Get the min-batch of data x_i.
  Get two views of x_i. x_i^s, x_i^t.
  $optimizer.zero\_grad()$.
  $loss$ initialized with 0.
  $z_t = g(x_i^t, \theta^t, \mathcal{A}^{(K)})$.
  Sample two model architectures $\mathcal{A}^{(m)}, \mathcal{A}^{(n)}$ from Φ to construct set $\Omega = \{\mathcal{A}^{(m)}, \mathcal{A}^{(n)}, \mathcal{A}^{(K)}\}$.
  for $\mathcal{A}^{(k)}$ in Ω **do**
    $loss += criterion(z_t, g(x_i^s, \theta^{s(k)}, \mathcal{A}^{(k)}))$.
  end for
  $loss.backward()$.
  $optimizer.step()$.
  $\theta^t \leftarrow \lambda\theta^s + (1-\lambda)\theta^s$.
end for

Model space of supernet. We choose the popular ResNet [22] as the basic architecture in our work. Depth[1] and width[2] are adopted as factors to formulate the model space. Sharing the same notion with [22], the output feature maps of each stage are denoted as $(C1, C2, C3, C4, C5)$ for future use. As shown in Table 1, the depth of stage start from (2, 2, 5, 2) to (4, 6, 29, 4) with a step of (1, 2, 2, 1), and the width of $stem$ and each stage start from (32, 48, 96, 192, 384) to (64, 80, 160, 320, 640) with a step of (16, 16, 32, 64, 128).

3.3. Self-supervised Model Selection

This part reveals the core motivation behind our adoption of supernet in the regime of SSL. Beyond the value of

[1]Number of *bottleneck* blocks in each stage.
[2]Number of channels of 3×3 convolutions in each stage.

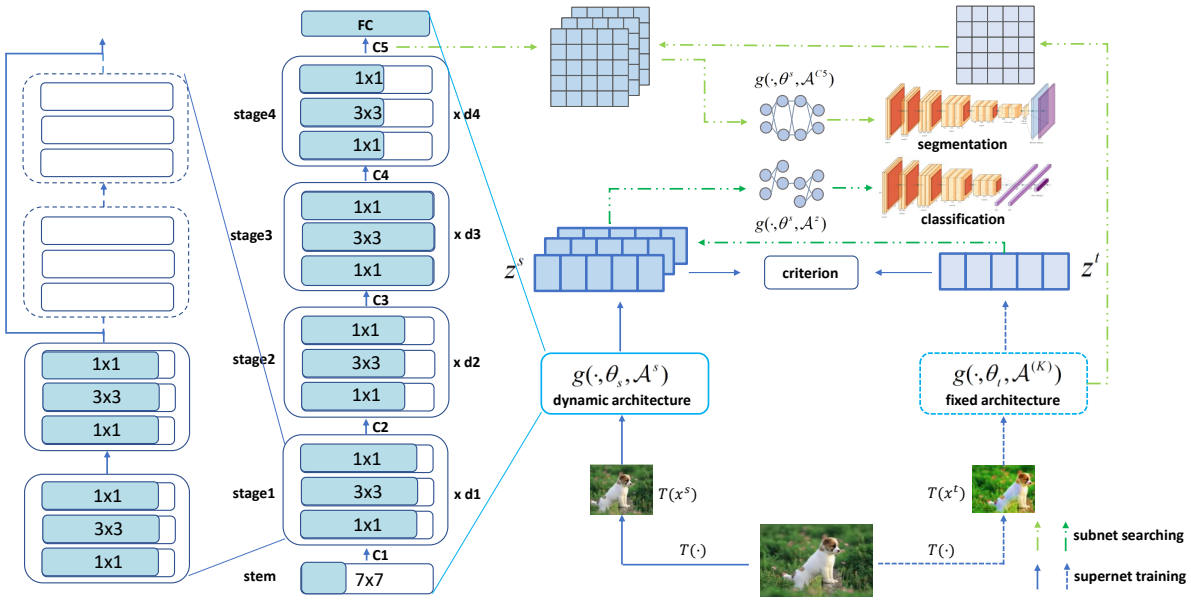

Figure 2. Pipeline of our method. It contains two stages. In the first stage, we fix the architecture of the *key-branch* as a steady teacher, and vary the architectures of the *query-branch* for supernet training. In the second stage, we propose a domain-aware and task-aware self-supervised metric for subnet searching, based on the similarity of task-specific features extracted from target dataset between subnets and model of the *key-branch*. d{1,2,3,4} : number of *bottleneck* blocks in this stage.

layer name	W_{range}	W_{step}	D_{range}	D_{step}
$stem$	[32, 64]	16	-	-
$stage1$	[48, 80]	16	[2, 4]	1
$stage2$	[96, 160]	32	[2, 6]	2
$stage3$	[192, 320]	64	[5, 29]	2
$stage4$	[384, 640]	128	[2, 4]	1

Table 1. Model space of supernet. Width and depth of each stage are sampled in range with certain step.

training massive models simultaneously, it presents a feasible metric for self-supervised NAS and provides abundant architecture candidates for searching. As is universally acknowledged that *the bigger the better* works mostly for deep neural networks, the distance between subnets and the largest model naturally becomes a metric for model selection.

Domain-awareness. This mechanism enables us to apply NAS on the downstream data during self-supervised learning, which shrinks the domain gap. Specifically, given a constraint of computation budget C, we feed the downstream data $\mathcal{D}$ into *teacher branch* and obtain the exemplar feature representations $z_i^t = g(x_i, \theta^t, \mathcal{A}^{(K)})$ for $x_i \in \mathcal{D}$. Then we randomly sample subnets under the constraint of budget and collect feature representations $z_i^{s(k)} = g(x_i, \theta^{s(k)}, \mathcal{A}^{(k)})$ for each model on the downstream data. We denote the similarity function by H' and the conquering model is selected to maximize the similarity on the entire downstream dataset:

$$\max_k \sum_i^{|D|} H'(z_i^t, z_i^{s(k)}) \tag{3}$$

The whole process does not involve any fully-supervised metric such as accuracy or precision.

Task-aware metrics. The gaps between different vision tasks are always unignorable. To enable our pre-trained supernet to serve various downstream vision tasks, the architecture search has to be conducted under task-aware metrics. Model selected under a single self-supervised metric is not suitable for different tasks. For instance, classifier of image classification task mostly handles the $avg\text{-}pooled$ feature that focuses on the global information of image, while in object detection, equipped with FPN [27], detectors use the multi-stage features of backbone for inference. These task-specific types of features matter most for the specific downstream task. Hence we search for different models by measuring the distance of the features directly used in head of downstream task. We call this task-aware metrics. The pipeline of our method is demonstrated in Figure 2.

For task of image classification, we direct use the feature of z. For object detection, we adopt feature of C5

for Faster-RCNN-C4 and features from C2-C5 for Faster-RCNN-FPN. And for task of semantic segmentation, we adopt features from C4-C5. The influence of task-aware metric is detailed in Table 9.

Similarity of relative relations. During the task-aware searching, the size of the feature map of subnets might be inconsistent with it of teacher. Thus we turn to utilize the relative relations of features to evaluate the similarity of two feature maps. We use $M \in \mathbb{R}^{C' \times HW}$ and $M' \in \mathbb{R}^{C \times HW}$ to represent features from student and teacher respectively, where H,W denote the height and width, C and C' represent number of channels, and $\{m_1, m_2, ..., m_{HW}\}$ means the feature in each column of matrix M. We define r_{ij} to express the vector relative relation of vector m_j and m_i as in Eq. 4, and the similarity of relative relation R on feature maps can be formulated in Eq. 5. We combine this with the task-aware metric, and conduct model selection in Eq. 3.

$$r_{ij} = -\log \frac{\exp\left(m_i \cdot m_j / \tau\right)}{\sum_k^{HW} \left(m_i \cdot m_k / \tau\right)} \tag{4}$$

$$R(M, M') = \sum_i^{HW} \sum_j^{HW} -r'_{ij} \log r_{ij} \tag{5}$$

4. Experiments

4.1. Experiments Setup

Datasets. We instantiate our method with the popular MoCo v2 [11] and train on ImageNet [13] which has ∼1.28 million images in 1000 categories. For the purpose of verifying transferability of models we search on downstream tasks, we experiment on ImageNet [13] semi-supervised classification, COCO [28] instance segmentation, PASCAL VOC [16] object detection, and Cityscapes [12] semantic segmentation. In the ablation study, to validate the generalizability of our method, we combine our method with other self-supervised learning methods [19,38,49] and train them on ImageNet-10%.

Training details. When supernet trained on Imagenet, we use SGD as the optimizer. The SGD weight decay is 0.000075. We train 200 epochs using a batch size of 1024 on 16 GPUs and an initial learning rate of 0.12. For supernet trained on ImageNet-10%, we all follow their official default settings.

Searching details. In order to verify that our method can effectively deal with various scenarios, we compute FLOPs for all of our subnets according to 224x224 input resolution and divide them into seven groups at 1 GFLOPs interval from 1 GFLOPs ∼ 8 GFLOPs. We randomly sample one hundred subnets in different groups and search the network in target dataset according to the task-aware metric to find the best one in every group. When verifying our method on various downstream tasks, we will display our experiment results of each group.

4.2. Ranking Correlation

Firstly, we verify that the similarity of the task-specific features between pre-trained subnets and the largest network can be a reliable indicator to assess the performance of these subnets. We sample fifty subnets uniformly and experiment on the above ImageNet-1% (IN-1%) semi-supervised classification, VOC object detection, COCO instance segmentation and Cityscapes semantic segmentation. We compute the Spearman [37] correlation between the similarity ranking and their final performance ranking. The results are shown in Table 2. For the low ranking correlation of semantic segmentation, we infer the reason is that the data distribution of the Cityscapes [12] dataset is largely different from the ImageNet [13] dataset used in the self-supervised training stage.

Architecture	Dataset	Feature	Correlation
ResNet-FC [22]	IN-1% [13]	z	0.90
FasterRCNN-C4 [36]	VOC [16]	C5	0.84
MaskRCNN-FPN [21]	COCO [28]	C2-C5	0.86
FCN [32]	Cityscapes [12]	C4-C5	0.63

Table 2. Ranking correlation. Architecture : The specific algorithm architecture adopted when transferring to downstream tasks. Feature : The feature utilized for model selection. IN-1% : ImageNet-1%.

4.3. Results on Various Downstream Tasks

Linear evaluation on ImageNet. We train supervised linear classifiers (a fully-connected layer with softmax) on frozen features after BN statics calibration for all of networks we search, following the procedure described in [43] (detailed in supplementary materials). We report 1-crop, top-1 classification accuracy on the ImageNet validation set. The result is shown in Figure 3 and the network architectures we search are described in appendix. We notice .The top-1 accuracy of linear evaluation of the model we search in Group 3G-4G is 68.5% which outperforms ResNet50 (3.8G, 67.5%) by 1%.

We find that relative small models benefit most from the training strategy. But the searched architecture in 7G∼8G group performs slightly worse than R101. We infer that when one model architecture outperforms the largest one, its output features are also far away from the largest teacher.

Semi-supervised classification on ImageNet. Next, following [10], we evaluate performance of the architectures

Model	FLOPs	Params	Depth	Width	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
R50* [11]	3.8G	25.5M	[3, 4, 6, 3]	[64, 64, 128, 256, 512]	38.7	59.2	42.3	35.5	56.2	37.9
R50† [40]	3.8G	25.5M	[3, 4, 6, 3]	[64, 64, 128, 256, 512]	38.6	59.5	42.1	35.2	56.3	37.5
Group	FLOPs	Params	Depth	Width	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
1G∼2G	1.8G	13.6M	[2, 2, 5, 2]	[32, 48, 96, 192, 512]	36.2	56.6	39.5	33.4	53.9	35.4
2G∼3G	2.7G	14.7M	[2, 2, 13, 2]	[48, 48, 96, 192, 384]	38.3	58.4	41.8	34.8	55.2	37.4
3G∼4G	3.7G	25.7M	[3, 2, 17, 3]	[32, 48, 96, 192, 512]	**39.9**	**60.2**	**43.5**	**36.0**	**57.1**	**38.6**
4G∼5G	4.2G	33.1M	[2, 2, 25, 2]	[64, 64, 128, 192, 384]	40.4	60.9	44.3	36.5	58.0	39.3
5G∼6G	5.9G	43.4M	[4, 6, 21, 4]	[32, 64, 96, 192, 640]	41.2	61.9	45.2	37.2	58.5	39.9
6G∼7G	6.6G	40.2M	[3, 6, 27, 3]	[64, 80, 96, 192, 640]	41.5	62.1	45.1	37.3	58.9	40.1

Table 3. Results of object detection and instance segmentation on COCO. * : Results of model pretrained through MoCo v2. † : Results of model pre-trained on ImageNet, fine-tuned on $train2017$ following 1x schedule and evaluated on $val2017$.

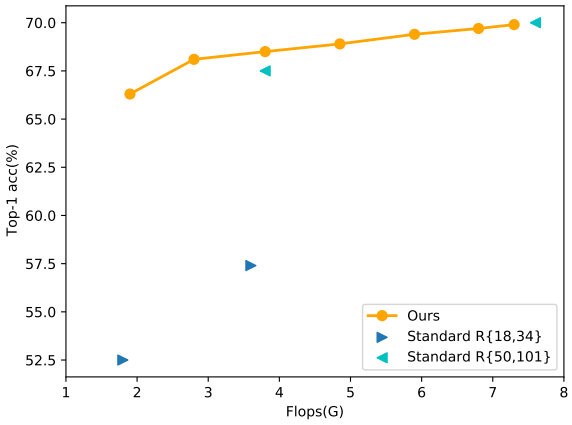

Figure 3. Linear evaluation on ImageNet. All of the classifiers are trained on the ImageNet train set with 100 epochs and evaluated on the ImageNet validation set. We compare it with standard ResNet architectures, which are all trained with MoCo v2 on ImageNet for 200 epochs.

Model	Params	Top-1		Top-5	
		1%	10%	1%	10%
R50* [11]	25.5M	39.8	61.8	68.3	85.1
Group	Params	1%	10%	1%	10%
1G∼2G	14.7M	45.8	61.9	73.7	84.9
2G∼3G	19.5M	46.5	63.0	74.7	85.8
3G∼4G	33.5M	**47.5**	**63.4**	**75.1**	**85.7**
4G∼5G	37.0M	47.8	64.3	75.3	86.4
5G∼6G	43.4M	49.0	65.1	76.2	86.7
6G∼7G	45.4M	49.3	65.4	76.3	87.0
7G∼8G	45.2M	49.5	65.6	76.5	86.9

Table 4. Results of semi-supervised classification on the 1% and 10% portion ImageNet. * : Results of models pre-trained through MoCo v2. † : Results of models pre-trained on ImageNet. All of the models are fine-tuned on the corresponding subset for 20 epochs and evaluated on the ImageNet validation set.

we search on ImageNet val set after fine-tuning on 1% and 10% subset of ImageNet $train$ set with annotations. For the convenience of description, we use ImageNet-1% and ImageNet-10% to represent the two subsets. The training procedure is detailed in supplementary materials. The results of Top-1 on the val set are reported in Table 4. We obtain +7.6% boost when fine-tuned on ImageNet-1% and gain +1.7% on ImageNet-10%. For ImageNet-1% semi-supervised setting, even the model selected from 1G∼2G group outperforms the ResNet50 baseline by a large margin.

COCO instance segmentation. For COCO instance segmentation, we follow the common setting [20] that fine-tune a Mask R-CNN detector (FPN) on COCO $train2017$ split (∼118k images) for all the architectures of our search with standard 1× schedule and evaluating on COCO $val2017$ split. The results are shown in Table 3. The pre-train architecture under 4 GFLOPs of our search outperforms the standard ResNet50 by $1.2AP$ in detection, and the promotion is $0.6AP$ for segmentation.

Object detection on PASCAL VOC. When transferring to VOC [16] object detection, following the [11]: a Faster R-CNN detector [36] (C4-backbone) is adopted. And it is trained on VOC $trainval07$+12 set for 24K iterations and evaluated on the VOC $test2007$ set. The results are presented in Table 5. With our framework, we can provide a model whose FLOPs and Params are similar to standard ResNet50, outperforms the R50 baseline by 1.2 AP.

Cityscapes semantic segmentation. Cityscapes [12] is a widely used benchmark for semantic segmentation. Following [38], we fine-tune the backbone of our search in the FCN [32] form on $train$ set (2975 images) for 40k iterations with batch size 16 and test on val set (500 images). The results are reported in Table 6. Although the correlation of our ranking strategy is fair, as shown in Table 2, we

Model	FLOPs	Params	AP	AP_{50}	AP_{75}
R50† [11]	3.8G	25.5M	53.5	81.3	58.8
R50* [11]	3.8G	25.5M	57.2	82.4	63.7
Group	FLOPs	Params	AP^b	AP^b_{50}	AP^b_{75}
1G∼2G	1.8G	12.2M	50.9	79.8	55.2
2G∼3G	2.8G	25.0M	58.0	82.7	64.5
3G∼4G	3.7G	25.6M	**58.5**	**83.0**	**65.0**
4G∼5G	4.2G	33.1M	59.1	83.2	65.3
5G∼6G	5.9G	43.4M	60.5	83.9	67.1
6G∼7G	6.9G	47.4M	60.4	83.5	67.0
7G∼8G	7.3G	45.3M	60.4	83.6	67.3

Table 5. Results of object detection on PASCAL VOC. * : Model pretrained by MoCo v2. † : Results of models pre-trained on ImageNet.

Model	Depth	Width	mIoU
R50†	[3, 4, 6, 3]	[64, 64, 128, 256, 512]	75.5
R50* [11]	[3, 4, 6, 3]	[64, 64, 128, 256, 512]	76.4
Group	Depth	Width	mIoU
1G∼2G	[2, 2, 5, 2]	[48, 48, 96, 192, 512]	72.7
2G∼3G	[3, 2, 9, 2]	[48, 48, 96, 192, 512]	75.2
3G∼4G	[2, 2, 17, 3]	[32, 48, 96, 192, 384]	**77.4**
4G∼5G	[2, 6, 19, 3]	[32, 48, 96, 192, 640]	77.0
5G∼6G	[2, 4, 25, 3]	[64, 64, 128, 192, 640]	78.1
6G∼7G	[4, 6, 23, 4]	[32, 64, 128, 192, 640]	77.6
7G∼8G	[4, 6, 21, 4]	[32, 64, 160, 192, 640]	78.2

Table 6. Results of semantic segmentation on Cityscapes. † : Models pre-trained on ImageNet. * : Models pre-trained through MoCo v2.

can still find the network outperforms the baseline with our effective training by 1 mIoU.

Transfer to other classification tasks. Furthermore, we evaluate our framework on more diverse classification datasets in VTAB [47] (detailed in supplementary materials). We only find the most similar pre-train architecture in 3G∼4G FLOPs group for the comparison with standard ResNet50. We perform fine-tuning on these datasets and report the results in Figure 4. The training details are detailed in supplementary materials. Compared to our MoCo v2 [20] baseline, we can get consistent improvement on these datasets. Although MoCo v2 is at a disadvantage in classification compared to supervised pre-training, thanks to the advantages of our training strategy and model customization, it can exceed the performance of supervised pre-training on some datasets. As the results show, the performance on eurasat [23] dataset attracts our attention. This dataset is a land cover classification dataset, and we find its data distribution is completely different from ImageNet [13]. Hence, when transferring to this dataset, these feature representations are of no use.

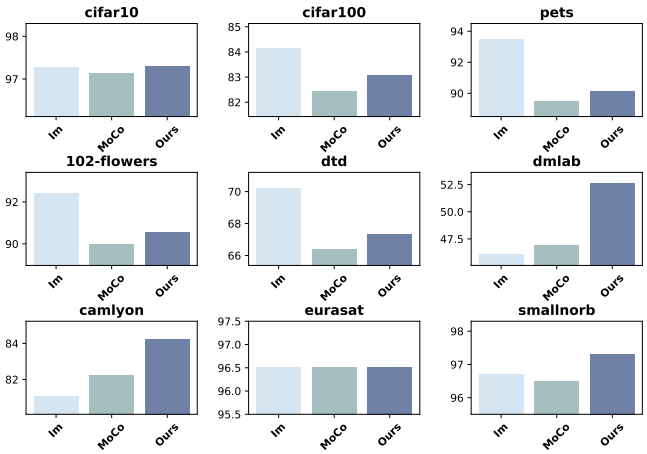

Figure 4. Transferablity to other data domains in image classification task. IN : Model pre-trained on ImageNet. MoCo : Model pre-trained through MoCo v2.

5. Ablation Study

Combining with contrastive learning methods. Our proposed pipeline is orthogonal to most contrastive learning methods. The reason for choosing MoCo v2 as baseline lies in its comprehensive performance on various downstream tasks. To demonstrate the generalizability of DATA, we instantiated with three classic contrastive self-supervised learning methods, BYOL [19], ReSSL [49] and DenseCL [38]. Due to the huge computation cost for training self-supervised models, we conduct experiments of these methods only on ImageNet-10%. Since [19, 49] are designed for classification task originally and [38] is mainly designed for the dense-prediction task, we report comparison results on area of their expertise. As shown in Table 7, consistent improvement could be observed.

Method	Task	performance
BYOL (repro)	Cls	23.7
Ours+BYOL	Cls	**24.8**
ReSSL (repro)	Cls	23.7
Ours+ReSSL	Cls	**26.7**
DenseCL (repro)	Det	49.1
Ours+DenseCL	Det	**50.2**

Table 7. Ablation study on combining DATA with other contrastive learning methods. Cls : Results of semi-supervised classification on the ImageNet-1% dataset. The top-1 accuracy adopted as metric. Det : Results of object detection on PASCAL VOC. AP is adopted as metric.

Ablation with teacher architecture. In this ablation study, we explore the impact of choice on teacher architec-

ture. Instead of fixing the architecture of *teacher branch* to the largest network, we compare with the setting using the same architecture as the sampled subnet in *student branch*, namely, $z_i^{t(k)} = g(x_i^t, \theta^{t(k)}, \mathcal{A}^{(k)})$. We base the experiment on MoCo v2 and train supernets on ImageNet-10%. Next, we extract the standard ResNet50 from supernet and evaluate it on the *val* set following the setting of linear classification. As reported in Table 8, we see that fixing the architecture of *teacher branch* is crucial, that the top-1 accuracy reaches 42.4% under linear evaluation protocol, 3.4% higher than the unfixed setting. This convey a message that a stable teacher is important for self-supervised supernet training. We also observe that this result outperform the vanilla MoCo v2 by 0.9%, which means supernet distillation is helpful over self-distillation.

Method	Fixed teacher arch	Top-1	Top-5
MoCo v2		41.5%	66.6%
Ours		39.0%	64.2%
Ours	✓	42.4%	67.2%

Table 8. Ablation on feature alignment. This table reports top-{1,5} accuracy of linear evaluation with 200 epochs on ImageNet-10%.

Ablation with task-aware metrics. Here we explore the effectiveness of task-aware metrics for model selection. For each downstream task, we select models according to metrics based on z, C5 and C2-C5 respectively. As reported in Table 9, task-aware metrics yield 0.4%, 0.7% and 0.7% improvements when task and metric are matched.

Dataset	Task	Architecture	Feature	Performance
IN-1% [13]	Semi-cls	ResNet-FC	z	47.5
			C5	47.3
			C2-C5	47.1
VOC [16]	Det	FasterRCNN-C4	z	57.8
			C5	58.5
			C2-C5	58.1
COCO [28]	Det	MaskRCNN-FPN	z	39.2
			C5	39.3
			C2-C5	39.9

Table 9. Ablation with task-aware metrics. IN-1% : ImageNet-1%. Semi-cls : Semi-supervised classification, using top-1 accuracy as metric. Det : Object detection, using AP@IoU as metric. All models share the similar computation budget with ResNet50.

Ablation with domain-awareness. We also explore the influence of domain-awareness. Specifically, we compare our searched models above with models searched through different datasets. Results are reported in Table 9. Note that these models are all in 3G∼4G group.

We find that searching by ImageNet seems to be an acceptable indicator for the performance of object detection on COCO. While in task of segmentation, searching without domain-awareness severely degrades the correlation(0.63 → 0.23) and mIoU(77.4 → 76.2).

Task	Target	Source	Correlation	Performance
Det	COCO	ImageNet	0.82	39.7
Det	COCO	COCO	0.86	39.9
Seg	Cityscapes	ImageNet	0.23	76.2
Seg	Cityscapes	Cityscapes	0.63	77.4

Table 10. Ablation on architecture search with domain awareness. Det : Object detection. Seg : Semantic segmentation. Target: Target dataset of downstream task where models are finetuned. Source: Source dataset for model selection.

6. Limitation

The major limitation of this work lies in the imbalanced training among subnets. Specifically, we find that the smaller two-thirds of subnets in supernet are trained well while the rest larger subnets are not. We infer that subnets could benefit more from knowledge distillation and most of their weights are always covered regardless of which subnet is sampled during training.

For our assumption, it indeed can not deal with this situation where candidate subnets are close to the largest teacher network. Because when one model architecture outperforms the largest one, its output features are also far away from the counterpart of largest teacher.

7. Conclusion

We have explored combining NAS with self-supervised learning and positive results are shown. Firstly, we manage to train massive weight-sharing subnets in a supernet simultaneously in the regime of self-supervised learning. More importantly, this mechanism of supernet training makes possible the unlabeled NAS since the feature distance between subnets and the largest network works perfectly as a self-supervised metric for model selection. Our work is orthogonal to most existing self-supervised learning methods and endows them the capability of customization on various downstream needs. We hope our approach could truly useful in real-world applications and our adventure on NAS in SSL could inspire more genius minds.

8. Acknowledgement

We thank Jiawei He, Shuwei Sun, Yuqi Wang, Lin Zhang and anonymous reviewers for their helpful discussions that improved this paper.

This work was supported in part by the Major Project for New Generation of AI (No.2018AAA0100400), the National Natural Science Foundation of China (No. 61836014, No. U21B2042, No. 62072457, No. 62006231).

References

[1] Youhei Akimoto, Shinichi Shirakawa, Nozomu Yoshinari, Kento Uchida, Shota Saito, and Kouhei Nishida. Adaptive stochastic natural gradient method for one-shot neural architecture search. In *ICML*, 2019. 2

[2] Amir Bar, Xin Wang, Vadim Kantorov, Colorado J Reed, Roei Herzig, Gal Chechik, Anna Rohrbach, Trevor Darrell, and Amir Globerson. Detreg: Unsupervised pretraining with region priors for object detection. *arXiv:2106.04550*, 2021. 2

[3] Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. Smash: one-shot model architecture search through hypernetworks. In *ICLR*, 2018. 2

[4] Xingyuan Bu, Junran Peng, Junjie Yan, Tieniu Tan, and Zhaoxiang Zhang. Gaia: A transfer learning system of object detection that fits your needs. In *CVPR*, 2021. 2

[5] Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once-for-all: Train one network and specialize it for efficient deployment. In *ICLR*, 2020. 2

[6] Han Cai, Ligeng Zhu, and Song Han. Proxylessnas: Direct neural architecture search on target task and hardware. In *ICLR*, 2019. 2

[7] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020. 2

[8] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 2, 3

[9] Minghao Chen, Houwen Peng, Jianlong Fu, and Haibin Ling. Autoformer: Searching transformers for visual recognition. In *ICCV*, 2021. 2

[10] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 2, 5

[11] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv:2003.04297*, 2020. 2, 5, 6, 7

[12] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*. 5, 6

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 5, 7, 8

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019. 2

[15] Xuanyi Dong and Yi Yang. Searching for a robust neural architecture in four gpu hours. In *CVPR*, 2019. 2

[16] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010. 5, 6, 8

[17] Zhiyuan Fang, Jianfeng Wang, Lijuan Wang, Lei Zhang, Yezhou Yang, and Zicheng Liu. Seed: Self-supervised distillation for visual representation. In *ICLR*, 2021. 3

[18] Luciano Floridi and Massimo Chiriatti. Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30(4):681–694, 2020. 2

[19] 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. In *NeurIPS*, 2020. 2, 5, 7

[20] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 2, 3, 6, 7

[21] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 5

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3, 5

[23] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019. 7

[24] Olivier J Hénaff, Skanda Koppula, Jean-Baptiste Alayrac, Aaron van den Oord, Oriol Vinyals, and João Carreira. Efficient visual pretraining with contrastive detection. In *ICCV*, 2021. 2

[25] Haris Iqbal. Harisiqbal88/plotneuralnet v1.0.0. https://doi.org/10.5281/zenodo.2526396, Dec. 2018. 1

[26] Nikos Komodakis and Spyros Gidaris. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018. 2

[27] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 4

[28] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 5, 8

[29] Chenxi Liu, Piotr Dollár, Kaiming He, Ross Girshick, Alan Yuille, and Saining Xie. Are labels necessary for neural architecture search? In *ECCV*, 2020. 3

[30] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. In *ICLR*, 2019. 2

[31] Songtao Liu, Zeming Li, and Jian Sun. Self-emd: Self-supervised object detection without imagenet. *arXiv:2011.13677*, 2020. 2

[32] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 5, 6

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

[34] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018. 2

[35] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 2

[36] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 5, 6

[37] C Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101, 1904. 5

[38] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *CVPR*, 2021. 2, 5, 6, 7

[39] Chen Wei, Lingxi Xie, Xutong Ren, Yingda Xia, Chi Su, Jiaying Liu, Qi Tian, and Alan L Yuille. Iterative reorganization with weak spatial constraints: Solving arbitrary jigsaw puzzles for unsupervised representation learning. In *CVPR*, 2019. 2

[40] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 6

[41] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018. 3

[42] Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In *CVPR*, 2021. 2

[43] Jiahui Yu and Thomas S Huang. Universally slimmable networks and improved training techniques. In *CVPR*, 2019. 5

[44] Jiahui Yu, Pengchong Jin, Hanxiao Liu, Gabriel Bender, Pieter-Jan Kindermans, Mingxing Tan, Thomas Huang, Xiaodan Song, Ruoming Pang, and Quoc Le. Bignas: Scaling up neural architecture search with big single-stage models. In *ECCV*, 2020. 2

[45] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *CVPR*, 2018. 2

[46] Jiahui Yu, Linjie Yang, Ning Xu, Jianchao Yang, and Thomas Huang. Slimmable neural networks. In *ICLR*, 2019. 2, 3

[47] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, et al. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv preprint arXiv:1910.04867*, 2019. 7

[48] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, 2016. 2

[49] Mingkai Zheng, Shan You, Fei Wang, Chen Qian, Changshui Zhang, Xiaogang Wang, and Chang Xu. Ressl: Relational self-supervised learning with weak augmentation. In *NeurIPS*, 2021. 2, 3, 5, 7

[50] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. In *ICLR*, 2017. 2

[51] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *CVPR*, 2018. 2