# SCARLET-NAS: Bridging the Gap between Stability and Scalability in Weight-sharing Neural Architecture Search

Xiangxiang Chu<sup>1</sup>, Bo Zhang<sup>1</sup>, Jixiang Li<sup>1</sup>, Qingyuan Li<sup>2</sup>, and Ruijun Xu<sup>1</sup>

<sup>1</sup> Xiaomi AI Lab {chuxiangxiang,zhangbo1,lijixiang,xuruijun}@xiaomi.com
<sup>2</sup> Xiaomi IoT liqingyuan@xiaomi.com

Abstract. To discover powerful yet compact models is an important goal of neural architecture search. Previous two-stage one-shot approaches are limited by search space with a fixed depth. It seems handy to include an additional skip connection in the search space to make depths variable. However, it creates a large range of perturbation during supernet training and it has difficulty giving a confident ranking for subnetworks. In this paper, we discover that skip connections bring about significant feature inconsistency compared with other operations, which potentially degrades the supernet performance. Based on this observation, we tackle the problem by imposing an equivariant learnable stabilizer to homogenize such disparities (see Fig.1). Experiments show that our proposed stabilizer helps to improve the supernet's convergence as well as ranking performance. With an evolutionary search backend that incorporates the stabilized supernet as an evaluator, we derive a family of state-ofthe-art architectures, the SCARLET<sup>3</sup> series of several depths, especially SCARLET-A obtains 76.9% top-1 accuracy on ImageNet<sup>4</sup>.

## 1 Introduction

Incorporating scalability into neural architecture search is crucial to exploring efficient networks. The handcrafted way of scaling models up and down is to stack more or fewer cells [13,37]. However, model scaling is nontrivial which involves tuning width, depth, and resolution altogether. To this end, a compound scaling method is proposed in [29], it starts with a searched mobile baseline EfficientNet-B0 and 'grid-search' the combination of these three factors to achieve larger models. In this paper, we are mainly concerned about finding models of varying depths, while the input resolution is kept fixed since it can be simply scaled manually.

To achieve such scalability, we first need to construct a search space of variable depths. To this end, skip connections are commonly used in differentiable approaches [2,31], but they face a common issue of undesired *skip connection aggregation* as noted by [3,34], which yields non-optimal results. Recent advances

 $<sup>^3</sup>$  SCAlable supeRnet with Learnable Equivariant sTablizer

<sup>&</sup>lt;sup>4</sup> Models: https://github.com/xiaomi-automl/SCARLET-NAS

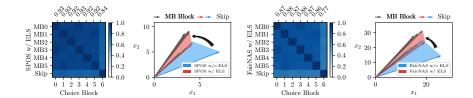


Fig. 1. ELS helps to calibrate feature inconsistencies in scalable supernets with boosted cosine similarity (compared to Fig.4) and reduced angles among feature vectors (from blue shade to red). The cosine similarity is calculated on the third layer's output feature maps from 7 paralleled choice blocks (MobileNetV2's blocks numbered 0 to 5 and a skip connection). Each block's feature vectors are projected to 2-dimensional space  $(x_1, x_2)$  to draw their relative angles (shaded in color). Other layers also have similar results. Left Pair: Single Path One-Shot [12], Right Pair: FairNAS [6].

in one-shot approaches take a two-stage mechanism: single-path supernet optimization and searching [12,6]. A supernet is an embodiment of the search space, whose single path is a candidate model. Their single-path paradigm is more efficient and less error-prone, which also potentially avoids the aggregation problem but they carefully removed skip connections from search space. In this light, we integrate skip connections in their search space under the same single-path setting for a comprehensive investigation. We name the supernet in this new search space as a scalable supernet.

Our contributions can be summarized as follows,

**First**, we are the first to thoroughly investigate scalability in one-shot neural architecture search. We discover that a vanilla training of scalable supernets suffers from instability (see Fig. 3) and leads to weak evaluation performance. As FairNAS [6] suggests that feature similarity is critical for single-path training, we find that this requirement is rigorously broken by skip connections (Fig. 4).

**Second**, based on the above observation, we propose a simple *learnable stabilizer* to calibrate feature deviation (see Fig.1). It is proved effective to restore stability (see Fig. 3) while all submodels still have invariant representational power. Experiments on NAS-Bench-101 [33] testify that it also substantially improves the ranking performance which is crucial for the second searching stage. Our pipeline is exemplified in Fig. 2.

Last but not the least, we perform a single proxyless evolutionary search on ImageNet after training the scalable supernet. The total cost is 10 GPU days. Three new state-of-the-art models of different depths are generated. Specifically, SCARLET-A obtains 76.9% top-1 accuracy on ImageNet with 25M fewer FLOPS than EfficientNet-B0 (76.3%)<sup>5</sup>. Moreover, we manually upscale the searched models with zero cost to have comparable FLOPS with EfficientNet variants and we also achieve competitive results.

<sup>&</sup>lt;sup>5</sup> Searching EfficientNet-B0 is similar to MnasNet [28] which takes 2304 TPU days.

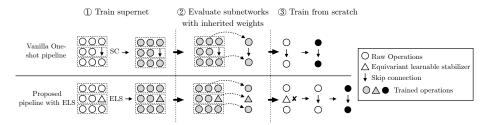


Fig. 2. Our proposed SCARLET-NAS pipeline where the scalable supernet is stabilized by ELS, which also provides good ranking ability compared with the vanilla approach. ELS is removed from the final subnetwork to train from scratch. Note SC and ELS can appear on each row (layer), only one is drawn for brevity.

# 2 Preliminary Background

#### 2.1 Single-Path Supernet Training

The weight-sharing mechanism is now widely applied in neural architecture search as it saves a tremendous amount of computation [23,22,1]. It is usually embodied as a supernet that incorporates all subnetworks. The supernet is trained till convergence only once, from which all subnetworks can inherit weights for evaluation (so-called one-shot models) without extra fine-tuning. It is thus named the one-shot approach, as opposed to those who train each child network independently [37,28]. Methods vary on how to train the supernet. In this paper, we concentrate on the single-path way [12,6], which is more memory-friendly and efficient.

Single Path One-Shot [12] utilizes a supernet  $\mathcal{A}$  with 20 layers, and there are 4 choice blocks per layer based on ShuffleNet [35]. The total size of the search space reaches  $4^{20}$ . It uniformly samples a single-path model (say a with weights  $W_a$ ) to train at each step, after which only this activated path in the supernet gets its weights  $W_a$  updated. Formally, this process is to reduce the overall training loss  $\mathcal{L}_{train}$  of the supernet,

$$W_{\mathcal{A}} = \operatorname{argmin}_{W} \mathbb{E}_{a \sim \Gamma_{A}} [\mathcal{L}_{train}(\mathcal{A}(a, W_{a}))] \tag{1}$$

Notice that it differs from the nested manner in differential approaches [22,10] where  $\Gamma$  is not fixed but used as a representation for variable architectural weights.

FairNAS [6] rephrases each supernet training step as training m single-path models either sequentially or in parallel. These models are built on choice blocks uniformly sampled without replacement (denoted as  $a \sim \Psi_{\mathcal{A}}$ ). During each step, all blocks in the supernet are trained once. The weights are aggregated and also updated once in a single step. It can be formulated as,

$$W_{\mathcal{A}} = \operatorname{argmin}_{W} \mathbb{E}_{a \sim \Psi_{\mathcal{A}}} \left[ \frac{1}{m} \sum_{i}^{m} \mathcal{L}_{train} (\mathcal{A}(a_{i}, W_{a_{i}})) \right]$$
 (2)

By ensuring the same amount of training for each block, FairNAS achieves a notable improvement in supernet performance. Interestingly enough, features learned by each block (of the same layer) in thus-trained supernet have high channel-wise similarities. This will be later proved a useful hint to restore training stability when skip connections are involved.

## 2.2 Model Ranking

Searching is essentially based on ranking. Incomplete training can give a rough guess [37] but it is too costly. Differentiable methods [22] consider the magnitude of architectural coefficients as each operation's importance. However, there is a large discrepancy when discretizing such continuous encodings. As we are focusing on the two-stage weight-sharing neural architecture search method, we rely on the supernet to evaluate models. It is thus of uttermost importance for it to have a good model ranking ability. FairNAS [6] has shown that *strict fairness* during supernet training has a strong impact on it. In particular, they adopted Kendall Tau [17] to measure the correlation between the performance of one-shot models (predicted by the supernet) and stand-alone models (trained from scratch). Tau value ranges from -1 to 1, meaning the order is completely inverted or identical. Ideally, we would like a tau of 1, which gives the exact ground truth ranking of submodels.

## 3 Training Instability of Scalable Supernet

## 3.1 Degraded Supernet Performance

The skip connection plays a role in changing depths for architectures in MobileNetV2's block-level search space [2,31]. We detail it as  $S_1$  and its variant  $S_2$  in C.1. To investigate scalability in one-shot approaches, we train the supernet in the previously discussed single-path fashion (Section 2.1) in search space  $S_1$ . Surprisingly, we find them suffering from severe **training instability**, which is illustrated in Fig. 3. Unlike the reported stable training process for the supernets without skip connections [12,6], we instead observe much higher variances (shadowed in blue at the top of Fig. 3) and lower training accuracies (solid line in blue)

Training instability also deteriorates one-shot model performance. We sample 1024 models to measure their accuracies on the ImageNet validation dataset. Fig. 3 demonstrates that the majority of one-shot models from both SPOS (bottom left in blue) and FairNAS (bottom right in blue) are underestimated, which are mainly close to 0. This phenomenon hasn't been observed in reduced  $S_1$  (without skip connections) by previous work.

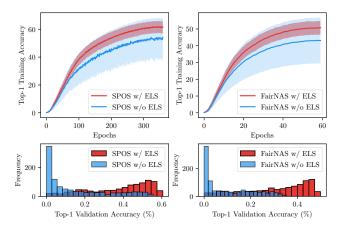


Fig. 3. Training supernet with Single Path One-Shot [12] and FairNAS [6] on ImageNet with and without Equivariant Learnable Stabilizer (ELS) in search space  $S_1$ . Top: The supernets with ELS enjoy better convergence (red thick lines) and small variance (red shaded area). Bottom: Histogram of randomly sampled 1k one-shot models' accuracies. Supernets with ELS have an improved estimation of subnetworks.

In particular, we need to neither overestimate nor underestimate the sampled submodels. This is hard for the scalable supernet trained so far. We can easily draw an example in Table 1, where model A is underestimated with only 1% accuracy and B overestimated (49%, much better than A). The ground truth is just the opposite, A has 74% which is better than B with 73.3%. We later show how we design an ELS for the supernet training to rectify this mistake.

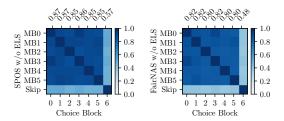
**Table 1.** ImageNet performance of model A and B (denoted by the choice block IDs) in  $S_1$ . Both are mistakenly estimated by the supernet trained w/o ELS. Instead, enabling ELS gives the right ranking.

Models	Top-1 (%)	Top-1 (%)	Top-1 (%)
$(in S_1)$	(est. w/o ELS)	(est. w/ ELS)	(standalone)
A(0,5,0,6,3,2,3,0,2,1,3,5,2,4,4,4,5,3,6)	1.0	53.1	74.0
B(5,0,1,0,2,6,6,4,3,1,5,1,0,2,4,4,1,1,2)	49.5	49.6	73.3

## 3.2 Skip Connections Break Feature Similarity

A well-trained supernet matters for one-shot models' ranking. We are thus driven to unveil what causes such a phenomenon to find a cure for stabilizing the training process.

Inspired by the analysis of the underlying working mechanism in the singlepath training [6], we pick the outputs of the third layer (for an example) in the formerly trained supernets to calculate their cosine similarities across different choice blocks, which are depicted as  $7 \times 7$  similarity matrices in Fig. 4. The first six inverted bottlenecks of different configurations yield quite similar high-dimensional features (with a shape of  $32 \times 28 \times 28$ ) and their cosine similarities are high (all above 0.85). Meanwhile, the feature maps from the skip connection (the last choice block) are quite distinct from other blocks and the average cosine similarity is below 0.6. This disparity is observed in both training methods.



**Fig. 4.** Cosine similarity matrices of the third layer's outputs (averaged on 32 channels of  $28 \times 28$  feature maps) from 7 choice blocks of supernets trained without ELS. The average similarity is shown as x-axis at the top. The skip connection yields different feature maps from others. **Left:** Single Path One-Shot [12], **Right:** FairNAS [6]

Feature disparity troubles the training for the next layer and consequently the whole supernet. As the fourth layer randomly selects one output from the third layer, the unique skip connection disrupts feature similarities. This discrepancy of inflowing features (occurs in other layers too) will get magnified layer by layer and finally deteriorate supernet training. This is shown on the top of Fig. 3. What's worse, it makes big trouble for the supernet to predict submodels' performance. Such a supernet becomes nearly useless because it severely underestimates or overestimates candidate architectures, shown at the bottom of Fig. 3. Therefore, we attribute the instability to low similarities of features across different paralleled choices, mainly from skip connections.

# 4 Scalable Neural Architecture Search

### 4.1 Improve Supernet Training with a Learnable Stabilizer

Based on the previous discussion, one direct approach to stabilize the training process is to boost the cross-block similarities by replacing the *parameter-free* skip connection with a learnable stabilizer. Ideally, the stabilizer will deliver similar features as other choice blocks. What's more important, the stabilizer must be equivariant in terms of representational capacity since we want to remove it eventually (see the third step in Figure 2). This is detailed as Definition 1.

**Definition 1.** Equivariant Learnable Stabilizer. A plug-in learnable stabilizer is called an Equivariant Learnable Stabilizer (ELS) iff a model with such a stabilizer is exactly equivalent to the one without it in terms of representational capacity.

For a search space S like  $S_1$  with n choices per layer, we denote  $x_l^{c_l}$  as the input with  $c_l$  channels to layer l, and  $f_l^o$  the o-th operation function in that layer. Without loss of generality, we put the skip connection as the last choice, while other choices all start with a convolution operation. The equivalence requirement for an equivariant learnable stabilizer function  $f_l^{ELS}$  can then be formulated as,

$$f_{l+1}^{o}(x_l^{c_l}) = f_{l+1}^{o}(f_l^{ELS}(x_l^{c_l})), \forall o \in \{0, 1, 2, ..., n-1\}.$$
(3)

As for S, we can utilize the property of matrix multiplication to find a simple ELS function: a  $1 \times 1$  convolution without batch normalization or activation. This is given as Lemma 1 and proven in the A.

**Lemma 1.** Let  $f_l^{ELS} = Conv_{(c_l,c_{l+1},1,1)}$ , then Equation 3 holds.

By adopting the learnable  $1 \times 1$  convolution as an ELS, we observe improved stability in supernet training and better evaluation of subnetworks (Fig. 3). We still maintain scalability since we can remove ELS based on Equation 3.

## 4.2 Neural Architecture Search with the Scalable Supernet

Being a two-stage one-shot method like [1,12,6], we have so far focused on supernet training. For the second searching stage, evolutionary algorithms are mostly used. For instance, FairNAS [6] utilizes the well-known NSGA-II algorithm [8] where they examine three objectives: classification accuracies, multiply-adds and the number of parameters. In practice, they are of different importance. We are more concerned about accuracies (performance) and multiply-adds (speed) than the number of parameters (memory cost), which calls for a weighted solution like [5]. It is however nontrivial for our scalable search space. First of all, models with too many skip connections are easily sorted as frontiers because of low multiply-adds. Although such a model dominates others but it usually comes with a low accuracy which is not desired. So we set a minimum accuracy requirement  $acc_{min}$ . Second, we are searching models for mobile deployment, where we should encourage increasing the number of parameters to prevent underfitting rather than overfitting [35]. Last, for practical reasons, we also need to set maximum multiply-adds  $madds_{max}$ . Formally, we describe our searching process as a constrained multi-objective optimization as follows,

$$max \quad \{acc(m), -madds(m), params(m)\}, \quad m \in \text{ search space } S$$
 $s.t. \quad w_{acc} + w_{madds} + w_{params} = 1, \forall w >= 0$ 
 $acc(m) > acc_{min}, madds(m) < madds_{max}.$ 

$$(4)$$

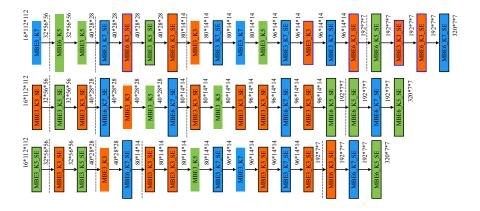
Specifically, we adopt a similar evolutionary searching algorithm based on NSGA-II [8] as in FairNAS [6] with some modifications. For handling weights of different objectives, we make use of weighted crowding distance [11] for non-dominated sorting. We set  $w_{acc}=0.4, w_{madds}=0.4, w_{params}=0.2$ . The constraints are set to  $madds_{max}=500M$  and  $acc_{min}=0.4$ . Notice that we treat these two constraints in sequential order to reduce cost. As calculating multiply-adds is much faster than accuracies, models violating  $madds_{max}$  are immediately removed for further evaluation. The whole search pipeline is presented in Algorithm 1 and Fig. 10 (both in B).

## 5 Experiments

## 5.1 Dataset, Training, and Searching

**Dataset.** For training and searching, we use the ILSVRC2012 dataset [9]. To be consistent with previous work [28], the validation set consists of 50k images selected from the training set. The original validation set serves as the test set.

**Supernet Training.** For FairNAS experiments in  $S_1$ , we follow [6] except that we train the supernet for 60 epochs. It costs nearly 8 GPU days. For SPOS, we train it for 360 epochs to have the same amount of weight updates per block. As for  $S_2$  with more choices, we use the same setting except for a smaller batch size of 256, which results in higher top-1 accuracy on average.



**Fig. 5.** The architectures of SCARLET-A, B and C (from top to bottom). Downsampling points are indicated by dashed lines. The stem and tail parts are omitted for brevity.

**Evolutionary Searching.** We search proxylessly in  $S_2$  on ImageNet. The evolution covered 8400 models (a population of 70 models evolved for 120 generations). It costs 2 GPU days on a Tesla V100. The final architectures SCARLET-A, B and C (shown in Fig. 5) are sampled from the Pareto front at equal distance

and are trained from scratch. Due to the equivalence requirement, we remove ELS to achieve two competitive models with shorter depths, SCARLET-B and C.

Single Model Training. To train the selected single model, we follow Mnas-Net [28] with vanilla Inception preprocessing [27]. We train EfficientNet and SCARLET models without AutoAugment [7] to have a fair comparison with state-of-the-art architectures. The batch size is 4096. The initial learning rate is 0.256 and it decays at an amount of 0.01 every 2.4 epochs. The dropout with a rate 0.2 [25] is put before the last FC layer. The weight decay rate  $(l_2)$  is  $1.0 \times 10^{-5}$ . The RMSProp optimizer has a momentum of 0.9.

## 5.2 ImageNet Classification

**Table 2.** Comparison with state-of-the-art architectures on ImageNet classification task. $^{\ddagger}$ : model trained from scratch by us without AutoAugment.

Models	×+	Params	Top-1	Top-5
	(M)	(M)	(%)	(%)
MobileNetV2 [24]	300	3.4	72.0	91.0
MobileNetV3 [14]	219	5.4	75.2	92.2
MnasNet-A1 [28]	312	3.9	75.2	92.5
MnasNet-A2 [28]	340	4.8	75.6	92.7
FBNet-B [31]	295	4.5	74.1	-
Proxyless-R [2]	320	4.0	74.6	92.2
Proxyless GPU [2]	465	7.1	75.1	-
Single-Path [26]	365	4.3	75.0	92.2
Single Path One-Shot [12]	328	3.4	74.9	92.0
FairNAS-A [6]	388	4.6	75.3	92.4
MixNet-M [30]	360	5.0	$76.6^{\dagger} (77)$	93.2
EfficientNet B0 [29]	390	5.3	76.3	93.2
SCARLET-A (Ours)	365	6.7	76.9	93.4
SCARLET-B (Ours)	329	6.5	76.3	93.0
SCARLET-C (Ours)	280	6.0	75.6	92.6

Comparison of State-of-the-art Mobile Architectures We give full train results of the SCARLET series on ImageNet dataset in Table 2. Although in absence of AutoAugment tricks [7], SCARLET-A still clearly surpasses EfficientNet-B0 (+0.6% higher accuracy) using fewer FLOPS. The shallower model SCARLET-B achieves 76.3% top-1 accuracy with 329M FLOPS, which exceeds several models of a similar size by a clear margin: MnasNet-A1 (+1.1%), Proxyless-R (+1.7%). Notably, to be comparable to our shallowest model SCARLET-C (75.6%), MnasNet-A1 comes with 21% more FLOPS at the cost of 200× GPU

days. Even without mixed convolution, SCARLET-A still outperforms MixNet-M [30], which has 76.6% accuracy when we trained it with the same tricks. We further give a closer examination of the SCARLET series in C.5.

Comparison of Models at a Larger Scale Higher accuracy requirements beyond mobile settings are also considered. To be comparable with EfficientNet's scaled variants, we simply *manually upscale* our SCARLET baseline models to have the same resolution and FLOPS without any extra tuning cost. We compare the results with other state-of-the-art methods in Table 3.

**Table 3.** Single-crop results of scaled architectures on ImageNet validation set. \*: Retrained without fixed AutoAugment (AA),†: w/o fixed AA.

Methods	Resolution	Depth	Channel	$\times +$	Params	Top-1	Top-5
		(×)	(x)	(B)	(M)	(%)	(%)
DenseNet-264 [16]	224×224	-	-	6	34	77.9	93.9
Xception [4]	$299 \times 299$	-	-	8.4	23	79.0	94.5
EfficientNet B2 [29]	$260 \times 260$	1.2	1.1	1.0	9.2	$79.4^*$	$94.7^{*}$
$\textbf{SCARLET-A2}^\dagger$	$260 \times 260$	1.0	1.4	1.0	12.5	79.5	94.8
ResNeXt-101 [32]	$320 \times 320$	-	-	32	84	80.9	95.6
PolyNet [36]	$331 \times 331$	-	-	35	92	81.3	95.8
SENet [15]	$320 \times 320$	-	-	42	146	82.7	96.2
EfficientNet B4 [29]	$380 \times 380$	1.8	1.4	4.2	19	82.6	96.3
$\textbf{SCARLET-A4}^\dagger$	$380 \times 380$	2.0	1.4	4.2	27.8	82.3	96.0

At the level of 1 billion FLOPS, while EfficientNet-B2 is based on grid search at a very high cost on GPUs [29], our SCARLET-A2 (79.5%) is from upscaling with zero cost. No AutoAugment tricks are applied for a fair comparison. Moreover, Xception [4] uses 8 times more FLOPS to reach 79.0%. Notably, our SCARLET-A4 achieves new state-of-the-art top-1 accuracy 82.3% again without extra costs using only 4.2 billion FLOPS. By contrast, SENet [15] uses  $10\times$  FLOPS.

## 5.3 Transferability to CIFAR-10

Table 4 shows our transfer results on CIFAR-10 dataset [19]. We utilize similar training settings from [18]. In particular, each model is loaded with ImageNet pre-trained weights and finetuned for 200 epochs with a batch size of 128. The initial learning rate is set to 0.025 with a cosine decay strategy. We also adopted AutoAugment policy for CIFAR-10 [7]. The dropout rate is 0.3. To achieve comparable top-1 accuracy as NASNet-A Large, our SCARLET-A only uses 33× fewer FLOPS. SCARLET-B doesn't utilize the mixed convolution but it is still comparable to MixNet [30]. In particular, our smallest model SCARLET-C is close to MixNet-M, saving 22% FLOPS.

Models Input Size  $\times + (M)$ Top-1 (%) NASNet-A Large [37]  $331 \times 331$ 12030 98.00 MixNet-M [30]  $224 \times 224$ 35297.92SCARLET-A  $224 \times 224$ 36498.05 SCARLET-B  $224{\times}224$ 328 97.93 SCARLET-C  $224 \times 224$ 27997.91

**Table 4.** Transferring SCARLET models to CIFAR-10. †: Reported by [18].

## 5.4 Object Detection

To verify the transferability of our models on the object detection task, we utilize drop-in replacements of backbones of the RetinaNet framework (Res101+FPN) [20]. To make fair comparisons, we focus on the mobile settings of the backbone. All methods are trained for 12 epochs on COCO dataset [21] with a batch size of 16 (train2017 for training and val2017 for reporting results). The initial learning rate is 0.01 and decayed by 0.1 on epoch 8 and 11. Compared with recent NAS models in Table 5, we utilize fewer FLOPS to have better results, suggesting a better transferability.

Table 5. Object detection result of various drop-in backbones on the COCO dataset.

Backbones	×+	Acc	AP	$AP_{50}$	$AP_{75}$	$AP_S$	$AP_M$	$AP_L$
	(M)	(%)	(%)	(%)	(%)	(%)	(%)	(%)
MnasNet-A2 [28]	340	75.6	30.5	50.2	32.0	16.6	34.1	41.1
MobileNetV3 [14]	219	75.2	29.9	49.3	30.8	14.9	33.3	41.1
SingPath NAS [26]	365	75.0	30.7	49.8	32.2	15.4	33.9	41.6
SCARLET-A	365	76.9	31.4	51.2	33.0	16.3	35.1	41.8
SCARLET-B	329	76.3	31.2	51.2	32.6	17.0	34.7	41.9

## 6 Ablation Study and Analysis

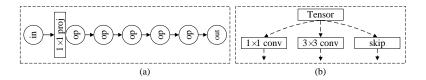
## 6.1 Training Stability

Compared with skip connection, ELS can help stabilize the training process of a scalable supernet, shown in Fig. 3. We believe it is due to boosted cross-block features similarities (increased by 0.3 compared with pure skip connection). Interestingly enough, ELS is also able to close up the feature angle discrepancy. This phenomenon is depicted in Fig. 1. Informally, ELS plays an important role in rectifying the features' phase gap between skip connections and other homogeneous choices. Essentially, ELS is a near-homogeneous to an inverted bottleneck block, while a skip connection is instead heterogeneous. As a result, for

both one-shot approaches, supernets with ELS enjoy higher training accuracies (red solid line) and lower variances (red shaded area). Although there is a small proportion of one-shot models with low accuracy, they can be easily excluded from the proposed constrained optimization process.

## 6.2 Ranking Ability with and without ELS

The most important role of the supernet in the two-stage approach is to evaluate the performance of the subnetworks, so-called 'ranking ability'. To find out the contribution of ELS, we perform experiments on a common NAS benchmark NAS-Bench-101 [33] with some adaptations. Specifically, we construct a supernet to have a stack of 9 cells, each cell has at most 5 sequential internal nodes, each node has 3 optional operations:  $1 \times 1$  Conv,  $3 \times 3$  Conv and a skip connection. The first node is preceded by a 1x1 Conv projection. The designed cell and node choices are shown in Fig. 6.



**Fig. 6.** Ranking analysis is based on a subspace of NAS-Bench-101. (a) A cell is a stack of 5 nodes. An additional  $1 \times 1$  conv projection is added before the first one to avoid channel mismatch. (b) For each node, we can select one operation from 3 choices.

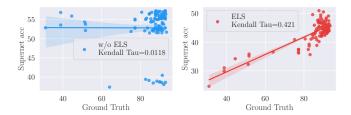


Fig. 7. Comparison of ground-truth vs. estimated accuracy correlation between the supernets trained with and without ELS, based on 100 sampled models from NAS-Bench-101 [33]. ELS substantially boosts the ranking ability of the supernet.

We train such a supernet with and without ELS on CIFAR-10. For both experiments, we train for 100 epochs with a batch size of 96, and a learning rate of 0.025. We randomly sample 100 models to lookup their ground-truth accuracies from NAS-Bench-101. We calculate the ranking ability of each supernet using

Kendall Tau [17], shown in Fig 7. The one with ELS reaches a tau value of 0.421, indicating much higher correlation.

**ELS vs. Skip Connection.** To give a clearer comparison between the skip connection (SC) and the proposed ELS, we illustrate their functionality in Table 6. Both operations are foldable, meaning they are used in supernet training, but later removed (folded) to build the corresponding subnetworks as they both creates an identity transformation before folding and after (see also Fig. 2). The difference is that, ELS is learnable so that it gives more consistent feature maps in each layer. This is crucial to improve the supernet ranking, attested by Fig 7.

Table 6. Comparison of Skip Connections (SC) and ELS as per foldability and ranking.

Operation	Foldable	Identity Mapping	Learnable	Feature Similarity	Ranking
SC	/	<b>✓</b>	X	Low	Poor
ELS	✓	✓	1	High	Good

## 6.3 Equivariant vs. Non-equivariant Stabilizer

The equivalence requirement for the stabilizer (Equation 3) plays a pivotal role in our approach. We evaluate a subnetwork with ELS as it is an identical proxy to the one without it. A stabilizer that violates the equivalence requirement will give wrong evaluation.

For example, we make a simple modification by adding a ReLU function to ELS, this makes the stabilizer non-equivariant because of non-linearity. Can we use a supernet with this stabilizer to correctly evaluate a model? Given a model denoted by choice indices:  $M_{(1,3,1,0,12,0,0,0,12,12,12,12,12,0,0,0,12,12,19)}$ , when we train it respectively with ELS and with ELS-ReLU on the same settings, we find them have different representational power, as shown in Fig. 8. The one with ELS-ReLU overestimates the model compared to the one with ELS-no-ReLU, which reflects its truth by Lemma 1.

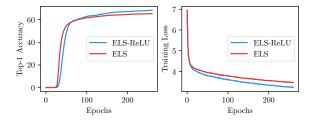


Fig. 8. Adding ReLU to ELS gives a wrong evaluation (overestimation) of a subnetwork.

## 6.4 Constrained Optimization

For multi-objective evolutionary search in scalable search space, to limit the minimum accuracy is more than necessary. In a standard NSGA-II [8] process, models with many skip connections are easily picked for the high-ranking non-dominated sets. For an extreme example, a model consisting of skip connections for all 19 layers has minimum multiply-adds, it will always stay as a boundary node. This brings in *gene contamination* for the evolution process as this poorperforming gene never dies out.

To demonstrate the necessity of a minimum accuracy constraint, we compare the case with  $acc_{min} = 0$  and  $acc_{min} = 0.4$  in Fig. 9. We can observe the number of skip connections has been greatly reduced (red line in the right figure). As a consequence, the evolution converges to a better Pareto Front (red line in the left figure): higher validation accuracies at the same level of multiply-adds.

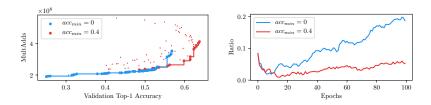


Fig. 9. Ablation study on constrained optimization. Left: Pareto front of MultAdds vs. Accuracy. Right: The ratios of skip connections per epoch.

#### 6.5 Component Analysis

A supernet trained without ELS can't deliver good search results. Using the same searching strategy NSGA-II, its best model found below 380M FLOPS obtains 71.3% top-1 accuracy on ImageNet, while SCARLET-A (365M) has 76.9%. Therefore, the contribution to the final performance comes mainly from ELS in the first stage. The searching strategy heavily depends on ranking ability.

# 7 Conclusion

In this paper, we expose a critical failure in single-path one-shot neural architecture search when scalability is considered. We discover the underlying feature dissimilarity hinders supernet training. The proposed equivariant learnable stabilizer is effective to rectify such discrepancy while maintaining the same representational power for subnetworks. We also employ a weighted multi-objective evolutionary search to find a series of state-of-the-art SCARLET architectures. Good transferability is achieved on various vision tasks. Compared with unnecessarily costly EfficientNet, our method is a step forward towards more efficient and flexible neural architecture search.

## References

- 1. Bender, G., Kindermans, P.J., Zoph, B., Vasudevan, V., Le, Q.: Understanding and Simplifying One-Shot Architecture Search. In: ICML. pp. 549–558 (2018)
- Cai, H., Zhu, L., Han, S.: ProxylessNAS: Direct Neural Architecture Search on Target Task and Hardware. In: ICLR (2019)
- 3. Chen, X., Xie, L., Wu, J., Tian, Q.: Progressive Differentiable Architecture Search: Bridging the Depth Gap between Search and Evaluation. In: ICCV (2019)
- 4. Chollet, F.: Xception: Deep Learning with Depthwise Separable Convolutions. In: CVPR. pp. 1251–1258 (2017)
- Chu, X., Zhang, B., Xu, R.: MoGA: Searching Beyond MobileNetV3. In: ICASSP (2020)
- Chu, X., Zhang, B., Xu, R., Li, J.: FairNAS: Rethinking Evaluation Fairness of Weight Sharing Neural Architecture Search. arXiv preprint. arXiv:1907.01845 (2019)
- 7. Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: AutoAugment: Learning Augmentation Policies from Data. In: CVPR (2019)
- 8. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)
- 9. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR. pp. 248–255. Ieee (2009)
- 10. Dong, X., Yang, Y.: Searching for a Robust Neural Architecture in Four GPU Hours. In: CVPR. pp. 1761–1770 (2019)
- 11. Friedrich, T., Kroeger, T., Neumann, F.: Weighted Preferences in Evolutionary Multi-Objective Optimization. In: AJCAI. pp. 291–300. Springer (2011)
- 12. Guo, Z., Zhang, X., Mu, H., Heng, W., Liu, Z., Wei, Y., Sun, J.: Single Path One-Shot Neural Architecture Search with Uniform Sampling. arXiv preprint. arXiv:1904.00420 (2019)
- 13. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: CVPR. pp. 770–778 (2016)
- 14. 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: ICCV (2019)
- Hu, J., Shen, L., Sun, G.: Squeeze-and-Excitation Networks. In: CVPR. pp. 7132–7141 (2018)
- Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely Connected Convolutional Networks. In: CVPR. pp. 4700–4708 (2017)
- 17. Kendall, M.G.: A New Measure of Rank Correlation. Biometrika  ${\bf 30}(1/2),~81–93~(1938)$
- Kornblith, S., Shlens, J., Le, Q.V.: Do Better Imagenet Models Transfer Better?
   In: CVPR. pp. 2661–2671 (2019)
- Krizhevsky, A., Hinton, G., et al.: Learning Multiple Layers of Features from Tiny Images. Tech. rep., Citeseer (2009)
- Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal Loss for Dense Object Detection. In: ICCV (2017)
- 21. 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: ECCV. pp. 740–755. Springer (2014)
- 22. Liu, H., Simonyan, K., Yang, Y.: DARTS: Differentiable Architecture Search. In: ICLR (2019)

- Pham, H., Guan, M.Y., Zoph, B., Le, Q.V., Dean, J.: Efficient Neural Architecture Search via Parameter Sharing. In: ICML (2018)
- 24. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: MobileNetV2: Inverted Residuals and Linear Bottlenecks. In: CVPR. pp. 4510–4520 (2018)
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A Simple Way to Prevent Neural Networks from. Overfitting. JMLR 15(1), 1929–1958 (2014)
- Stamoulis, D., Ding, R., Wang, D., Lymberopoulos, D., Priyantha, B., Liu, J., Marculescu, D.: Single-Path NAS: Designing Hardware-Efficient ConvNets in less than 4 Hours. In: ECMLPKDD (2019)
- 27. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. In: AAAI (2017)
- 28. Tan, M., Chen, B., Pang, R., Vasudevan, V., Le, Q.V.: Mnasnet: Platform-Aware Neural Architecture Search for Mobile. In: CVPR (2019)
- Tan, M., Le, Q.V.: EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In: ICML (2019)
- 30. Tan, M., Le., Q.V.: MixConv: Mixed Depthwise Convolutional Kernels. BMVC (2019)
- 31. 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. CVPR (2019)
- 32. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated Residual Transformations for Deep Neural Networks. In: CVPR. pp. 1492–1500 (2017)
- Ying, C., Klein, A., Christiansen, E., Real, E., Murphy, K., Hutter, F.: Nas-bench-101: Towards reproducible neural architecture search. In: ICML. pp. 7105–7114 (2019)
- 34. Zela, A., Elsken, T., Saikia, T., Marrakchi, Y., Brox, T., Hutter, F.: Understanding and Robustifying Differentiable Architecture Search. In: ICLR (2020)
- 35. Zhang, X., Zhou, X., Lin, M., Sun, J.: ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. In: CVPR (June 2018)
- 36. Zhang, X., Li, Z., Change Loy, C., Lin, D.: PolyNet: A Pursuit of Structural Diversity in Very Deep Networks. In: CVPR. pp. 718–726 (2017)
- 37. Zoph, B., Vasudevan, V., Shlens, J., Le, Q.V.: Learning Transferable Architectures for Scalable Image Recognition. In: CVPR. pp. 8697–8710 (2018)

## A Proof

Proof. First, we prove that Equation 3 (main text) holds for  $\forall o \in \{0,1,...,n-2\}$ . In this case, it's sufficient to prove the output of the first convolution  $Conv_{(c_l,m,k,k)}$  can be exactly matched by adding  $Conv_{(c_l,c_{l+1},1,1)}$  before  $Conv_{(c_l+1,m,k,k)}$ . Let  $W^1_{c_l,c_{l+1},1,1}$  and  $W^2_{c_l,m,k,k}$  be the weight tensors of  $Conv_{(c_l,c_{l+1},1,1)}$  and  $Conv_{(c_{l+1},m,k,1)}$  respectively. Let  $W^3_{c_l,m,k,k}$  be the weight tensors of  $Conv_{(c_l,m,k,1)}$ . Let w be one element of the tensor. We have

$$y = Conv_{(c_l, c_{l+1}, 1, 1)}(x_l^{c_l}), z = Conv_{(c_{l+1}, m, k, 1)}(y)$$
(5)

$$y(i,j,c) = \sum_{p=1}^{c_l} w_{p,c,1,1}^1 x(i,j,p)$$
 (6)

Also,

$$z(i,j,c) = \sum_{q=1}^{k} \sum_{p=1}^{c_{l+1}} w_{p,c,q,q}^{2} y(i+q,j+q,p)$$

$$= \sum_{q=1}^{k} \sum_{p=1}^{c_{l+1}} w_{p,c,q,q}^{2} (\sum_{u=1}^{c_{l}} w_{u,p,1,1}^{1} x(i+q,j+q,u))$$

$$= \sum_{q=1}^{k} \sum_{p=1}^{c_{l+1}} \sum_{u=1}^{c_{l}} w_{p,c,q,q}^{2} w_{u,p,1,1}^{1} x(i+q,j+q,u)$$

$$= \sum_{q=1}^{k} \sum_{v=1}^{c_{l}} w_{u,c,q,q}^{3} x(i+q,j+q,u)$$

$$(7)$$

Therefore, the first part is proved by setting

$$w_{u,c,q,q}^3 = \sum_{p=1}^{c_{l+1}} w_{p,c,q,q}^2 w_{u,p,1,1}^1.$$
 (8)

For o = n - 1, we replace a skip connection with an ELS. We can iteratively apply the first part of the proof till the end of searchable layers.

# B Algorithm

Our constrained and weighted NAS pipeline is listed in Algorithm 1 and Fig. 10.

# C Experiments

## C.1 Search Space

For later experiments, we add skip connections to commonly used search space to construct  $S_1$  and  $S_2$ . They are described as follows,

**Algorithm 1** The constrained and weighted NAS pipeline.

```
Input: Supernet S, the number of generations N, population size n, validation
dataset D, constraints C, objective weights w
Output: A set of K individuals on the Pareto front.
Train supernet S defined on the scalable search space.
Uniformly generate the populations P_0 and Q_0 until each has n individuals satisfying
C_{\text{FLOPS}}, C_{\text{Accuracy}}.
for i = 0 to N - 1 do
   R_i = P_i \cup Q_i
   F = \text{non-dominated-sorting}(R_i)
  Pick n individuals to form P_{i+1} by ranks and the crowding distance weighted by
  Q_{i+1} = \emptyset
   while size(Q_{i+1}) < n do
     M = \text{tournament-selection}(P_{i+1})
     q_{i+1} = \operatorname{crossover}(M) \cup \operatorname{hierarchical-mutation}(M) {Check the FLOPS constraint
     at first (It takes < 1ms).
     if FLOPS(q_{i+1}) > FLOPS_{max} then
        continue
     end if
     Evaluate model q_{i+1} with S on D {Check the accuracy constraint (It takes
     \approx 60s).
     if Accuracy(q_{i+1}) > Acc_{min} then
        Add q_{i+1} to Q_{i+1}
     end if
  end while
end for
Select K equispaced models near Pareto-front from P_N
```

**Search Space**  $S_1$ . It is similar to ProxylessNAS [2], where MobileNetV2 [24] is adopted as its backbone. In particular,  $S_1$  is represented as a block-level supernet with L=19 layers of N=7 choices each. Its total size is  $7^{19}$ . The choices are,

- MobileNetV2's inverted bottleneck blocks [24] of two expansion rates (x) in (3,6), three kernel sizes (y) in (3,5,7), labelled as MBExKy<sup>6</sup>,
- skip connection (the 6th choice<sup>7</sup>).

**Search Space**  $S_2$ . On top of  $S_1$ , we give each inverted bottleneck a squeeze-and-excitation [15] option (e.g., ExKy, ExKy\_SE), similar to MnasNet [28]. Its total size thus becomes  $13^{19}$ .

We have to notice that skip connections are commonly used [28,22,1], but meticulously neglected in recent single-path one-shot methods [12,6].

<sup>&</sup>lt;sup>6</sup> The order of numbering o = (x-3) + (y-3)/2.

<sup>&</sup>lt;sup>7</sup> zero-based numbering

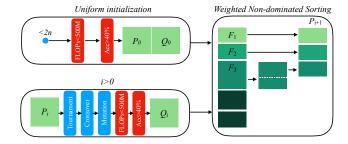


Fig. 10. Constrained and weighted NSGA-II Pipeline. It starts with a uniform initialization (top left) with some constraints (red) to generate the initial population. The trained scalable supernet serves as a fast evaluator to decide the relative performance of each model so that they can be grouped into several Fronts  $(F_1, F_2, ...)$  by weighted non-dominated sorting (right). Only the top n of them make up the next generation  $P_{i+1}$ , based on which  $Q_{i+1}$  is produced with tournament selection, crossover and mutation (blue) under the same constraints (bottom left). The whole evolution loops until we reach Pareto-optimality.

## C.2 NSGA-II Hyperparameters

The hyperparameters for the weighted NSGA-II approach are given in Table 7.

Table 7. Hyperparameters for the weighted NSGA-II approach.

Item	value	Item	value
Population N	70	Mutation Ratio	0.8
$p_{rm}$	0.2	$p_{re}$	0.65
$p_{pr}$	0.15	$p_M$	0.7
$p_{K-M}$	0.3		

## C.3 More Details about Scalable Supernet with ELS

Given an input of a chickadee<sup>8</sup> image from ImageNet, we illustrate both high-level and low-level feature maps of the trained supernet with our proposed improvements in Figure 11. Pure skip connection easily interferes with the training process as it causes incongruence with other choice blocks. Note the channel size of feature map after Choice 6 in Figure 11 (a) is half of others because the previous channel size is 16, while other choice blocks output 32 channels. This effect is largely attenuated by ELS. As it goes deeper, we still observe consistent high-level features. Specifically, when ELS is not enforced, high-level features

<sup>&</sup>lt;sup>8</sup> ImageNet ID: n01592084\_7680

of deeper channels easily get blurred out, while the supernet with ELS enabled continues to learn useful features in deeper channels.

## C.4 Search Space Evaluation

NAS results can benefit from good search space. To prove the validity of the proposed method, we show our search space has a wide range and is not particularly designed. We pick two extreme cases, one with all identity blocks (only the stem and the tail remains), the other with all K7E6s. They have the minimum and the maximum FLOPS respectively. We list their evaluation result in Table 8. The former has 24.1% top-1 accuracy on ImageNet, and the latter 76.8% at a cost of 557M FLOPs. Both are infeasible solutions as they violate either  $acc_{min}$  or  $madds_{max}$ . It's thus a challenging task to deal with such search space for ordinary search techniques.

Table 8. Full train results of models with minimal and maximal FLOPS.

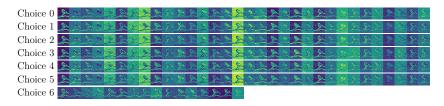
Models	FLOPS (M)	$> madds_{max}$	Top-1 (%)	Top-5 (%)	$< acc_{min}$
All Identity	23	No	24.1	45.0	Yes
All K7E6	557	Yes	76.8	93.3	No

## C.5 Analysis of SCARLET Models

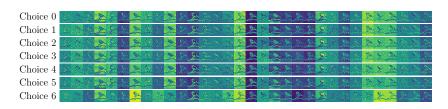
SCARLET-A makes full use of large kernels (five  $5 \times 5$  and seven  $7 \times 7$  kernels) to enlarge receptive field. Besides it activates many squeezing and excitation (12 out of 19) blocks to improve its classification performance. At the early stage, it appreciates either large kernels and small expansion ratios or small kernels and large expansion ratios to balance the trade-off between accuracy and FLOPs.

SCARLET-B chooses two identity operations. Compared with A, it shortens network depth at the last stages. Besides, it utilizes squeezing and excitation block extensively (14 out of 17). It places a large expansion block with large kernels at the tail stage.

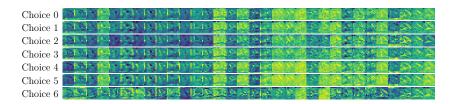
SCARLET-C uses three identity operations and utilizes small expansion ratio extensively to cut down the FLOPs, large expansion ratio at the tail stage whose resolution is  $7 \times 7$ . It prefers large kernels before the downsampling layers. Besides, it makes an extensive use of squeeze and excitation to boost accuracy.



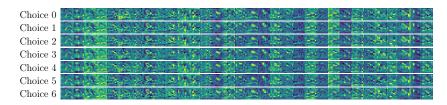
(a) First choice blocks' feature maps without ELS



(b) First choice blocks' feature maps with ELS



(c) High-level choice blocks' feature maps without ELS



(d) High-level choice blocks' feature maps with ELS

 ${f Fig.\,11.}$  Learned low-level and high-level features for the supernet with and without ELS.