Fair DARTS: Eliminating Unfair Advantages in Differentiable Architecture Search

Xiangxiang Chu[1[0000−0003−2548−0605]], Tianbao Zhou[2[0000−0002−2133−059X]⋆], Bo Zhang[1[0000−0003−0564−617X]⋆], and Jixiang Li[1[0000−0001−5949−1498]]

[1] Xiaomi AI Lab
{chuxiangxiang,zhangbo11,lijixiang}@xiaomi.com
[2] Minzu University of China
tianbaochou@163.com

Abstract. Differentiable Architecture Search (DARTS) is now a widely disseminated weight-sharing neural architecture search method. However, it suffers from well-known performance collapse due to an inevitable aggregation of skip connections. In this paper, we first disclose that its root cause lies in an **unfair advantage** in **exclusive competition**. Through experiments, we show that if either of two conditions is broken, the collapse disappears. Thereby, we present a novel approach called Fair DARTS where the exclusive competition is relaxed to be collaborative. Specifically, we let each operation's architectural weight be independent of others. Yet there is still an important issue of discretization discrepancy. We then propose a **zero-one** loss to push architectural weights towards zero or one, which approximates an expected multi-hot solution. Our experiments are performed on two mainstream search spaces, and we derive new state-of-the-art results on CIFAR-10 and ImageNet[3].

Keywords: Differentiable Neural Architecture Search · Image Classification · Failure of DARTS

1 Introduction

In the wake of the DARTS's open-sourcing [19], a diverse number of its variants emerge in the *neural architecture search* community. Some of them extend its use in higher-level architecture search spaces with performance awareness in mind [3, 31], some learn a stochastic distribution instead of architectural parameters [31, 32, 35, 9, 10], and others offer remedies on discovering its lack of robustness [22, 4, 18, 15, 34].

In spite of these endeavors, the aggregation of skip connections in DARTS that noticed by [4, 18, 1, 34] has not been solved with perfection. Observing that the aggregation leads to a dramatic **performance collapse** for the resulting architecture, P-DARTS [4] utilizes dropout as a workaround to restrict the number of skip connections during optimization. DARTS+ [18] directly puts a hard

⋆ Equal Contribution.
[3] Code is available here: https://github.com/xiaomi-automl/FairDARTS

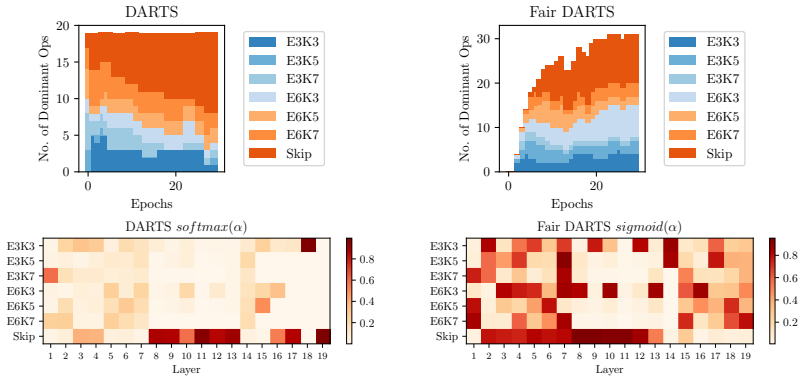

Fig. 1. Top: Stacked area plot of the number of dominant operations[5] of DARTS and Fair DARTS when searching on ImageNet in search space S_2 (19 searchable layers). **Bottom:** Heatmaps of softmax (DARTS) and sigmoid (Fair DARTS) values in the last searching epoch. DARTS finally chooses a shallow model (11 layers removed by activating skip connections only) which obtains 66.4% top-1 accuracy. While in Fair DARTS, all operations develop independently that an excessive number of skip connections no longer leads to poor performance. Here it infers a deeper model (only one layer is removed) with 75.6% top-1 accuracy

limit of two skip-connections per cell. RobustDARTS [34] finds out that these solutions coincide with high validation loss curvatures. To some extent, these approaches consider the poor-performing models as impurities from the solution set, for which they intervene in the training process to filter them out.

On the contrary, we extend the solution set and revise the optimization process so that aggregation of skip connections no longer causes the collapse. Moreover, there remains a **discrepancy problem** when discretizing continuous architecture encodings. DARTS [19] leaves it as future work, but till now it has not been deeply studied. We reiterate the basic premise of DARTS is that the continuous solution approximates a one-hot encoding. Intuitively, the smaller discrepancies are, the more consistent it will be when we transform a continuous solution back to a discrete one. We summarize our contributions as follows:

Firstly, we disclose the root cause that leads to the collapse of DARTS, which we later define as an *unfair advantage* that drives skip connections into a monopoly state in *exclusive competition*. These two indispensable factors work together to induce a performance collapse. Moreover, if either of the two conditions is broken, the collapse disappears.

Secondly, we propose the first **collaborative competition** approach by offering each operation an independent architectural weight. The unfair advantage no longer prevails as we break the second factor. Furthermore, to address the discrepancy between the continuous architecture encoding and the derived

[5] In DARTS, it refers to the one with the highest architectural weight. In FairDARTS, it means the one whose $\sigma > \sigma_{threshold}$. Here we use $\sigma_{threshold} = 0.75$.

discrete one in our method, we propose a novel auxiliary loss, called *zero-one loss*, to steer architectural weights towards their extremities, that is, either completely enabled or disabled. The discrepancy thus decreases to its minimum.

Thirdly, based on the root cause of the collapse, we provide a unified perspective to view current DARTS cures for skip connections' aggregation. The majority of these works either make use of dropout [27] on skip connections [4, 34], or play with the later termed *boundary epoch* by different early-stopping strategies [18, 34]. They can all be regarded as preventing the first factor from taking effect. Moreover, as a direct application, we can derive a hypothesis that adding Gaussian noise also disrupts the unfairness, which is later proved to be effective.

Lastly, we conduct thorough experiments in two widely used search spaces in both proxy and proxyless ways. Results show that our method can escape from performance collapse. We also achieve state-of-the-art networks on CIFAR-10 and ImageNet.

2 Related Work

Lately, neural architecture search [36] has grown as a well-formed methodology to discover networks for various deep learning tasks. Endeavors have been made to reduce the enormous searching overhead with the weight-sharing mechanism [2, 23, 19]. Especially in DARTS [19], a nested gradient-descent algorithm is exploited to search for the graphical representation of architectures, which is born from gradient-based hyperparameter optimization [20].

Due to the limit of the DARTS search space, ProxylessNAS [3] and FBNet [31] apply DARTS in much larger search spaces based on MobileNetV2 [26]. ProxylessNAS also differs from DARTS in its supernet training process, where only two paths are activated, based on the assumption that one path is the best amongst all should be better than any single one. From a fairness point of view, as only two paths enhance their ability (get parameters updated) while others remain unchanged, it implicitly creates a bias. FBNet [31], SNAS [32] and GDAS [10] utilize the differentiable Gumbel Softmax [21, 14] to mimic one-hot encoding. However, the one-hot nature implies an exclusive competition, which risks being exploited by unfair advantages.

Superficially, the most relevant work to ours is RobustDARTS [34]. Under several simplified search spaces, they state that the found solutions generalize poorly when they coincide with high validation loss curvature, where the supernet with an excessive number of skip connections happens to be such a solution. Based on this observation, they impose early-stop regularization by tracking the largest eigenvalue. Instead, our method doesn't need to perform early stopping.

3 The Downside of DARTS

In this section, we aim to excavate the disadvantages of DARTS that possibly impede the searching performance. We first prepare a minimum background.

3.1 Preliminary of Differentiable Architecture Search

For the case of convolutional neural networks, DARTS [19] searches for a *normal cell* and a *reduction cell* to build up the final architecture. A cell is represented as a directed acyclic graph (DAG) of N nodes in sequential order. Each node stands for a feature map. The edge $e_{i,j}$ from node i to j operates on the input feature x_i and its output is denoted as $o_{i,j}(x_i)$. The intermediate node j gathers all inputs from the incoming edges,

$$x_j = \sum_{i<j} o_{i,j}(x_i). \tag{1}$$

Let $\mathcal{O} = \{o^1_{i,j}, o^2_{i,j}, ..., o^M_{i,j}\}$ be the set of M candidate operations on edge $e_{i,j}$. DARTS relaxes this categorical choice to a softmax over all operations in $\mathcal{O}$ to form a mixed output:

$$\bar{o}_{i,j}(x) = \sum_{o\in\mathcal{O}} \frac{\exp(\alpha_{o_{i,j}})}{\sum_{o'\in\mathcal{O}} \exp(\alpha_{o'_{i,j}})} o(x), \tag{2}$$

where each operation $o_{i,j}$ is associated with a continuous coefficient $\alpha_{o_{i,j}}$. Regarding edge $e_{i,j}$, this softmax is utilized to approximate one-hot encoding $\beta_{i,j} = (\beta_{o^1_{i,j}}, \beta_{o^2_{i,j}}, ..., \beta_{o^M_{i,j}})$. Formally, let $\alpha_{o_{i,j}}$ denote the architectural weights vector ($\alpha_{o^1_{i,j}}$, $\alpha_{o^2_{i,j}}$, ..., $\alpha_{o^M_{i,j}}$). DARTS thus assumes the following as a valid approximation,

$$softmax(\alpha_{o_{i,j}}) \approx \beta_{i,j}. \tag{3}$$

The architecture search problem is reduced to learning α^* and network weights w^* that minimize the validation loss $\mathcal{L}_{val}(w^*, \alpha^*)$. DARTS resolves this problem with a bi-level optimization,

$$\begin{aligned} &\min_{\alpha} \mathcal{L}_{val}(w^*(\alpha), \alpha) \\ &\text{s.t. } w^*(\alpha) = \text{argmin}_w \mathcal{L}_{train}(w, \alpha). \end{aligned} \tag{4}$$

We also adopt two common search spaces, the DARTS [19] search space (S_1) and the ProxylessNAS [3] search space (S_2) with minor modifications. More details are given in Section 2 (supplementary).

In S_2, the output of the l-th layer is a softmax-weighted summation of N choices. Formally, it can be written as

$$x_l = \sum_{k=1}^{N} \frac{\exp(\alpha^k_{l-1,l})}{\sum_{j=1}^{N} \exp(\alpha^j_{l-1,l})} o^k_{l-1,l}(x_{l-1}). \tag{5}$$

3.2 Performance Collapse Caused by Intractable Skip Connections

DARTS suffers from significant performance decay when *skip connections* become dominant [4, 18]. It was described as a competition-and-cooperation issue

in the bi-level optimization [18]. Still, the reason behind this behavior is not clear, we hereby provide a different perspective.

First, to confirm this issue, we run DARTS $k = 4$ times with different random seeds. Following DARTS, we select 8 top-performing operations per cell (2 each for 4 intermediate nodes). Here we say one operation is *dominant* if it has top-2 $softmax(\alpha)$ among all incoming edges' candidates of a certain node. The results are shown in Fig. 2. In the beginning, all operations are given the same opportunity. As the over-parameterized network gradually converges, there is an evident aggregation of skip connections after 20 epochs (5 out of 8 in an extreme case).

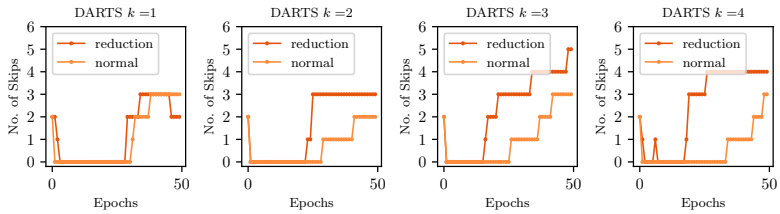

Fig. 2. The number of dominant skip connections continues to grow when searching with DARTS (run $k = 4$ times) on CIFAR-10 (in S_1)

When we utilize DARTS directly on ImageNet in S_2, which is a single branch architecture, the same phenomenon rigorously reappears. The number of dominant skip-connections (highest $softmax(\alpha)$ among all operations in that layer) steadily increases and reaches 11 out of 19 layers in the end, which is shown on the left of Fig. 1.

But why is this happening? The underlying reasons are rarely discussed in depth. A brief and superficial analysis regarding information flow is given in [4]. However, we claim that the reason for excessive skip connections is from **exclusive competition** among various operations. In Equation 2 and Equation 5, the skip connection is softmax-weighted and added to the output, which resembles a basic residual module as in ResNet [11]. While this module greatly benefits the training, the architectural weight of a skip connection increases much faster than its competitors. Moreover, *the softmax operation inherently provides an exclusive competition since increasing one is at the cost of suppressing others*. As a result, skip connections become gradually dominant during optimization. We have to keep in mind that skip connection works well because it is in cooperation with convolutions [11]. However, DARTS picks the top-performing one (skip connection here) and discards its collaborator (convolution), which results in a degenerate model.

We further study this effect from the experiments on CIFAR-10 by recording the competition progress in Fig. 3. The derived model has 8 skip connections in

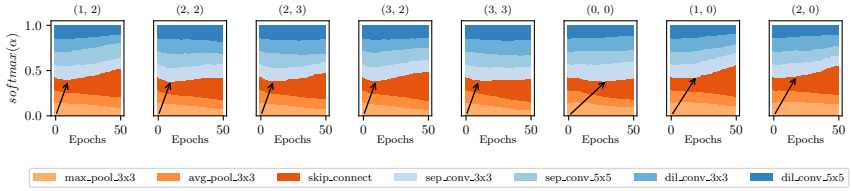

Fig. 3. The softmax evolution where skip connections gradually become dominant when running DARTS on CIFAR-10 (in S_1). Last two subplots of edge (1,0) and (2,0) are from the normal cell, the rest are from the reduction cell. Black arrows point to boundary epochs where skip connections start to demonstrate its strength

total[6]. ResNet [11] discovers that *skip connections begin to demonstrate power after a few epochs compared with models without them.* Interestingly, a similar phenomenon is also observed in our experiments. We term this tipping point a *boundary epoch*. The boundary epochs may vary from edge to edge, but are generally at the early stage. From Fig. 3, we observe that skip connections colored in red-orange progressively obtain higher architectural weights after some certain boundary epochs. Meantime, other operations are suppressed and steadily decline. We consider this benefit from the residual module as an unfair advantage by Definition 1.

Definition 1. ***Unfair Advantage.*** *Suppose that choosing one operation among others is a competition. This competition is deemed* ***exclusive*** *when only restricted operations can be selected. An operation in an exclusive competition is said to have an* ***unfair advantage*** *if this advantage contributes more to competition than to the performance of a resulted network.*

From the above discussion, we can draw **Insight 1: The root cause of excessive skip connections is the inherent unfair competition.** The skip connection has an unfair advantage by forming a residual module which is convenient for the supernet training, but not equally beneficial for the performance of the outcome network where the residual module is broken.

3.3 Non-negligible Discrepancy of Discretization

Apart from the above issue, DARTS reports that it suffers from discrepancies when discretizing continuous encodings [19]. To verify the problem, we run DARTS in S_1 on CIFAR-10, and in S_2 on ImageNet. The values of $softmax(\alpha)$ of the last iteration are displayed in Fig. 4 (S_1) and on the bottom left of Fig. 1 (S_2). For S_1, the largest value is about 0.3 while the smallest one is above 0.1[7]. This range is somewhat too narrow to differentiate 'good' operations from 'bad'. For instance on edge 2 of the reduction cell, the values are very close to each

[6] corresponding to the experiment ($k = 3$) in Fig. 2.
[7] We run DARTS 4 times and it holds every time.

other, [0.174, 0.170, 0.176, 0.112, 0.116, 0.132, 0.118], it's hard to say that an operation weighted by 0.176 is better than the other by 0.174. For S_2, the top-1 values are not so evidently particular from layer 2 to 7. Take the second layer for example, we have to use [0.235, 0.057, 0.17, 0.016, 0.187, 0.269, 0.066] to approximate [0, 0, 0, 0, 0, 1, 0]. This again confirms the existence of discrepancy.

In summary, DARTS is usually far from a good resemblance to a one-hot representation as required by its premise in Equation 3. We often have to make ambiguous choices without high confidence. Hence, we learn **Insight 2: Relaxing from discrete categorical choices to continuous ones should make a close approximation**.

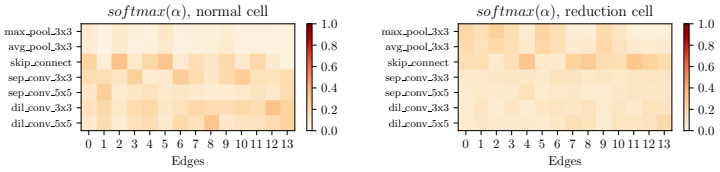

Fig. 4. Heatmap of softmax values in the normal cell and the reduction cell at the last searching epoch when running DARTS on CIFAR-10 (in search space S_1)

4 Fair DARTS

4.1 Stepping out the Pitfalls of Skip Connections

Based on **Insight 1**, we propose a *cooperative mechanism* to eliminate the existing unfair advantage. Not only should we exploit skip connection for smoother information flow, but we also have to provide equal opportunities for other operations. In a word, they need to avoid being trapped by unfair advantage from skip connections. On this regard, we apply a *sigmoid activation* (σ) for each $\alpha_{i,j}$, so that each operation can be switched on or off independently without being suppressed. Formally, we replace Equation 2 with the following,

$$\bar{o}_{i,j}(x) = \sum_{o \in \mathcal{O}} \sigma(\alpha_{o_{i,j}}) o(x). \tag{6}$$

It's trivial to show that even if $\sigma(\alpha_{skip})$ saturates to 1, other operations still can be optimized cooperatively. Promising operations continue to grow their architectural weights to reduce $\mathcal{L}_{val}$, which leads to a **multi-hot** approximation. Instead, DARTS attempts to derive a one-hot estimation. The difference is that we have extended the solution set. Consequently, it allows us to tackle the discretization discrepancy. We are left to find out how to drive $\sigma(\alpha)$ towards each extremity (0 or 1). Next, we discuss it in greater detail.

4.2 Resolve Discrepancy from Continuous Representation to Discrete Encoding

To abide by **Insight 2**, we explicitly coerce an extra loss called *zero-one loss* to push the sigmoid value of architectural weights towards 0 or 1. Let $L_{0-1} = f(z)$ denote this loss component, where $z = \sigma(\alpha)$. To achieve our goal, the loss design must meet three basic criteria, a) It needs to have a global maximum at $z = 0.5$ (a fair starting point) and a global minimum at 0 and 1. b) The gradient magnitude $\frac{df}{dz}|_{z\approx 0.5}$ has to be adequately small to allow architectural weights to fluctuate, but large enough to attract z towards 0 or 1 when they are a bit far from 0.5. c) It should be differentiable for backpropagation.

According to the first requirement, we move $\sigma(\alpha)$ away from 0.5 towards 0 or 1 to minimize the discretization gap. The second one enacts explicit necessary constraints. Particularly, small gradients around the peak avoid stepping easily into two ends. Larger gradients around 0 and 1 instead help to quickly capture z nearby. Quite straightforward, we come up with a loss function to meet the above requirements, formally as,

$$L_{0-1} = -\frac{1}{N}\sum_{i}^{N}(\sigma(\alpha_i) - 0.5)^2 \tag{7}$$

In order to control its strength, we weight this loss by a coefficient w_{0-1}, thus the total loss for α is formulated as,

$$L_{total} = \mathcal{L}_{val}(w^*(\alpha), \alpha) + w_{0-1}L_{0-1}. \tag{8}$$

Like DARTS [19], the architectural weights can be optimized through back-propagation. From Equation 8, the search objective is to find an architecture of high accuracy with a good approximation from a continuous encoding to a discrete one.

Moreover, the second requirement is indispensable, otherwise the gradient-based approach may step into local minimum too early. Here we design another loss as a negative example. Let L'_{0-1} be the following,

$$L'_{0-1} = -\frac{1}{N}\sum_{i}^{N}|(\sigma(\alpha_i) - 0.5)|. \tag{9}$$

It's trivial to see that $\frac{d|z-0.5|}{dz}|_{z>0.5} = 1$ and $\frac{d|z-0.5|}{dz}|_{z<0.5} = -1$. Once z stays away from 0.5, it may receive the same gradient (1 or -1) in the later iterations, thus rapidly pushing the architectural weights towards two ends. This phenomenon is illustrated in Fig. 5.

To conclude, by combining Equation 4, 6 and 8, our method which we call Fair DARTS, can be now formally written as

$$\begin{aligned} &\min_{\alpha} \mathcal{L}_{val}(w^*(\alpha), \alpha) + w_{0-1}L_{0-1} \\ &\text{s.t. } w^*(\alpha) = \text{argmin}_w \mathcal{L}_{train}(w, \alpha). \\ &\bar{o}_{i,j}(x) = \sum_{o\in\mathcal{O}} \sigma(\alpha_{o_{i,j}})o(x). \end{aligned} \tag{10}$$

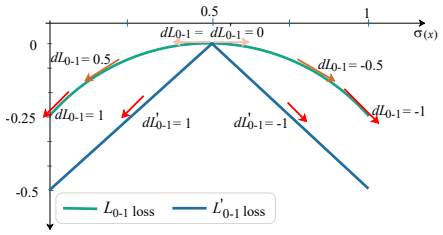

Fig. 5. Illustration about the auxiliary loss design: L_{0-1} (proposed) and L'_{0-1} (control)

It is also important to recognize that our zero-one loss is specially designed for Fair DARTS. Pushing $\sigma(\alpha)$ of one edge towards 0 or 1 is independent of others. It cannot be directly applied to DARTS given the exclusive competition by softmax. As the architectural weights converge to their extremities, it's natural to use a threshold value $\sigma_{threshold}$ in our approach to infer submodels instead of *argmax* .

5 Experiments and Results

5.1 Searching Architectures for CIFAR-10

At the search stage, we use similar hyperparameters and tricks as [19]. We apply the *first-order* optimization and it takes 10 GPU hours. All experiments are done on a Tesla V100. We select our target models with $\sigma_{threshold} = 0.85$[8]. We use the same data processing and training trick as [19, 4].

Our collaborative approach performs well with skip connections aggregation. To verify this, we repetitively search 7 times on different random seeds and report the number of skip connections in Fig. 7 (see supplementary). Since the number of skip connections is more reasonable, we obtain an average top-1 accuracy 97.46%. Especially, the smallest FairDARTS-a reaches 97.46% accuracy on CIFAR-10 with reduced parameters and multiply-adds. A complete result of FairDARTS searched cells are shown in the supplementary (Fig. 5, Fig. 6 and Table 5).

5.2 Transferring to ImageNet

As a common practice, we transfer two searched cells (FairDARTS-a and b[9]) to ImageNet. We keep the same configurations and use the identical training tricks as DARTS [19]. Compared with SNAS [32] and DARTS, FairDARTS-A only uses 3.6M number of parameters and 417M multiply-adds to obtain 73.7% top-1 accuracy on ImageNet validation set. FairDARTS-B also achieves state-of-the-art 75.1% in S_1 with a smaller number of parameters than comparable counterparts.

[8] The maximum number of edges for a node is also limited to 2 as in DARTS.
[9] Their architectures are given in Fig. 5 and 6 (supplementary).

Table 1. Comparison of architectures on CIFAR-10. †: MultAdds computed using the genotypes provided by the authors. ⋆: Averaged on training the best model for several times . ‡: Averaged on models from 7 runs of FairDARTS (Search + Full Train)

Models	Params (M)	×+ (M)	Top-1 (%)	Type
NASNet-A [36]	3.3	608†	97.35	RL
ENAS [23]	4.6	626†	97.11	RL
MdeNAS[35]	3.6	599†	97.45	MDL
DARTS(second order)⋆[19]	3.3	528†	97.24±0.09	GD
SNAS⋆ [32]	2.8	422†	97.15±0.02	GD
GDAS [10]	3.37	519†	97.07	GD
SGAS (Cri.2 avg.) [16]	3.9±0.22†	640±39†	97.33±0.21	GD
P-DARTS [4]	3.4	532†	97.5	GD
PC-DARTS [33]	3.6	558†	97.43	GD
RDARTS [34]	-	-	97.05	GD
FairDARTS-a	**2.8**	**373**	97.46	GD
FairDARTS‡	3.32±0.46	458±61	97.46±0.05	GD

Table 2. Comparison of architectures on ImageNet. ⋆: Based on its published code. †: Searched on CIFAR-10. ††: Searched on CIFAR-100. ‡: Searched on ImageNet (cost more than those transferred). •: in GPU days. ◇: w/ SE and Swish

Models	×+ (M)	Params (M)	Top-1 (%)	Top-5 (%)	Cost•
MobileNetV2(1.4) [26]	585	6.9	74.7	92.2	-
NASNet-A [36]	564	5.3	74.0	91.6	2000
AmoebaNet-A[25]	555	5.1	74.5	92.0	3150
MnasNet-92 [28]	388	3.9	74.79	92.1	1667
DARTS [19]	574	4.7	73.3	91.3	4
FBNet-C [31]	375	5.5	74.9	92.3	9
Proxyless GPU‡ [3]	465*	7.1	75.1	92.4	8.3
FairNAS-C‡ [7]	321	4.4	74.7	92.1	10
SNAS [32]	522	4.3	72.7	90.8	1.5
GDAS [10]	581	5.3	74.0	91.5	0.2
P-DARTS††[4]	577	5.1	74.9*	92.3*	0.3
PC-DARTS† [33]	586	5.3	74.9	92.2	3.8
FairDARTS-A†	417	3.6	73.7	91.7	0.4
FairDARTS-B†	541	4.8	75.1	92.5	0.4
FairDARTS-C‡	380	4.2	75.1	92.4	3
FairDARTS-D‡	440	4.3	**75.6**	**92.6**	3
MobileNetV3 [12]	219	5.4	75.2	92.2	-
MoGA-A [6]	304	5.1	75.9	92.8	12
MixNet-M [30]	360	5.0	77.0	93.3	-
EfficientNet B0 [29]	390	5.3	76.3	93.2	-
SCARLET-A [5]	365	6.7	76.9	93.4	10
FairDARTS-C◇	386	5.3	**77.2**	**93.5**	3

5.3 Searching Proxylessly on ImageNet

Relaxing exclusive competition to collaboration greatly extends the size of the search space. In ProxylessNAS [3], there are 19 searchable layers and each layer contains 7 choices, consisting of 7^{19} possible models. In our approach, every choice can be activated independently, thus, S_2 contains $(2^7)^{19} = 128^{19}$ possible models. To our knowledge, this is a most gigantic search space ever proposed, about 18^{19} times that of [3].

For this search phase, we train for 30 epochs with a batch size of 1024, which takes about 3 GPU days. The final architectural weight matrix (after sigmoid activation) on the bottom right of Fig. 1 is used to derive target models. Under this cooperative setting, the skip connections and other inverted bottleneck blocks can be both chosen to work together, where the former facilitates the training and the latter learn the residual information[17]. In contrast, under the competitive setting of DARTS, it's impossible to achieve this, as shown in the bottom left of Fig. 1. Within 19 layers have 11 skip connection operation is preferred, which cuts down the overall depth of searchable layers to 8.

To be fair, we select at most two choices per layer if there are more than two above $\sigma_{threshold}$ (0.75) and use the same training tricks as [28]. We exclude squeeze and excitation [13] and refrain from using AutoAugment [8] tricks though they can boost the classification accuracy further. The searched model FairDARTS-D is shown in Fig. 6, which places the summation of two inverted bottleneck blocks nearby the down-sampling stage to keep more information. It also utilizes large kernels and big expansion blocks at the tail end. Further, We raise the $\sigma_{threshold}$ as 0.8 to get a more lightweight model FairDARTS-C. FairDARTS-C achieves 75.1% top-1 accuracy using only 4.2 M number of parameters. To make comparisons with EfficientNetB0 [29], MobileNetV3 [12] and MixNet [30], FairDARTS-C obtains 77.2% top-1 accuracy with the same tricks such as squeeze-and-excitation [13], AutoAugment [8] and Swish [24].

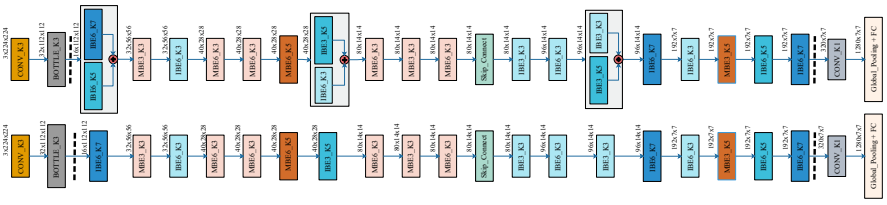

Fig. 6. The Architecture of Fair DARTS-D (top) and C (bottom). IBEx_Ky refers to an inverted bottleneck without an inset skip connection, while MBEx_Ky is the one with it. BOTTLE_K3 is the inverted bottleneck without expansion

6 Ablation Study and Analysis

6.1 Removing Skip Connections from S_1

As unfair advantages are mainly from skip connections, if we remove them from S_1 and get the reduced search space $S_1 \setminus \{skip\}$, we should expect a fair play even in an exclusive competition. Several runs of this experiment also show that there is indeed no more prevailing operations that suppress others, including other parameter-less ones like max-pooling and average pooling (Fig. 7). For $S_1 \setminus \{skip\}$, we run all the experiments with 7 different random seeds and we train the searched models from scratch. The best models ($96.88 \pm 0.18\%$) are slightly higher than DARTS ($96.76 \pm 0.32\%$)[10], but lower than FairDARTS ($97.41 \pm 0.14\%$) in S_1. The lowered accuracy indicates that adequate skip connections are indeed beneficial for accuracy.

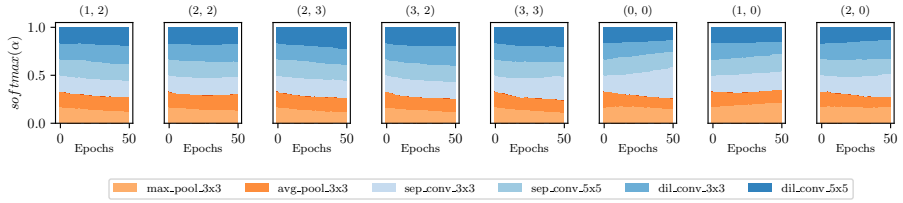

Fig. 7. Stacked area plot of the softmax evolution in $S_1 \setminus \{skip\}$ when running DARTS on CIFAR-10. With unfair advantages removed, all operations enjoy a fair treatment

6.2 How Does Zero-One Loss Matter?

Removing Zero-One Loss. We design two comparison groups for Fair DARTS with and without *zero-one loss*. Other settings are kept the same. We count the distribution of the sigmoid outputs from architectural weights and plot it on the left of Fig. 8. The one without zero-one loss covers a wide range between 0 and 0.6. So we have to make ambiguous choices again. Whereas the proposed loss has narrowed the distribution into two ends around 0 and 1. To further evaluate the influences of removing L_{0-1}, we repeat the searching for 7 times using different random seeds. The averaged top-1 accuracy for these models is 97.33 ± 0.15 (532M FLOPS on average, 74 M more than FairDARTS with L_{0-1}). Therefore, although the unfair advantage is balanced, making ambiguous choices still bring noises to the final search result, which is better solved by L_{0-1}. Discrepancy elimination seems to helps find more light-weight and accurate models.

Zero-One Loss Design. We run two experiments on CIFAR-10, one with L_{0-1} (proposed) and the other L'_{0-1} (control). To some extent, L_{0-1} allows

[10] This differs from DARTS' reported values as it trains one model for several times.

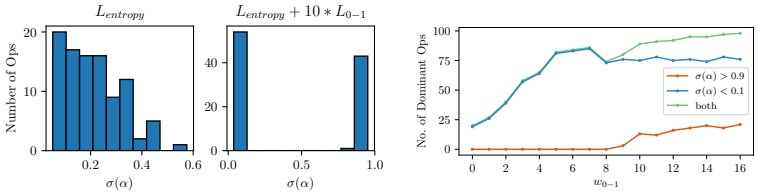

Fig. 8. Left: Histogram of sigmoid values in the last searching epoch without (left) and with L_{0-1} (right). On the right, this auxiliary loss has pushed the values towards 0 or 1. **Right:** Number of dominant operations in the last searching epoch running Fair DARTS on CIFAR-10 w.r.t the sensitivity weight w_{0-1} of auxiliary loss L_{0-1}

stepping out of the local minimum while L'_{0-1} selects operations only at an early stage which depends greatly on the initialization. This matches our analysis in Section 4.2. The detailed results under both loss functions are shown in Fig. 2 (supplementary).

Loss Sensitivity. As the weight w_{0-1} of this auxiliary loss goes higher, it should squeeze more operations towards two ends, but it must not overshadow the main entropy loss. We perform several experiments where an integer w_{0-1} varies within [0, 16]. The right of Fig. 8 shows the final number of dominant operations for each. We select a reasonable $w_{0-1} = 10$ for the best trade-off.

6.3 Discussions From Fairness Perspective

We review the existing methods that seek to avoid the discussed weaknesses. In general, adding dropouts [4, 34] to operations is similar to blending them with a simple additive Gaussian noise, both reduce the performance gain from unfair advantages. Early-stopping [18] avoids the case before unfairness prevails.

Adding dropout to skip connections reduces unfairness. The operation-level dropout [27] inserted after skip connections by P-DARTS [4] can be viewed as an alleviation of unfair advantage. However, it comes with two obvious drawbacks. First, this dropout rate is hard to tune. Second, it is not so effective that they must involve another prior: setting the number of skip connections in the final cell to M. This is a very strong prior for searching good architectures [18].

Adding dropout to all operations also helps. Dropout troubles the training of skip connections and thus weakens the unfair advantage. Reasonably, higher dropout rates are more effective, especially for parameter-free operations. Therefore, RobustDARTS [34] adds dropout to all operations and obtains promising results.

Early stopping matters. DARTS+ [18] explicitly limits the maximum number of skip connections, which can be viewed as an early-stopping strategy nearby the previously mentioned *boundary epoch*, right before too many skip connections rise into power. RobustDARTS [34] also exploits early-stopping when the maximal Hessian eigenvalues change too fast.

Table 3. Experiment 1: Random sampling (7 models each, averaged) from regularized search space ($M = 2$). **Experiment 2:** Adding Gaussian noise to DARTS (repeated 4 times, averaged)

Methods	CIFAR-10 Top-1 Acc (%)
Random (M=2)	97.01 ±0.24
Random (M=2, MultAdds $\geq$ 500M)	97.14 ±0.28
DARTS + Gaussian (cosine decay)	97.12 ±0.23

Limiting the number of skip connections is a strong prior. In the regularized search space of P-DARTS [4] and DARTS+ [18], we find that simply by restricting $M = 2$, it is possible to generate competitive models even *without searching*. We randomly sample models from their search space and report the results in Table 3. In Experiment 1, the second group restricts the multiply-adds to be above 500M, to further leverage the average performance. Surprisingly, both groups outperform DARTS [19].

Random noise can break unfair advantage. Based on our theory, we can boldly postulate that adding a random noise also disrupts the unfair advantage. Therefore, on top of DARTS [19], we mix the skip connections' architectural weights with a standard Gaussian noise $\mathcal{N}(0, 1)$, which has a cosine decay on 50 epochs. The results strongly confirm our hypothesis, as shown in Table 3. We repeat it 4 times to have similar results.

Remove unfair advantages or destroy the exclusive competition? In principle, we can break either one of the indispensable factors to avoid collapse. However, FairDARTS breaks the latter which is simple and effective. Besides, it paves the way to eliminate the discrepancy by scheming an auxiliary loss L_{0-1}. Otherwise, the discrepancy issue remains hard to solve. However, to tackle the discrepancy issue, it's promising that the existing approaches might benefit from tricks like L_{0-1}. This remains to be our future work.

7 Conclusion

We unveil two indispensable factors of the DARTS's aggregation of excessive skip connections: **unfair advantages** and **exclusive competition**. We prove that breaking any one of them can improve the robustness. First, by allowing collaborative competition, each operation develops its architectural weight independently. Meanwhile, the non-negligible discrepancy of discretization is reduced at maximum by coercing a novel auxiliary loss which polarizes the architectural weights. In this regard, we achieve state-of-the-art performance both on CIFAR-10 and ImageNet. Second, disturbing the differentiable process with a Gaussian noise removes unfair advantage which leads to competitive results.

One of our future work is to make it more memory-friendly. As Gumbel softmax is used to replace categorical distribution [31], is there a similar way to our approach? More methods remain to be explored on our basis.

References

1. Bi, K., Hu, C., Xie, L., Chen, X., Wei, L., Tian, Q.: Stabilizing DARTS with Amended Gradient Estimation on Architectural Parameters. arXiv preprint arXiv:1910.11831 (2019)
2. Brock, A., Lim, T., Ritchie, J.M., Weston, N.: SMASH: One-Shot Model Architecture Search through HyperNetworks. In: International Conference on Learning Representations (2018)
3. Cai, H., Zhu, L., Han, S.: ProxylessNAS: Direct Neural Architecture Search on Target Task and Hardware. In: International Conference on Learning Representations (2019)
4. Chen, X., Xie, L., Wu, J., Tian, Q.: Progressive Differentiable Architecture Search: Bridging the Depth Gap between Search and Evaluation. International Conference on Computer Vision (2019)
5. Chu, X., Zhang, B., Li, Q., Xu, R.: SCARLET-NAS: Bridging the gap between scalability and fairness in neural architecture search. arXiv preprint arXiv:1908.06022 (2019)
6. Chu, X., Zhang, B., Xu, R.: MoGA: Searching Beyond MobileNetV3. In: International Conference on Acoustics, Speech, and Signal Processing (2020), https://arxiv.org/pdf/1908.01314.pdf
7. Chu, X., Zhang, B., Xu, R., Li, J.: FairNAS: Rethinking Evaluation Fairness of Weight Sharing Neural Architecture Search. arXiv preprint arXiv:1907.01845 (2019)
8. Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: AutoAugment: Learning Augmentation Policies from Data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
9. Dong, X., Yang, Y.: One-Shot Neural Architecture Search via Self-Evaluated Template Network. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3681–3690 (2019)
10. Dong, X., Yang, Y.: Searching for a Robust Neural Architecture in Four GPU Hours. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1761–1770 (2019)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)
12. 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: International Conference on Computer Vision (2019)
13. Hu, J., Shen, L., Sun, G.: Squeeze-and-Excitation Networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7132–7141 (2018)
14. Jang, E., Gu, S., Poole, B.: Categorical Reparameterization with Gumbel-Softmax. In: International Conference on Learning Representations (2017)
15. Li, G., Zhang, X., Wang, Z., Li, Z., Zhang, T.: StacNAS: Towards stable and consistent optimization for differentiable Neural Architecture Search. arXiv preprint arXiv:1909.11926 (2019)
16. Li, G., Qian, G., Delgadillo, I.C., Müller, M., Thabet, A., Ghanem, B.: Sgas: Sequential greedy architecture search. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2020)

17. Li, Y., Yuan, Y.: Convergence Analysis of Two-layer Neural Networks with Relu Activation. In: Advances in Neural Information Processing Systems. pp. 597–607 (2017)
18. Liang, H., Zhang, S., Sun, J., He, X., Huang, W., Zhuang, K., Li, Z.: DARTS+: Improved Differentiable Architecture Search with Early Stopping. arXiv preprint arXiv:1909.06035 (2019)
19. Liu, H., Simonyan, K., Yang, Y.: DARTS: Differentiable Architecture Search. In: International Conference on Learning Representations (2019)
20. Maclaurin, D., Duvenaud, D., Adams, R.: Gradient-based hyperparameter optimization through reversible learning. In: International Conference on Machine Learning (2015)
21. Maddison, C.J., Mnih, A., Teh, Y.W.: The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. In: International Conference on Learning Representations (2017)
22. Nayman, N., Noy, A., Ridnik, T., Friedman, I., Jin, R., Zelnik-Manor, L.: XNAS: Neural Architecture Search with Expert Advice. In: Advances in Neural Information Processing Systems (2019)
23. Pham, H., Guan, M.Y., Zoph, B., Le, Q.V., Dean, J.: Efficient Neural Architecture Search via Parameter Sharing. In: International Conference on Machine Learning (2018)
24. Ramachandran, P., Zoph, B., Le, Q.V.: Searching for activation functions. arXiv preprint arXiv:1710.05941 (2017)
25. Real, E., Aggarwal, A., Huang, Y., Le, Q.V.: Regularized Evolution for Image Classifier Architecture Search. International Conference on Machine Learning, AutoML Workshop (2018)
26. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: MobileNetV2: Inverted Residuals and Linear Bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4510–4520 (2018)
27. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A Simple Way to Prevent Neural Networks from Overfitting. The Journal of Machine Learning Research **15**(1), 1929–1958 (2014)
28. Tan, M., Chen, B., Pang, R., Vasudevan, V., Le, Q.V.: MnasNet: Platform-Aware Neural Architecture Search for Mobile. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
29. Tan, M., Le, Q.V.: EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In: International Conference on Machine Learning (2019)
30. Tan, M., Le., Q.V.: MixConv: Mixed Depthwise Convolutional Kernels. The British Machine Vision Conference (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. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
32. Xie, S., Zheng, H., Liu, C., Lin, L.: SNAS: Stochastic Neural Architecture Search. International Conference on Learning Representations (2019)
33. Xu, Y., Xie, L., Zhang, X., Chen, X., Qi, G.J., Tian, Q., Xiong, H.: PC-DARTS: Partial Channel Connections for Memory-efficient Differentiable Architecture Search. In: International Conference on Learning Representations (2020)
34. Zela, A., Elsken, T., Saikia, T., Marrakchi, Y., Brox, T., Hutter, F.: Understanding and robustifying differentiable architecture search. In: International Conference on Learning Representations (2020), https://openreview.net/forum?id=H1gDNyrKDS

35. Zheng, X., Ji, R., Tang, L., Zhang, B., Liu, J., Tian, Q.: Multinomial Distribution Learning for Effective Neural Architecture Search. International Conference on Computer Vision (2019)
36. Zoph, B., Vasudevan, V., Shlens, J., Le, Q.V.: Learning Transferable Architectures for Scalable Image Recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. vol. 2 (2018)