Understanding and Enhancing Differentiable Architecture Search from Information Bottleneck Perspective

Haidong Kang[1], **Lianbo Ma**[1*], **Pengjun Chen**[1], **Qiang He**[1], **Bo Yi**[1]

[1]Northeastern University, China
kanghaidong@qhd.neu.edu.cn, malb@swc.neu.edu.cn, pjchen@stumail.neu.edu.cn,
heqiangcai@gmail.com, yibo@cse.neu.edu.cn

Abstract

Performance collapse is an intractable issue of Differentiable Architecture Search (DAS), where severe performance degradation of DAS happens when it trains on different search spaces or datasets. We theoretically analyze the issue from the information bottleneck (IB) perspective, and disclose that a solution to overcome this problem is to seek the bifurcation point of IB tradeoff between compression and prediction of the supernet. To this end, we propose a simple yet highly effective method, namely, Batch Entropy-decay Regularization (BER), to guide the learning of DAS, which restricts compression in DAS by imposing a penalty on the architecture parameters. Comprehensive theoretical analyses demonstrate that BER is able to completely resolve DAS's performance collapse issue. Compared with a number of state-of-the-art DAS variants, BER shows its overwhelmingly better performance on 7 search spaces (i.e., NAS-Bench-201, DARTS, S1-S4, MobileNet-like) and 5 popular datasets (i.e., CIFAR-10, CIFAR-100, ImageNet1k, PASCAL VOC 2007, and MS COCO 2017).

Extended version — https://github.com/NEU-MLTeam/r-DARTS/blob/main/r-darts_appendix.pdf

Introduction

Neural Architecture Search (NAS) has emerged as a promising way for automatic design of task-specific deep neural networks (DNNs) (Mei et al. 2019), avoiding dependence of extensive human expertise.

In practice, the DNNs designed by NAS often exhibit better performance than the handcrafted ones (Ren et al. 2021). The typical way of existing NAS methods is to relax the discrete operation selection problem to learn differentiable architecture parameters, termed as Differentiable Architecture Search (DAS) (Bender et al. 2018). In particular, DAS focuses on the optimization of supernet, where the network weights and architecture parameters are optimized alternately in a differentiable manner (Brock et al. 2017), (Pham et al. 2018), as shown in Fig. 1a, and it shows better search efficiency than others (e.g., reinforcement learning (Zoph and Le 2017; Baker et al. 2016), evolutionary algorithm (Xie et al. 2020) based methods). In DAS, the performance of candidate architectures is evaluated based on the weights directly inherited from the supernet; thus, the search cost is significantly reduced.

However, existing DAS methods (e.g., DARTS (Liu, Simonyan, and Yang 2018)) suffer from the performance collapse issue, i.e., severe performance degradation on specific search spaces and/or datasets (Zela et al. 2019). Especially, recent studies (Xu et al. 2019; Zela et al. 2019) have demonstrated that DAS tends to select parameter-free operations (e.g., skip connection, etc.) in the later stage of architecture search, which leads to the performance deterioration (Chu et al. 2020a). To overcome this issue, several solutions have been developed, including exploiting effective indicators (e.g., Hessian eigenvalues (Zela et al. 2019)), limiting the number of parameter-free operations (e.g., skip connections (Chen et al. 2019; Liang et al. 2019)), regularizing architecture parameters (Ye et al. 2022), and adjusting the search/discretization strategies (Dong and Yang 2019; Zhang et al. 2021a; Chu et al. 2020a). However, the design of these methods lacks a unified and in-depth theoretic overview and principled explanation, since the theoretical foundation understanding of DAS' learning process remains unsatisfactory. Basic questions about the cause of dominant parameter-free operations in the search, and the design principles of optimization methods, are not yet well understood.

In this paper, we attempt to explain and overcome this problem from the information theoretic perfective. First, we find that the learning process of DAS can be explained by the Information Bottleneck (IB) theory (Tishby, Pereira, and Bialek 2000), which indicates that the goal of DAS's learning is to search for an information theoretic tradeoff between compression and prediction. Then, we explain the reason why the performance collapse happens in the architecture search: the entropy value of architecture parameters in the compression stage is too large, leading to the performance collapse. This is also consistent with the dominant problem of parameter-free operations, since excessive selection of parameter-free operations often leads to high entropy value.

Derived from the IB theory, we propose an efficient Batch Entropy-decay regularization (BER) method to guide the learning of DAS, termed as r-DARTS. The comparison between our method and existing ones is listed in Fig. 1.

*Corresponding author

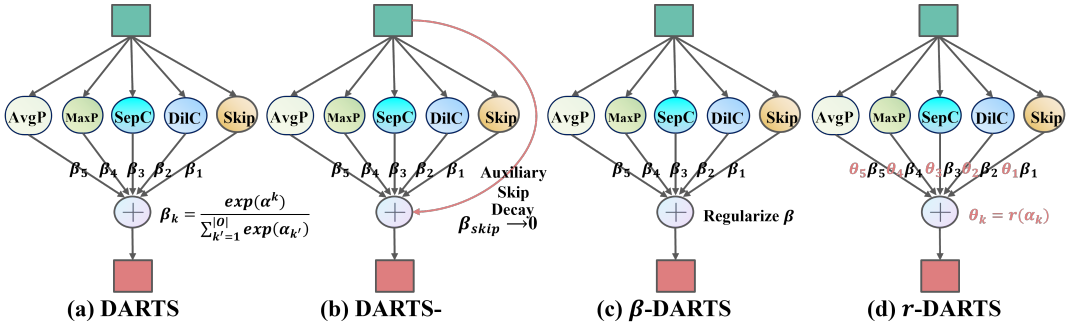


Figure 1: Illustrative examples (a) DARTS (Liu, Simonyan, and Yang 2018), (b) DARTS- (Chu et al. 2020a), (c) β-DARTS (Ye et al. 2022), and (d) our proposed r-DARTS. DARTS- employs an auxiliary skip operation with a decay rate β_{skip}. β-DARTS aims to keep the value (β) of activated architecture parameters from being too large. r-DARTS introduces the batch entropy-decay regularization to prevent the entropy value (r) of architecture parameters from being too large.

In contrast to existing methods, our BER restricts compression of DAS by imposing a penalty on the entropy of architecture parameters, keeping the entropy values from too large, different from existing methods that strive to prevent the architecture parameter values from too large. We provide theoretical analysis and extensive experiments on different search spaces (e.g., NAS-Bench-201, and DARTS) with multiple datasets (e.g., CIFAR-10, CIFAR-100, and ImageNet1k) to validate the effectiveness of the proposed method in dealing with the performance collapse issue.

To sum up, the contributions of this paper include:

(1) We rethink the learning process of DAS from the information bottleneck perspective, and find that the goal of DAS can be formulated as an information theoretic *tradeoff between compression and prediction*. From this concept, we disclose that the increasing entropy incurred by unrestricted compression process is the key cause that leads to the performance collapse issue.

(2) To the best of our knowledge, this work is the first attempt to employ IB concept to understand and resolve the performance collapse of DAS. To achieve this, we propose a novel *Batch Entropy-decay Regularization (BER)* to guide the learning of DAS, keeping the entropy value of target architecture parameters from too large. This is different from existing methods, which aim to restrict the numbers of parameter-free operations.

(3) We provide theoretical analysis and extensive experiments to validate the effectiveness and advantages of the proposed method in handling the performance collapse. Especially, our r-DARTS achieves optimal performance on various search spaces with popular benchmarks. As shown, our proposed BER is simple and efficient, and it can be easily embedded in DAS without requiring any extra training.

Rethinking DAS's Learning from Information Bottleneck Perspective

Revisit the Performance Collapse of DAS

The goal of DAS (i.e., DARTS) is to explore the architecture of both normal and reduction cells for constructing the entire network (Liu, Simonyan, and Yang 2018). A cell can be represented as a directed acyclic graph (DAG) with N nodes, where each node corresponds to a latent representation, and each edge (i, j) transforms information between nodes (i and j) with several candidate operations. DAS employs continuous relaxation through architecture parameters α to blend the outputs of various DNN operations. This transformation converts the discrete operation selection into a differentiable bi-level optimization problem:

$$\bar{o}^{(i,j)}(x) = \sum_{k=1, o\in\mathcal{O}}^{|\mathcal{O}|} \beta_k^{(i,j)} o_k(x), \quad \beta_k^{(i,j)} = \frac{\exp\left(\alpha_k^{(i,j)}\right)}{\sum_{k'=1}^{|\mathcal{O}|} \exp\left(\alpha_{k'}^{(i,j)}\right)}, \quad (1)$$

where the operation mixing weights for a pair of nodes (i, j) are parameterized by a vector $\alpha^{(i,j)}$ of dimension $|\mathcal{O}|$. x and $\bar{o}$ are the input and mixed output of an edge, respectively, $\mathcal{O}$ is the candidate operation set (e.g., convolution, max pooling, zero), where function $o_k(\cdot)$ represents k-th operation. β denotes the softmax-activated architecture parameter set. The task of DARTS is to learn a set of the operation mixing weights $\alpha^{(i,j)}$, namely, architecture parameters $\alpha = \{\alpha^{(i,j)}\}$.

In this way, we can perform architecture search in a differentiable manner by solving the following bi-level optimization objective:

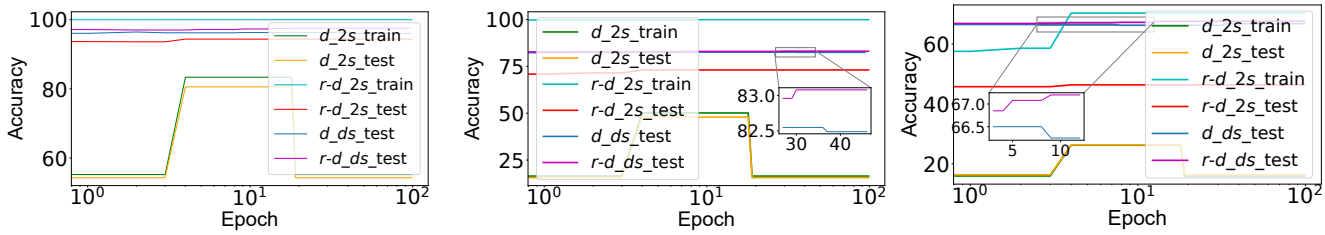
$$\begin{aligned} &\nabla_\alpha \mathcal{L}_{val}(w^*(\alpha), \alpha) \\ &\approx \nabla_\alpha \mathcal{L}_{val}(w - \xi \nabla_w \mathcal{L}_{train}(w, \alpha), \alpha), \end{aligned} \quad (2)$$

where ξ denotes the learning rate for a step of architecture parameter optimization, L_{val} and L_{train} are Cross-Entropy loss function. Typically, such bi-level optimization for architecture parameters α and network weights w is realized by gradient descent, and w^* is approximated by one-step forward on current w.

Rethinking Information Characteristics of DAS

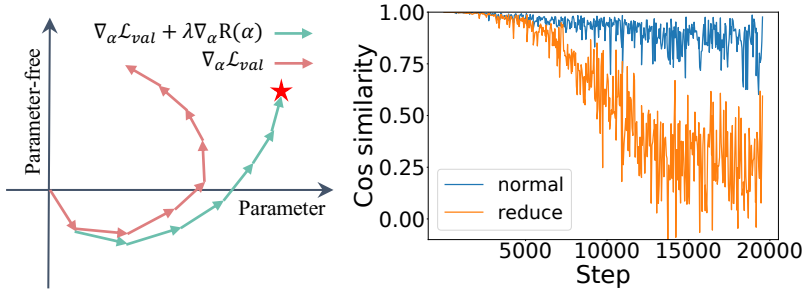
To rigorously explain the regularization mechanism in Differentiable Architecture Search (DAS), we revisit the architecture optimization problem from a **variational Information Bottleneck (VIB)** (Tishby and Zaslavsky 2015; Sun et al. 2022b; Alemi et al. 2016; Dai et al. 2018) perspective. The IB principle seeks a compressed representation of the input x that is maximally informative for predicting the target y. Formally, the IB objective is:

$$\min_{p(z|x)} \ \mathbb{E}_{p(x,y)} \left[\mathbb{E}_{p(z|x)}[-\log p(y|z)]\right] + \beta I(x; z), \quad (3)$$



(a) CIFAR-10 (b) CIFAR-100 (c) ImageNet16-120

Figure 2: Accuracy comparison of DARTS(d) and our r-DARTS(r-d) on DARTS(ds) and NAS-Bench-201($2s$) search space with CIFAR-10/100 and ImageNet16-120. (a) CIFAR-10; (b) CIFAR-100; (c) ImageNet16-120.



(a) An illustrative example (b) Reduce and normal cell

Figure 3: (a) is an example of BER to DAS. The salmon pink arrows represent the optimization of DAS, green arrows denote the optimization updates of r-DARTS; (b) is mean cosine similarity of gradient between $\nabla_\alpha \mathcal{L}_{val}$ and $\nabla_\alpha \mathcal{L}_{val} + \nabla_\alpha r(\alpha)$ on reduce normal cell, smaller cosine similarity represents the bigger angle of gradient's direction between DAS and our r-DARTS.

where z is a latent representation and β controls the trade-off between predictive accuracy and compression.

In the context of DAS optimization, we treat the latent variable $z = (\alpha, w)$ as jointly representing the architecture α and network weights w. To make the IB loss tractable, we introduce variational approximations: (1) $q(\alpha|x)$: approximate posterior over architecture; (2) $q(w|\alpha, x)$: posterior over weights conditioned on architecture; and (3) $p(\alpha), p(w|\alpha)$: prior distributions over architecture and weights. This yields the following variational upper bound:

$$\mathcal{L}_{\text{IB}} = \mathbb{E}_{p(x,y)} \left[\mathbb{E}_{q(\alpha|x)} \mathbb{E}_{q(w|\alpha,x)} [-\log p(y|\alpha, w)] \right] + \beta \mathrm{KL}[q(\alpha|x) \| p(\alpha)] + \tag{4}$$

$$\beta \mathbb{E}_{q(\alpha|x)} \mathrm{KL}[q(w|\alpha, x) \| p(w|\alpha)]. \tag{5}$$

We can transfer the above Equation to a two-stage optimization strategy via VIB:

$$\min_{q(w|\alpha,x)} \mathbb{E}_{p(x,y)} \left[\mathbb{E}_{q(w|\alpha,x)} [-\log p(y|\alpha, w)] \right] + \tag{6}$$

$$\beta \mathrm{KL}[q(w|\alpha, x) \| p(w|\alpha)]. \tag{7}$$

Assuming $q(w|\alpha, x)$ is sharply peaked, we approximate with point estimate $w^*(\alpha)$, leading to standard SGD on training data.

$$\min_{q(\alpha|x)} \mathbb{E}_{p(x,y)} \left[\mathbb{E}_{q(\alpha|x)} [-\log p(y|\alpha, w^*(\alpha))] \right] + \tag{8}$$

$$\beta \mathrm{KL}[q(\alpha|x) \| p(\alpha)]. \tag{9}$$

Assuming $q(\alpha|x)$ is parameterized by a softmax distribution and $p(\alpha)$ is a uniform or sparsity-inducing prior, we obtain the surrogate objective:

$$\mathcal{L}_{val}^{\text{IB}}(\alpha) = \mathcal{L}_{val}(w^*(\alpha), \alpha) + \beta \mathrm{KL}[q(\alpha) \| p(\alpha)]. \tag{10}$$

If we further assume $q(\alpha)$ is a relaxed categorical distribution (e.g., Gumbel-Softmax) and $p(\alpha)$ is uniform, the KL term becomes:

$$\mathrm{KL}[q(\alpha) \| p(\alpha)] = -H(q(\alpha)) + \text{const}. \tag{11}$$

Where const denotes constant. Therefore, we can obtain the simplified surrogate loss as follows:

$$\mathcal{L}_{val}^{\text{IB}}(\alpha) = \mathcal{L}_{val}(w^*(\alpha), \alpha) - \beta H(\alpha), \tag{12}$$

which $\mathcal{L}_{val}(w^*(\alpha), \alpha)$ is a validation cross-entropy loss. To solve the $\mathcal{L}_{val}^{\text{IB}}(\alpha)$, we use gradient decent in our paper. Although β theoretically represents the trade-off between information compression and prediction accuracy, reflecting the physical meaning of the information bottleneck, in practice it is treated as a hyperparameter and assigned a fixed value (Wei et al. 2022; Sun et al. 2022a). This is similar to

the role of the regularization weight λ in conventional regularization terms, which balances the training objective and the structural regularization.

In DAS, network weights, and architecture parameters are optimized alternately in a differentiable manner. Then, the optimization of DARTS can be divided into early epochs and later epochs. This can be validated by experimental observation (as shown in Fig. 4). These results strongly validate our statement that the learning process in DAS is divided into feature extraction and compression. From the above IB perspective, the learning process of DAS can be divided into two stages. As shown, in the early epochs of the search process (***feature extraction stage***), DAS selects operations that maximize learning of input features, i.e., parameter operations (e.g., 3x3 separable convolutions). These operations with their trainable parameters, are crucial for capturing and representing the significant features from the input, and thus they are dominant, being assigned with greater weights. In the later epochs of the search process, as the state of feature extracting tends to stabilize, the ***compression stage*** is activated, and DAS selects operations that maximize compression of learned features, i.e., parameter-free operations (e.g., skip connections). The primary task of the model learning turns to reduce redundant information from extracted features for gradual convergence. In this process, the mean weights of parameter-free operations gradually increase until dominant, since they are more suitable to the redundant feature elimination. For example, the max pooling 3 x 3 operation can reduce the feature map to 9× its original size.

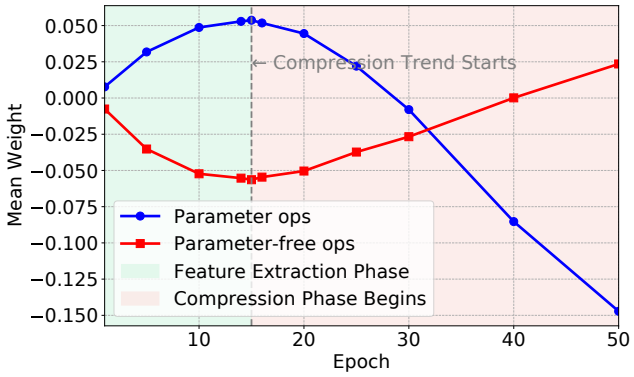


Figure 4: Empirical validation of feature extraction and compression in DAS.

Discussion. However, DAS compression can be excessive, favoring compression-friendly operations (e.g., max pooling, skip connections), which may cause optimization to stall at poor local minima due to very small effective learning rates, leading to performance collapse. To address this, based on the above Eq. 12, we propose Batch Entropy-decay Regularization (BER) to stabilize DAS optimization by minimizing the entropy of architecture parameters $H(\alpha)$. As shown in Fig. 3a, BER guides gradient updates to balance parameterized and parameter-free operations, preventing the dominance of the latter. This is supported by the mean cosine similarity results in Fig. 3b and 3c, where BER maintains stable similarity throughout training.

Batch Entropy-decay Regularization for DAS

Batch Entropy-decay Regularization

The goal of our paper is to overcome performance collapse of DAS.

To fulfill this goal, we propose a simple yet effective Batch Entropy-decay Regularization (BER) method to guide the learning of DAS, dubbed, r-DARTS, tailored for DAS.

The entropy in terms of architecture parameters α actually represents the stability level of the searching and discretization process of DAS, and thus it is intuitive to impose the explicit regularization item on the entropy of α for regularizing the compression of DAS. Similar to the idea of most regularization methods, the core purpose of Batch Entropy-decay regularization (BER) is to constrain the value of r from changing too much, formulated as:

$$\bar{\alpha}_k^t = r(\alpha_k^t)\alpha_k^t, \tag{13}$$

where r is the proposed BER function. For simplicity, we use a function r with α as the independent variable to express the total influence of BER here. Then, the single-step update of the architecture parameters can be formulated as:

$$\alpha^{t+1} \leftarrow \alpha^t - \eta_\alpha \cdot \nabla_\alpha \mathcal{L}_{val}(\omega^t, \alpha^t), \tag{14}$$

where η_α and $\nabla_\alpha \mathcal{L}_{val}(\omega^t, \alpha^t)$ are the learning rate of architecture parameters and the corresponding gradient of the loss with respect to α^t. However, the entropy in terms of the values (α^t) of architecture parameters actually represents the stability level of the searching and discretization process of DAS, and thus it is intuitive to impose the explicit regularization item on the entropy of α^t for regularizing the compression of DAS. Then, let r and λ denote the regularization function and the hyperparameter respectively, the updated formulation of the loss function $\mathcal{L}^*_{val}(\omega^t, \alpha^t)$ can be defined as:

$$\mathcal{L}^*_{val}(\omega^t, \alpha^t) = \mathcal{L}_{val}(\omega^t, \alpha^t) + \lambda \cdot r(\alpha^t). \tag{15}$$

Consequently, the new formulation of the single-step gradient descend update of the α^t can be redefined as:

$$\bar{\alpha}^{t+1} \leftarrow \alpha^t - \eta_\alpha \cdot \nabla_\alpha \mathcal{L}_{val}(\omega^t, \alpha^t) - \eta_\alpha \lambda \cdot \nabla_\alpha r\left(\alpha^t\right), \tag{16}$$

where $\bar{\alpha}^{t+1}$ denotes the single-step update with regularization, $\nabla_\alpha r(\alpha^t)$ is the gradient of the regularization function with respect to α^t. Accordingly, all we need is to look for a suitable regularization function (r). In this work, we introduce a simple yet efficient Batch Entropy-decay regularization function. Let $\sigma(.)$ correspond to the softmax function, which is monotonically increasing. Then, the regularization function can be defined as:

$$\begin{aligned} r(\alpha^t) &= \frac{\sum_{k=1}^{N} r^*(\alpha_k^t)}{N}, \\ r^*(\alpha_k^t) &= \mathbb{E}_{p(\boldsymbol{\beta_k^t})}[-\log p(\boldsymbol{\beta_k^t})], \quad \boldsymbol{\beta_k^t} = \sigma(\alpha_k^t), \end{aligned} \tag{17}$$

Given that $|\mathcal{O}|$ denotes the size of the operation set and N represents the number of edges to search, $r(\alpha^t)$ is a mapping $\mathrm{R}^{N\times|\mathcal{O}|} \rightarrow \mathrm{R}$ that computes the mean entropy of the

architecture parameter α^tacross all edges. Similarly, $r^*(\alpha_k^t)$ is a mapping $\mathrm{R}^{|\mathcal{O}|} \rightarrow \mathrm{R}$ that calculates the entropy of the architecture parameter α_k^t on edge k. Observing the Eq. 17, we can get the following conclusions: (1) the BER function r monotone decrease; (2) When α is the largest, r is the smallest, when α is the smallest, r is the largest.

To conduct a detailed analysis on the efficiency of our proposed regularization, we decompose $\nabla_{\alpha_k^t} r^*(\alpha_k^t)$ as follows:

$$\nabla_{\alpha_k^t} r^*(\alpha_k^t) = \nabla_{\boldsymbol{\beta_k^t}} r^*(\alpha_k^t) \boldsymbol{J_\sigma}(\alpha_k^t), \tag{18}$$

where $\boldsymbol{J_\sigma}(\alpha_i^t) = diag(\boldsymbol{\beta_k^t}) - \boldsymbol{\beta_k^t}(\boldsymbol{\beta_k^t})^{\mathrm{T}}$ corresponds to the Jacobian matrix of the softmax function on one edge, and $diag(.)$ is the diagonal matrix, and $\nabla_{\beta_k^t} r(\alpha_k^t)$ can be rewritten as $-(I + \log \beta_k^t)$, I represents the vector of ones. Then, $\nabla \alpha_k^t r(\alpha_k^t)$ can be written as:

$$\begin{aligned}
\nabla_{\alpha_k^t} r^*(\alpha_k^t) &= -(I + \log \beta_k^t) \odot \beta_k^t + (I + \log \beta_k^t)(\beta_k^t(\beta_k^t)^T) \\
&= -\beta_k^t - \log \beta_k^t \odot \beta_k^t + I(\beta_k^t(\beta_k^t)^T) + \log \beta_k^t(\beta_k^t(\beta_k^t)^T) \\
&= -\beta_k^t - \log \beta_k^t \odot \beta_k^t + \beta_k^t + \log \beta_k(\beta_k^t)^T \odot \beta_k^t \\
&= -\log \beta_k^t \odot \beta_k^t - \mathbb{E}_{p(\beta_k^t)}[-\log_P(\beta_k^t)] \odot \beta_k^t \\
&= -\beta_k^t \odot (\mathbb{E}_{p(\beta_k^t)}[-\log p(\beta_k^t)] + \log \beta_k^t),
\end{aligned} \tag{19}$$

where $\odot$ denotes Hadamard product. By substituting Eq. 19 into Eq. 16, we can obtain the following equation:

$$\begin{aligned}
\bar{\alpha}^{t+1} \leftarrow \alpha^t - \eta_\alpha \cdot \nabla_\alpha \mathcal{L}_{val}(\omega^t, \alpha^t) - \\
\eta_\alpha \lambda \cdot \frac{[\nabla_{\alpha_1^t} r^*(\alpha_1^t), ..., \nabla_{\alpha_k^t} r^*(\alpha_k^t), ..., \nabla_{\alpha_N^t} r^*(\alpha_N^t)]}{N}.
\end{aligned} \tag{20}$$

From the above equation, we have: (1) r imposes a penalty on each operation in search space, and can be calculated by using known variables with negligible computation cost; (2) In the current iteration, the variance of α is determined by the batch data from validation datasets, thus called batch entropy-decay regularization.

Theoretical Analysis

The previous work (Zhou et al. 2020) theoretically and explicitly reveals that the performance collapse of DAS heavily relies on parameter-free operations (i.e., skip) in the DAS. Based on previous work (Zhou et al. 2020), we conduct a theoretical analysis about the effectiveness of our proposed method, which can validate our proposed method relies on parameter- operations (i.e., conv) more than parameter-free operations (i.e., skip) in the DAS. Therefore, we can prove our proposed method can overcome the performance collapse of DAS from a theoretical perspective. Suppose that the search space of DAS involves the parameter operations (e.g., conv, etc.) and parameter-free operations (e.g., skip, etc.), and the training loss is MSE. In our target scenario, we consider three representative operations in the network, i.e., zero, skip, and conv. Specially, the pooling reveals the same behaviors as skip. Then, the layers in cell can be defined as:

$$\begin{aligned}
X^{(l)} = \sum_{s=0}^{l-1} \big(&\alpha_{s,1}^{(l)} \mathrm{zero}(X) + \alpha_{s,2}^{(l)} \mathrm{skip}(X) \\
&+ \alpha_{s,3}^{(l)} \mathrm{conv}(W_s^{(l)}; X^{(s)})\big) \\
&\in \mathbb{R}^{m \times p} \quad (l = 1, \cdots, h-1),
\end{aligned} \tag{21}$$

where h denotes the number of nodes in cell, $X^{(l)}$ is output of previous l-th cell, W denotes model weights (i.e., convolution operation weights), $\boldsymbol{\alpha}_{s,t}^{(l)}$ denotes architecture parameters at t-th operation, $\mathrm{skip}(X) = X$, and $\mathrm{zero}(X) = 0$.

When fixing the values of architecture parameters to optimize network weights W via gradient descent, according to (Zhou et al. 2020), we have:

$$\varsigma_s = (\boldsymbol{\alpha}_{s,3}^{(h-1)})^2 \prod_{t=0}^{s-1} (\boldsymbol{\alpha}_{t,2}^{(s)})^2, \tag{22}$$

where $0 \leq s < l$. By observing ς_s, we can find that $\boldsymbol{\alpha}_{t,2}^{(s)}$ of skip operation is larger than $\boldsymbol{\alpha}_{s,3}^{(h-1)}$ of conv operation. By imposing BER on the loss of DAS according to Eq. 13, we have:

$$\varsigma_s = (\boldsymbol{r}_{s,3}^{(h-1)} \cdot \boldsymbol{\alpha}_{s,3}^{(h-1)})^2 \prod_{t=0}^{s-1} (\boldsymbol{r}_{t,2}^{(s)} \cdot \boldsymbol{\alpha}_{t,2}^{(s)})^2. \tag{23}$$

As mentioned before, r becomes smaller when α is larger, and r becomes larger when α is smaller. Given that $\boldsymbol{\alpha}_{t,2}^{(s)} > \boldsymbol{\alpha}_{s,3}^{(h-1)}$, we can obtain $\boldsymbol{r}_{s,3}^{(h-1)} > \boldsymbol{r}_{t,2}^{(s)}$, which keep consistent with our experimental observation. This indicates the convergence of DAS depends on the weights $\boldsymbol{\alpha}_{s,3}^{(h-1)}$ convolution operation heavier than weights $\boldsymbol{\alpha}_{t,2}^{(s)}$ of skip connections via impose influence of BER in DAS. Finally, we prove our method can overcome the performance collapse of DAS from a theoretical perspective.

Experiment and Discussions

Experimental Settings

We conduct extensive experimental comparison on popular search spaces (NAS-Bench-201 and DARTS, S1-S4) and benchmarks (i.e., CIFAR-10, CIFAR-100, ImageNet1k, PASCAL VOC 2007, and MS COCO 2017) to validate the effectiveness of our proposed method. The detailed experimental settings are presented in **App. A**.

Results on NAS-Bench-201

The experimental results on NAS-Bench-201 with three datasets (CIFAR-10, CIFAR-100, and ImageNet16-120) are summarized in Table 1. Considering the randomness of searching, we report the averaged results with 10 runs. As shown, our r-DARTS achieves remarkable accuracy results in the three datasets and significantly surpasses its peer competitors in terms of search speed. More specifically, our r-DARTS achieves best accuracy results on NAS-Bench-201 with CIFAR-10 and ImageNet-16-120, i.e., 94.37 % and

Methods	Search Cost (GPU-hours)	CIFAR-10		CIFAR-100		ImageNet16-120		Search (Method)
		valid (%)	test (%)	valid (%)	test (%)	valid (%)	test (%)	
DARTS(1st) (Liu, Simonyan, and Yang 2018)	3.20	39.77±0.00	54.30±0.00	15.03±0.00	15.61±0.00	16.43±0.00	16.32±0.00	Gradient
GDAS (Dong and Yang 2019)	8.70	89.89±0.08	93.61±0.09	71.34±0.04	70.70±0.30	41.59±1.33	41.71±0.98	Gradient
PC-DARTS (Xu et al. 2019)	-	89.96±0.15	93.41±0.30	67.12±0.39	67.48±0.89	40.83±0.08	41.31±0.22	Gradient
DARTS- (Chu et al. 2020a)	3.20	91.03±0.44	93.80±0.40	71.36±1.51	71.53±1.51	44.87±1.46	45.12±0.82	Gradient
SNAS (Xie et al. 2020)	-	90.10±1.04	92.77±0.83	69.69±2.39	69.34±1.98	42.84±1.79	43.16±2.64	Gradient
DSNAS (Hu et al. 2020)	-	89.66±0.29	93.08±0.13	30.87±16.40	31.01±16.38	40.61±0.09	41.07±0.09	Gradient
iDARTS (Zhang et al. 2021a)	-	89.86±0.60	93.58±0.32	70.57±0.24	70.83±0.48	40.38±0.59	40.89±0.68	Gradient
β-DARTS (Ye et al. 2022)	3.20	**91.55±0.00**	94.36±0.00	73.49±0.00	**73.51±0.00**	46.37±0.00	46.34±0.00	Gradient
Λ-DARTS (Movahedi et al. 2023)	-	**91.55±0.00**	94.36±0.00	73.49±0.00	**73.51±0.00**	46.37±0.00	46.34±0.00	Gradient
GDAS-AER^c (Jing, Chen, and Xu 2023)	47.9	90.17±0.16	93.37±0.07	70.15±0.86	70.35±0.41	42.12±1.63	42.26±1.20	Gradient
IS-DARTS (He et al. 2024)	2.0	**91.55±0.00**	94.36±0.00	**73.49±0.00**	**73.51±0.00**	46.37±0.00	46.34±0.00	Gradient
r-DARTS(λ=0.50, C10, e=15)	**0.80**	**91.55±0.00**	94.36±0.00	**73.49±0.00**	**73.51±0.00**	46.37±0.00	46.34±0.00	Gradient
r-DARTS(λ=0.75, C10, e=5)	**0.16**	91.50±0.00	**94.37±0.00**	73.31±0.00	73.09±0.00	45.59±0.00	46.33±0.00	Gradient
r-DARTS(λ=0.75, C100, e=5)	**0.20**	91.07±0.00	93.99±0.00	72.51±0.00	72.88±0.00	**46.38±0.00**	**47.31±0.00**	Gradient
NASWOT (Mellor et al. 2021)	0.10	89.69 ± 0.73	92.96 ± 0.81	69.86 ± 1.21	69.98 ± 1.22	43.95 ± 2.05	44.44 ± 2.10	Training-free
AZ-NAS (Lee and Ham 2024)	0.59	-	93.49 ± 0.30	-	70.33 $\pm$ 1.16	-	44.34 $\pm$ 1.26	Training-free
SWAP (Peng et al. 2024)	-	-	90.48 ± 0.94	-	67.13 ± 1.83	-	35.40 ± 3.96	Training-free
ParZC (Dong et al. 2025)	0.02	-	94.36 ± 0.01	-	73.49 ± 0.02	-	46.34 ± 0.04	Training-free
optimal	-	91.61	94.37	73.49	73.51	46.77	47.31	

Table 1: Performance comparison on NAS-Bench-201 with CIFAR-10/100, and ImageNet16-120 datasets over 10 runs. Specifically, our r-DARTS is only performed in CIFAR-10. "e" denotes the number of epochs when r-DARTS has converged. "-" indicates that the value is not found in the original paper.

Methods	Top-1(%)	# Params (M)	FLOPS (M)	Search Cost (GPU-days)
ResNet50 (He et al. 2016)	75.3	25.6	4100	
MobileNetV1 (Howard et al. 2017)	70.6	4.2	575	-
AmoebaNet-A (Zoph et al. 2018)	74.5	6.4	555	3150
ProxylessNAS-RL (Cai, Zhu, and Han 2018)	74.6	5.8	465	8.3
NASNet-A (Real et al. 2019)	74.0	5.3	564	2000
DARTS (Liu, Simonyan, and Yang 2018)	73.3	4.7	574	4
FBNet (Wu et al. 2019)	74.9	5.5	375	216
PC-DARTS(Img) (Xu et al. 2019)	75.8	5.3	597	3.7
P-DARTS(C100) (Chen et al. 2019)	75.3	5.1	577	0.3
DARTS+ (Liang et al. 2019)	76.3	5.1	591	0.2
DARTS-(Img) (Chu et al. 2020a)	76.2	4.9	467	4.5
FairDARTS-B(Img) (Chu et al. 2020b)	75.1	4.8	541	-
SNAS(C10) (Xie et al. 2020)	72.7	4.3	522	1.5
DARTS+PT(C10) (Wang et al. 2021)	74.5	4.6	-	0.8
DOTS(C10) (Gu et al. 2021)	75.7	5.2	581	0.3
β-DARTS(C100) (Ye et al. 2022)	75.8	5.4	597	0.4
Λ-DARTS (Movahedi et al. 2023)	75.7	5.2	-	-
OLES (Jiang et al. 2023)	75.5	4.7	-	0.4
FP-DARTS(C10) (Wang et al. 2023)	75.7	5.4	-	0.08
PDARTS-AER^{ed} (Jing, Chen, and Xu 2023)	76.0	5.1	578	2.0
IS-DARTS (He et al. 2024)	75.9	6.4	-	0.42
r-DARTS(λ=1, C10)	**76.9**	4.9	573	0.25
NAO (Luo et al. 2018)	74.3	11.4	584	200
SemiNAS (Luo et al. 2020)	76.5	6.3	599	4
WeakNAS (Wu et al. 2021)	76.5	5.5	591	2.5
NASI-ADA(C10) (Shu et al. 2022)	75.0	4.9	559	0.01
SWAP (Peng et al. 2024)	76.0	5.8	-	0.006

Table 2: Results on the DARTS search space in ImageNet1k.

47.31 %. For the **generalization ability**, we can find that the architecture discovered in CIFAR-10 achieves competitive results in CIFAR-10, CIFAR-100, and ImageNet-16-120. For the **convergence ability**, as shown in Table 1, compared with the its peer competitors on NAS-Bench-201, the convergence rate of r-DARTS is 20.0$\times$ more than β-DARTS. Although training-free methods (i.e., SWAP) obtain better speed, r-DARTS achieves optimal accuracy. In summary, the above results validate that r-DARTS has a strong ability of seeking the optimal architecture rapidly.

Results on DARTS Search Space

ImageNet1k. we provide performance comparison in the large dataset ImageNet1k, as depicted in Table 2. Compared with the results on a small dataset (CIFAR-10), our proposed r-DARTS achieves better accuracy with the least search costs in ImageNet1k. This indicates that our r-DARTS has great potential in exploring optimal architectures that can generalize to other large datasets without restarting NAS. In addition, the architecture found by r-DARTS also outperforms all the competitors in terms of all other metrics. These encouraging results again validate the effectiveness of our method in transferring knowledge from the small dataset (CIFAR-10) to large datasets (ImageNet1k), which leads to significant improvement in terms of search accuracy and search efficiency. Notably, r-DARTS shows that differentiable NAS is still superior to training-free methods (i.e., SWAP) in terms of accuracy. The detailed experimental results on **CIFAR-10/100** are presented in **App. B**.

Memory Analysis

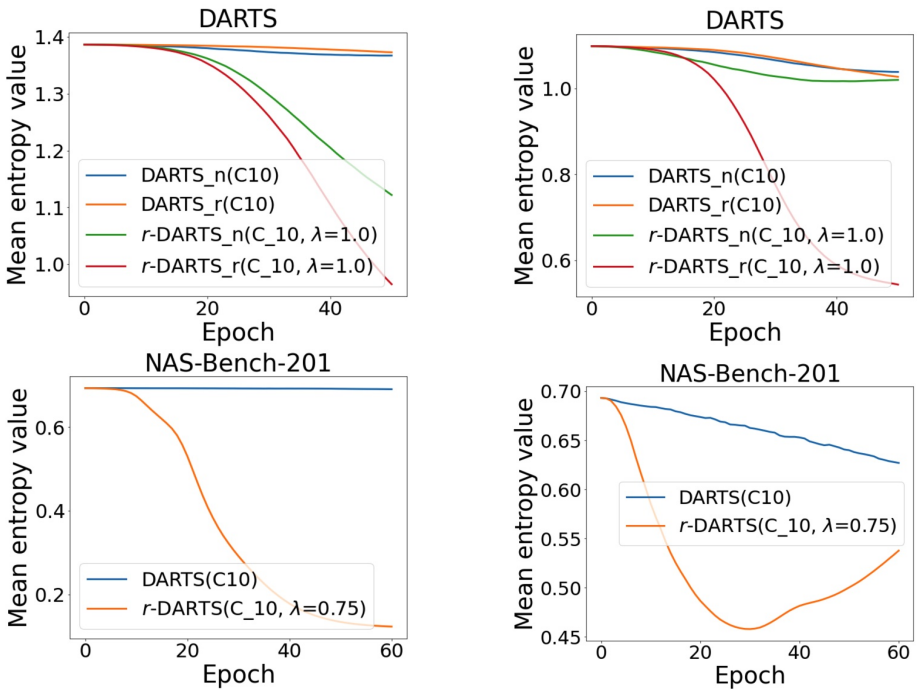
To validate the memory efficiency of r-DARTS, we conducted additional memory experiments on DARTS, and 201 search spaces during the search phase in the CIFAR-10 dataset. As shown in Table 3, compared with its peer competitors (i.e., SNAS, DrNAS), our method achieves the least memory cost on the DARTS search space. We can find that our method is highly efficient, with the same memory costs comparable to DARTS, and almost no significant increase in memory cost. This demonstrates the memory efficiency of our method, which only consumes 2984 bytes of GPU memory. The details are presented in **App. C**.

Search Space	DARTS	SNAS	DrNAS	r-DARTS(ours)
DARTS	8554	9400	10580	8554
NAS-Bench-201	2012	-	949	2012

Table 3: GPU memory comparison.

Effectiveness on MobileNet-like Search Space

To further enhance the effectiveness of our r-DARTS, we consider a wide scope of the state-of-the-art baselines on MobileNet-like search space in the ImageNet1K dataset, and Table 4 provides an empirical comparison between our proposed r-DARTS and peer competitors. We can find that our



(a) Parameter op (b) Parameter-free op

Figure 5: Mean entropy comparison of DARTS and r-DARTS with parameter-free and parameter op across search spaces.

r-DARTS surpasses its competitors in terms of Top-1/5 accuracy, showing superior performance on MobileNet-like search space. Notably, our r-DARTS achieves the highest Top-1 accuracy of 78.2, which is higher than SOTA methods (i.e., GM+ProxylessNAS) 2.6%.

Method	Top-1 (%)	Top-5 (%)	Params (M)
PloxylessNAS(Cai, Zhu, and Han 2018)	75.1	92.8	7.1
FBNet-C(Wu et al. 2019)	74.9	92.1	4.4
FairNAS-A(Chu et al. 2020b)	75.3	7.6	4.6
FairDARTS-D(Chu et al. 2020b)	75.6	92.4	5.3
RLNAS(Zhang et al. 2021b)	75.6	92.6	5.3
GM+ProxylessNAS(Hu et al. 2022)	75.6	93.0	4.9
OLES(Jiang et al. 2023)	75.3	92.4	4.7
ZiCo (Li et al. 2023)	78.1	-	-
AZ-NAS (Lee and Ham 2024)	77.3	93.7	5.8
ParZC (Dong et al. 2025)	77.4	93.9	6.2
r-DARTS(λ=1)	**78.9**	**94.5**	6.4

Table 4: Results on MobileNet-like in ImageNet1K.

Discussion. Surprised by the promising performance of r-DARTS, we give an explanation from the entropy regularization viewpoint. Fig. 5 shows the mean entropy comparison of DARTS and our r-DARTS with parameter-free and parameter operations across various search spaces (i.e., DARTS benchmark, and NAS-Bench-201 benchmark). From Fig. 5, we can find that the overall mean entropy of parameter operations can be degraded significantly by our r-DARTS on both DARTS benchmark and NAS-Bench-201 benchmark. Especially, the parameter-free operations achieve smaller mean entropy than those parameter ones (i.e., 0.544 *v.s.* 0.965) on DARTS benchmark. These results reveal that there indeed exist bifurcation points between compression of α and prediction of Y in DAS, while the unrestricted compression of architecture parameters leads to the performance collapse.

Additional Evaluation

Due to the page limit of the main text, we provide more experimental results in **App. C-H**. 1. **Memory Analysis** is provided in **App. C**. 2. **Effectiveness of Performance Collapse** are presented in **App. D**. 3. **Results on S1-S4 Search Spaces** are presented in **App. E**. 4. **Visualizations of Network Architectures** are presented in **App. F**. 5. **Effectiveness for Downstream Tasks** are presented in **App. G**. 6. **Ablation Studies** are presented in **App. H**.

Conclusion

This paper aims to overcome the performance collapse of DAS from the information bottleneck perspective. The goal is realized by the proposed BER technique. Different from existing work, we use the IB concept to formulate the learning process of DAS, and find that the unrestricted compression leads to the collapse issue. Derived from IB, the proposed BER can well guide the learning of DAS, by means of keeping the entropy value of target architecture parameters within the feasible range. We hope this work inspires further study on defying performance collapse from information-theoretic perspective. In future work, we plan to further reduce search cost of our method.

Acknowledgments

This work is supported by National Natural Science Foundation of China under Grant 62472079, the State Key Program of National Natural Science of China (62432003), Fundamental Research Funds for the Central Universities (N2417003), the Key Technologies R&D Program of Liaoning Province (2023JH1/10400082).

References

Alemi, A. A.; Fischer, I.; Dillon, J. V.; and Murphy, K. 2016. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*.

Baker, B.; Gupta, O.; Naik, N.; and Raskar, R. 2016. Designing neural network architectures using reinforcement learning. *ArXiv*, abs/1611.02167.

Bender, G.; Kindermans, P.-J.; Zoph, B.; Vasudevan, V.; and Le, Q. 2018. Understanding and simplifying one-shot architecture search. In *International Conference on Machine Learning*, 550–559. PMLR.

Brock, A.; Lim, T.; Ritchie, J. M.; and Weston, N. 2017. SMASH: One-Shot Model Architecture Search through HyperNetworks. *ArXiv*, abs/1708.05344.

Cai, H.; Zhu, L.; and Han, S. 2018. Proxylessnas: Direct neural architecture search on target task and hardware. *ArXiv*, abs/1812.00332.

Chen, X.; Xie, L.; Wu, J.; and Tian, Q. 2019. Progressive differentiable architecture search: Bridging the depth gap between search and evaluation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1294–1303.

Chu, X.; Wang, X.; Zhang, B.; Lu, S.; Wei, X.; and Yan, J. 2020a. Darts-: robustly stepping out of performance collapse without indicators. *ArXiv*, abs/2009.01027.

Chu, X.; Zhou, T.; Zhang, B.; and Li, J. 2020b. Fair darts: Eliminating unfair advantages in differentiable architecture search. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV*, 465–480. Springer.

Dai, B.; Zhu, C.; Guo, B.; and Wipf, D. 2018. Compressing neural networks using the variational information bottleneck. In *International Conference on Machine Learning*, 1135–1144. PMLR.

Dong, P.; Li, L.; Tang, Z.; Liu, X.; Wei, Z.; Wang, Q.; and Chu, X. 2025. Parzc: Parametric zero-cost proxies for efficient nas. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 16327–16335.

Dong, X.; and Yang, Y. 2019. Searching for a Robust Neural Architecture in Four GPU Hours. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1761–1770. Los Alamitos, CA, USA.

Gu, Y.-C.; Wang, L.-J.; Liu, Y.; Yang, Y.; Wu, Y.-H.; Lu, S.-P.; and Cheng, M.-M. 2021. Dots: Decoupling operation and topology in differentiable architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12311–12320.

He, H.; Liu, L.; Zhang, H.; and Zheng, N. 2024. IS-DARTS: Stabilizing DARTS through Precise Measurement on Candidate Importance. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 12367–12375.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Howard, A. G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; and Adam, H. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Hu, S.; Wang, R.; Hong, L.; Li, Z.; Hsieh, C.-J.; and Feng, J. 2022. Generalizing few-shot nas with gradient matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Hu, S.; Xie, S.; Zheng, H.; Liu, C.; Shi, J.; Liu, X.; and Lin, D. 2020. DSNAS: Direct Neural Architecture Search Without Parameter Retraining. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 12081–12089. Los Alamitos, CA, USA.

Jiang, S.; Ji, Z.; Zhu, G.; Yuan, C.; and Huang, Y. 2023. Operation-level early stopping for robustifying differentiable NAS. *Advances in Neural Information Processing Systems*, 36.

Jing, K.; Chen, L.; and Xu, J. 2023. An architecture entropy regularizer for differentiable neural architecture search. *Neural Networks*, 158: 111–120.

Lee, J.; and Ham, B. 2024. AZ-NAS: Assembling Zero-Cost Proxies for Network Architecture Search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5893–5903.

Li, G.; Yang, Y.; Bhardwaj, K.; and Marculescu, R. 2023. Zico: Zero-shot nas via inverse coefficient of variation on gradients. *arXiv preprint arXiv:2301.11300*.

Liang, H.; Zhang, S.; Sun, J.; He, X.; Huang, W.; Zhuang, K.; and Li, Z. 2019. Darts+: Improved differentiable architecture search with early stopping. *ArXiv*, abs/1909.06035.

Liu, H.; Simonyan, K.; and Yang, Y. 2018. Darts: Differentiable architecture search. *ArXiv*, abs/1806.09055.

Luo, R.; Tan, X.; Wang, R.; Qin, T.; Chen, E.; and Liu, T.-Y. 2020. Semi-supervised neural architecture search. *Advances in Neural Information Processing Systems*, 33: 10547–10557.

Luo, R.; Tian, F.; Qin, T.; Chen, E.; and Liu, T.-Y. 2018. Neural architecture optimization. *Advances in Neural Information Processing Systems*, 31.

Mei, J.; Li, Y.; Lian, X.; Jin, X.; Yang, L.; Yuille, A.; and Yang, J. 2019. Atomnas: Fine-grained end-to-end neural architecture search. *ArXiv*, abs/1912.09640.

Mellor, J.; Turner, J.; Storkey, A.; and Crowley, E. J. 2021. Neural architecture search without training. In *International Conference on Machine Learning*, 7588–7598. PMLR.

Movahedi, S.; Adabinejad, M.; Imani, A.; Keshavarz, A.; Dehghani, M.; Shakery, A.; and Araabi, B. N. 2023. Λ-DARTS: Mitigating Performance Collapse by Harmonizing

Operation Selection among Cells. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*.

Peng, Y.; Song, A.; Fayek, H. M.; Ciesielski, V.; and Chang, X. 2024. SWAP-NAS: Sample-Wise Activation Patterns for Ultra-fast NAS. In *The Twelfth International Conference on Learning Representations*.

Pham, H.; Guan, M.; Zoph, B.; Le, Q.; and Dean, J. 2018. Efficient neural architecture search via parameters sharing. In *International Conference on Machine Learning*, 4095–4104. PMLR.

Real, E.; Aggarwal, A.; Huang, Y.; and Le, Q. V. 2019. Regularized evolution for image classifier architecture search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 4780–4789.

Ren, P.; Xiao, Y.; Chang, X.; Huang, P.-Y.; Li, Z.; Chen, X.; and Wang, X. 2021. A comprehensive survey of neural architecture search: Challenges and solutions. *ACM Computing Surveys (CSUR)*, 54(4): 1–34.

Shu, Y.; Cai, S.; Dai, Z.; Ooi, B. C.; and Low, B. K. H. 2022. NASI: Label- and Data-agnostic Neural Architecture Search at Initialization. In *International Conference on Learning Representations*.

Sun, Q.; Li, J.; Peng, H.; Wu, J.; Fu, X.; Ji, C.; and Yu, P. S. 2022a. Graph structure learning with variational information bottleneck. In *Proceedings of the AAAI conference on Artificial Intelligence*, volume 36, 4165–4174.

Sun, Z.; Ge, C.; Wang, J.; Lin, M.; Chen, H.; Li, H.; and Sun, X. 2022b. Entropy-driven mixed-precision quantization for deep network design. *Advances in Neural Information Processing Systems*, 35: 21508–21520.

Tishby, N.; Pereira, F. C.; and Bialek, W. 2000. The information bottleneck method. *ArXiv*, abs/0004057.

Tishby, N.; and Zaslavsky, N. 2015. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, 1–5. IEEE.

Wang, R.; Cheng, M.; Chen, X.; Tang, X.; and Hsieh, C.-J. 2021. Rethinking architecture selection in differentiable NAS. *ArXiv*, abs/2108.04392.

Wang, W.; Zhang, X.; Cui, H.; Yin, H.; and Zhang, Y. 2023. FP-DARTS: Fast parallel differentiable neural architecture search for image classification. *Pattern Recognition*, 136: 109193.

Wei, C.; Liang, J.; Liu, D.; and Wang, F. 2022. Contrastive graph structure learning via information bottleneck for recommendation. *Advances in Neural Information Processing Systems*, 35: 20407–20420.

Wu, B.; Dai, X.; Zhang, P.; Wang, Y.; Sun, F.; Wu, Y.; Tian, Y.; Vajda, P.; Jia, Y.; and Keutzer, K. 2019. Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10734–10742.

Wu, J.; Dai, X.; Chen, D.; Chen, Y.; Liu, M.; Yu, Y.; Wang, Z.; Liu, Z.; Chen, M.; and Yuan, L. 2021. Stronger nas with weaker predictors. *Advances in Neural Information Processing Systems*, 34: 28904–28918.

Xie, S.; Zheng, H.; Liu, C.; and Lin, L. 2020. SNAS: Stochastic Neural Architecture Search. *ArXiv*, abs/1812.09926.

Xu, Y.; Xie, L.; Zhang, X.; Chen, X.; Qi, G.-J.; Tian, Q.; and Xiong, H. 2019. Pc-darts: Partial channel connections for memory-efficient architecture search. *ArXiv*, abs/1907.05737.

Ye, P.; Li, B.; Li, Y.; Chen, T.; Fan, J.; and Ouyang, W. 2022. $\backslash beta$-DARTS: Beta-Decay Regularization for Differentiable Architecture Search. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10864–10873. IEEE.

Zela, A.; Elsken, T.; Saikia, T.; Marrakchi, Y.; Brox, T.; and Hutter, F. 2019. Understanding and robustifying differentiable architecture search. *ArXiv*, abs/1909.09656.

Zhang, M.; Su, S. W.; Pan, S.; Chang, X.; Abbasnejad, E. M.; and Haffari, R. 2021a. idarts: Differentiable architecture search with stochastic implicit gradients. In *International Conference on Machine Learning*, 12557–12566. PMLR.

Zhang, X.; Hou, P.; Zhang, X.; and Sun, J. 2021b. Neural architecture search with random labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10907–10916.

Zhou, P.; Xiong, C.; Socher, R.; and Hoi, S. C. H. 2020. Theory-inspired path-regularized differential network architecture search. *Advances in Neural Information Processing Systems*, 33: 8296–8307.

Zoph, B.; and Le, Q. V. 2017. Neural architecture search with reinforcement learning. *5th International Conference on Learning Representations*.

Zoph, B.; Vasudevan, V.; Shlens, J.; and Le, Q. V. 2018. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8697–8710.