SI-VDNAS: Semi-Implicit Variational Dropout for Hierarchical One-shot Neural Architecture Search

Yaoming Wang[1] , **Wenrui Dai**[2*] , **Chenglin Li**[1] , **Junni Zou**[2] and **Hongkai Xiong**[1]
[1]Department of Electronic Engineering, Shanghai Jiao Tong University, China
[2]Department of Computer Science & Engineering, Shanghai Jiao Tong University, China
{wang_yaoming, daiwenrui, lcl1985, zoujunni, xionghongkai}@sjtu.edu.cn

Abstract

Bayesian methods have improved the interpretability and stability of neural architecture search (NAS). In this paper, we propose a novel probabilistic approach, namely **Semi-Implicit Variational Dropout one-shot Neural Architecture Search (SI-VDNAS)**, that leverages semi-implicit variational dropout to support architecture search with variable operations and edges. SI-VDNAS achieves stable training that would not be affected by the over-selection of skip-connect operation. Experimental results demonstrate that SI-VDNAS finds a convergent architecture with only 2.7 MB parameters within 0.8 GPU-days and can achieve 2.60% top-1 error rate on CIFAR-10. The convergent architecture can obtain a top-1 error rate of 16.20% and 25.6% when transferred to CIFAR-100 and ImageNet (mobile setting).

1 Introduction

Deep neural networks (DNNs) have achieved impressive performance in the past decade. However, manually designed DNNs would suffer from unexpected performance loss and tedious design in specific tasks. As an important branch of automatic machine learning (AutoML), neural architecture search (NAS) allows people to get rid of cumbersome design of DNNs. Existing NAS approaches based on evolutionary algorithms or reinforcement learning suffer from excessive calculations and enormous time cost for finding a suitable network structure. Recently, gradient-based approaches have greatly improved the practicality of NAS with a reduced time cost and amount of calculations in search. One-shot NAS is popular in gradient-based approaches, in which each search is evaluated only once to further reduce the time cost. In fact, one-shot NAS transforms the architecture search into a super-net simplification problem to improve the search speed.

One-shot NAS commonly uses the weights of candidate operations [Liu *et al.*, 2019a] as the criterion for selecting final operations and simplifying the super-net. However, gradient descent algorithm has to be carefully designed in these methods, as skip-connect operations tend to be preserved with a large number of epochs and lead to evident degradation of evaluation performance. This problem can be relieved with additional hyperparameters to the gradient [Bi *et al.*, 2019], sigmoid function for selecting operations [Chu *et al.*, 2019] and violent early stopping mechanism [Liang *et al.*, 2019]. These methods improve weight-based selection of operations in NAS, but lack interpretability and stability due to the manually designed hyper-parameters and mechanism.

To improve the interpretability of operation selection in NAS, probabilistic methods have been introduced to determine the final operations [Xie *et al.*, 2019; Zhou *et al.*, 2019; Zheng *et al.*, 2019]. Architectures obtained by sampling instead of the size of parameter is natural and more automated, but the performance is not always satisfactory. Moreover, Gumbel-softmax function is incorporated with weight-based operation selection to enable multiple sampling for preserving variable operations on one edge [Chang *et al.*, 2019].

In this paper, we propose a novel probabilistic approach for hierarchical and stable one-shot neural architecture search, namely Semi-Implicit Variational Dropout one-shot NAS (SI-VDNAS). SI-VDNAS leverages variational dropout based on an annealed semi-implicit automatic relevance detection (ARD) prior to sample the super-net and suppress the over-selection of skip-connect operations. Furthermore, hierarchical structure is developed to separately achieve the selection of edges and operations. The benefits of SI-VDNAS are summarized as below.

- SI-VDNAS alleviates the performance loss led by over-selection of skip-connect operations with semi-implicit variational dropout.
- SI-VDNAS preserves variable operations and edges in NAS to improve the evaluation performance.

2 Preliminaries

In the rest of this paper, for clarity, we use $p(x)$ for the probability density function (PDF) of x and $P(x = x_i)$ for the probability that $x = x_i$.

2.1 One-Shot NAS

One-shot NAS is a sophisticated method, composed of several cells, testing once in super-net without testing all possible structures. In one-shot NAS, a cell is defined as a directed

*Corresponding author

acyclic graph (DAG) of N nodes $n_1, n_2, \cdots, n_N$, where directed edge(i, j) $(i < j)$(from input nodes to intermediate nodes or previous intermediate nodes to current intermediate node) connects the i-th node n_i and j-th node n_j. n_i is processed by the operations $o(\cdot)$ in the search space $\mathcal{O}$. The cell is assumed to have M input nodes $n_I^1, n_I^2, \cdots, n_I^M$ and a single output node n_O. The input nodes are defined as the outputs in the previous M cells and the output of the cell is obtained by applying a concatenation operation to all the $N - M - 1$ intermediate nodes. Recent approaches for one-shot NAS commonly adopt gradient descents with parameter sharing to update each candidate operation in each edge (i, j) in each epoch. Thus, one-shot NAS can keep the search space unchanged and speed up the search, making it possible to search under the mobile settings.

2.2 Variational Dropout

We assume that a neural network has L fully-connected layers, which is consistent with the structure in the super-net of NAS. Let us denote $A \in \mathbb{R}^{M\times I}$, $W \in \mathbb{R}^{I\times O}$ and $B \in \mathbb{R}^{M\times O}$ the input matrix, weight matrix and output matrix for each layer, respectively. Gaussian dropout adds multiplicative noise $\xi \in \mathbb{R}^{M\times I}$ to the input to each layer. Thus, B is obtained by $B = (A \circ \xi) * W$, where $\circ$ indicates the Hadamard product. In Gaussian dropout, the noise ξ is endowed with a Gaussian distribution $\mathcal{N}(1, \delta)$ with its variance $\delta = p/(1 - p)$ determined by the dropout rate p [Srivastava *et al.*, 2014]. Multiplying weight w_{ij} by a Gaussian noise $\xi \sim \mathcal{N}(1, \delta)$ is equivalent to sampling in a new Gaussian distribution $\mathcal{N}(w_{ij}, w_{ij}^2\delta)$. Note that each weight in each layer can be assigned an individual dropout rate δ_{ij}, rather than a global δ. Using the additive reparameterization [Molchanov *et al.*, 2017], we can substitute $\theta_{ij} = w_{ij}\delta ij$ with a low-variance form $\theta_{ij} = w_{ij}(1 + \sqrt{\delta ij}\epsilon_{ij})$ with $\epsilon_{ij} \sim \mathcal{N}(0, 1)$. This additive reparameterization can remove samples from the calculation graph to allow gradient back propagation.

2.3 Semi-implicit Distribution

Semi-implicit distribution $p(x)$ is defined based on an implicit distribution $p(z)$ and a conditional distribution with explicit PDF $p(x|z)$.

$$p(x) = \int p(x|z)p(z)dz \tag{1}$$

Here, $p(z)$ does not have explicit distribution, but its expectation can be estimated based on sampling. Given N samples from $p(z)$, the semi-implicit distribution $p(x)$ is estimated by

$$p(x) = \int p(x|z)p(z)dz \approx \frac{1}{N}\sum_{n=1}^{N} p(x|z^n), \tag{2}$$

where z^n represents the n-th sample extracted from $p(z)$.

3 Methodology

In this section, we elaborate the proposed SI-VDNAS, in which a hierarchical structure for NAS is developed based on the semi-implicit variational dropout with an annealed ARD prior. Semi-implicit variational dropout is able to obtain structures with varying numbers of operations on one edge. In addition to operation selection, the hierarchical structure enables separate selection of varying numbers of edges to improve the interpretability and stability of network search.

3.1 Semi-implicit Variational Dropout for NAS

SI-VDNAS leverages semi-implicit variational dropout to achieve NAS with an individual dropout rate for each operation, rather than a shared hyperparameter for all operations. The dropout noise is supposed to obey the semi-implicit distribution $q(\xi) = \int q(\xi|\psi)q(\psi)d\psi$, where $q(\xi|\psi) = \mathcal{N}(\xi|\psi, \psi\delta)$ and $q(\psi)$ is a Bernoulli distribution parameterized with δ, i.e., $P(\psi = 1) = 1/(1 + \delta)$ and $P(\psi = 0) = \delta/(1 + \delta)$. Note that δ is also introduced in the variance of $q(\xi|\psi)$. Thus, the semi-implicit distribution $q_\delta(\xi)$ is formulated with regard to δ.

$$q_\delta(\xi) = \int q_\delta(\xi|\psi)q_\delta(\psi)d\psi \tag{3}$$

In one-shot NAS, the dropout noise $\xi \sim q_\delta(\xi)$ is assigned to each candidate operation $o(x)$ on x. Let us define $\tau_{i,j}^o = \gamma_{i,j}^o \cdot \xi_{i,j}^o$ for the weight $\gamma_{i,j}^o$ of the operation o in the edge (i, j). For simplicity, we omit the superscripts and subscripts in the rest of this subsection. According to Eq. (3), $\tau \sim q_\delta(\tau) = \int q_\delta(\tau|\psi)q_\delta(\psi)d\psi$ with $q_\delta(\tau|\psi) = \mathcal{N}(\tau|\gamma\psi, \gamma^2\psi\delta)$ for each $\tau = \gamma \cdot \xi$. Due to the discrete Bernoulli distribution $q_\delta(\psi)$, the integral for the semi-implicit distribution $q_\delta(\tau)$ is intractable. Thus, we leverage the Monte Carlo sampling to calculate $q_\delta(\tau)$ from K samples $\psi^1, \cdots, \psi^K$ from $q_\delta(\psi)$.

$$q_\delta(\tau) = \int q_\delta(\tau|\psi)q_\delta(\psi)d\psi \approx \frac{1}{K}\sum_{k=1}^{K} q_\delta\left(\tau|\psi^k\right) \tag{4}$$

Here, we set $K = 1$ to make stochastic update based on semi-implicit variational dropout. The Evidence Lower BOund (ELBO) $\mathcal{L}(\delta)$ is developed for semi-implicit variational dropout by taking $q_\delta(\psi)$ as the posterior distribution.

$$\begin{aligned}\mathcal{L}(\delta) &= L_D(\delta) - D_{KL}\left(q_\delta(\tau)\|p(\tau)\right)\\ L_D(\delta) &= \sum_{n=1}^{N}\mathbb{E}_{q_\delta(\tau)}\left[\log p\left(y_n|x_n, \tau\right)\right]\end{aligned} \tag{5}$$

In Eq. (5), $L_D(\delta)$ is empirically obtained based on the training set that consists of N pairs of observations and labels $\{x_n, y_n\}$, $n = 1, \cdots, N$. Consequently, KL divergence $D_{KL}(q_\delta(\tau)\|p(\tau))$ is minimized for operation selection under the properly designed prior $p(\tau)$, as shown in Section. 3.2.

3.2 Annealed Semi-implicit ARD Prior

We further design a semi-implicit prior $p_{\lambda,\delta}(\tau)$ for semi-implicit variational dropout.

$$p_{\lambda,\delta}(\tau) = \int p_\lambda(\tau|\phi)p_\delta(\phi)d\phi \approx \frac{1}{N}\sum_{n=1}^{N} p_\lambda\left(\tau|\phi^n\right), \tag{6}$$

where $p_\delta(\phi)$ is also a Bernoulli distribution parameterized with δ and $p_\lambda(\tau|\phi) = \mathcal{N}(\tau|\lambda\gamma\phi, \eta^{-1}\phi)$ is a Gaussian PDF

associated with the variance η^{-1} of ARD prior and the annealed temperature λ. From Eq. (4) and (6), we formulate the KL divergence $D_{KL}\left(q_\delta(\tau)\|p(\tau)\right)$ for minimization.

$$D_{KL}\left(q_\delta(\tau)\|p_{\lambda,\delta}(\tau)\right) \approx \int \frac{1}{K}\sum_{k=1}^{K} q_\delta\left(\tau|\psi^k\right)\cdot \left[\log\frac{1}{K}\sum_{k=1}^{K} q_\delta\left(\tau|\psi^k\right) - \log\frac{1}{N}\sum_{n=1}^{N} p_\lambda\left(\tau|\phi^n\right)\right]d\tau \quad (7)$$

When we set $K = N = 1$ for stochastic update, we have

$$D_{KL}\left(q_\delta(\tau)\|p_{\lambda,\delta}(\tau)\right) \approx \mathbb{E}_{q_\delta(\tau|\psi^*)}\log\frac{q_\delta\left(\tau|\psi^*\right)}{p_\lambda\left(\tau|\phi^*\right)} \quad (8)$$

Here, we use Φ to represent ψ^* and ϕ^*, as they are taken from the same Bernoulli distribution. Thus, we have $D_{KL}(q_\delta(\tau)\|p_{\lambda,\delta}(\tau)) \approx \mathbb{E}_{q_\delta(\tau|\Phi)}\log(q_\delta(\tau|\Phi)/p_\delta(\tau|\Phi))$.

Proposition 1. *The proposed annealed semi-implicit ARD prior $p_{\lambda,\delta}(\tau)$ generalizes the ARD prior $\mathcal{N}(\tau|0,\eta^{-1})$ to induce the KL divergence $D_{KL}(q_\delta(\tau)\|p_{\lambda,\delta}(\tau))$ that are independent of the weight parameters γ.*

Proof. Given Φ sampled from the Bernoulli distribution $q_\delta(\phi)$, we have $q_\delta(\tau) \approx q_\delta(\tau|\Phi) = \mathcal{N}(\tau|\gamma\Phi, \gamma^2\Phi\delta)$ and $p_{\lambda,\delta}(\tau) \approx p_\lambda(\tau|\Phi) = \mathcal{N}(\tau|\lambda\gamma\Phi, \eta^{-1}\Phi)$. When $\Phi = 0$, $q_\delta(\tau)$ and $p_\delta(\tau)$ degenerate to the constant 0, which means their KL divergence equals to 0. For $\Phi = 1$, the KL divergence between $q_\delta(\tau)$ and $p_{\lambda,\delta}(\tau)$ is approximated by

$$\begin{aligned} &D_{KL}(q_\delta(\tau|\Phi)||p_\lambda(\tau|\Phi)) \\ &\quad= D_{KL}(\mathcal{N}(\tau|\gamma\Phi,\gamma^2\Phi\delta)||\mathcal{N}(\tau|\lambda\gamma\Phi,\eta^{-1}\Phi)) \\ &\quad= -\frac{1}{2}\log\eta\gamma^2\delta + 2\eta\gamma^2[(1-\lambda)^2+\delta] - \frac{1}{2} \end{aligned} \quad (9)$$

The optimal value $\eta^* = \arg\min_\eta D_{KL}(q_\delta(\tau|\Phi)||p_\lambda(\tau|\Phi))$. From the gradient of $D_{KL}(q_\delta(\tau|\Phi)||p_\lambda(\tau|\Phi))$, we have

$$\eta^* = \frac{1}{\gamma^2[(1-\lambda)^2+\delta]} \quad (10)$$

Thus, we can approximate $D_{KL}(q_\delta(\tau)\|p_{\lambda,\delta}(\tau))$ by

$$D_{KL}\left(q_\delta(\tau)\|p_{\lambda,\delta}(\tau)\right) \approx \frac{1}{2}\log[1+(1-\lambda)^2\delta^{-1}] \quad (11)$$

This KL divergence does not depend on weight parameters γ. Note that $p_{\lambda,\delta}(\tau)$ degenerates to the vanilla ARD prior $\mathcal{N}(\tau|0,\eta^{-1})$, when $\lambda = 0$ and $\Phi = 1$. The KL divergence degenerates to $0.5\cdot\log(1+\delta^{-1})$, which is adopted for the hierarchical structure in [Liu *et al.*, 2019b]. □

3.3 Hierarchical Structure

Edge selection is implicit in previous one-shot approaches, in which only operation weight parameters are used, resulting ambiguous edge selection and poor evaluation performance. Edge normalization introduces the edge weight parameters to assist operation weight parameters in searching the final structure [Xu *et al.*, 2019]. It can restrict the search of candidate edges between nodes, but cannot separately achieve the selection of edges and operations. In SI-VDNAS, we assign individual dropout rates to the edge weight parameters as well as operation weight parameters, separating the edge selection from the operation selection and achieving architecture with variable edges.

Let us denote $\tilde{f}_{i,j}(x)$ the output of mixed operations in edge (i,j) and $\tilde{h}_j(x)$ the mixed edge on the node j. The edge weight parameters φ and the operation weight parameters w are referred as the architecture parameters γ. Considering vanilla variational dropout, for the operation weight parameters, we assign the dropout rate $\xi^o_{i,j} \sim \mathcal{N}(\xi^o_{i,j}|1,\delta^o_{i,j})$ to the input $op^o_{i,j}(x)$ and obtain:

$$\tilde{f}_{i,j}(x) = \sum_{o\in\mathcal{O}} w^o_{i,j} op^o_{i,j}(x)\cdot\xi^o_{i,j} = \sum_{o\in\mathcal{O}} w^o_{i,j}\xi^o_{i,j}\cdot op^o_{i,j}(x) \quad (12)$$

$\tilde{f}_{i,j}(x)$ can be calculated using samples from $\mathcal{N}(\tilde{f}_{i,j}(x)|w^o_{i,j}op^o_{i,j}, (w^o_{i,j}op^o_{i,j})^2\delta^o_{i,j})$.

For the edge weight parameters, we assign the dropout rate $\zeta_{i,j} \sim \mathcal{N}(\zeta_{i,j}|1,\sigma_{i,j})$ to $\tilde{f}_{i,j}(x)$. $\tilde{h}_j(x)$ can be similarly sampled from $\mathcal{N}(\tilde{h}_j(x)|\varphi_{i,j}\tilde{f}_{i,j}(x), (\varphi_{i,j}\tilde{f}_{i,j}(x))^2\sigma_{i,j})$.

$$\begin{aligned} \tilde{h}_j(x) &= \sum_{i<j}\varphi_{i,j}\tilde{f}_{i,j}(x) * \zeta_{i,j} = \sum_{i<j}\varphi_{i,j}\zeta_{i,j} * \tilde{f}_{i,j}(x) \\ &= \sum_{i<j}\varphi_{i,j}\zeta_{i,j}\cdot\sum_{o\in\mathcal{O}} w^o_{i,j}\xi^o_{i,j}\cdot op^o(x) \end{aligned} \quad (13)$$

Without loss of generality, we consider arbitrary one edge (i,j). Denote $\mu^o = w^o\cdot\xi^o \sim \mathcal{N}(\mu^o|w^o,(w^o)^2\delta^o)$ and $\upsilon = \varphi\cdot\zeta \sim \mathcal{N}(\upsilon|\varphi,\varphi^2\sigma)$. The KL divergence can be developed for the approximate posterior distribution $q(\upsilon,\mu) = q(\upsilon|\mu)\prod_{o\in\mathcal{O}} q(\mu^o)$ and the hierarchical prior distribution $p(\upsilon,\mu) = p(\upsilon|\mu)\prod_{o\in\mathcal{O}} p(\mu^o)$ under independent μ and υ.

$$\begin{aligned} D_{KL}(q(\upsilon,\mu)||p(\upsilon,\mu)) &= D_{KL}(q(\upsilon|\mu)||p(\upsilon|\mu)) \\ &\quad+\sum_{o\in\mathcal{O}} D_{KL}(q(\mu^o)||p(\mu^o)) \end{aligned} \quad (14)$$

Consequently, we introduce semi-implicit variational dropout. The posterior distribution $q(\upsilon|\mu)$ is approximated by

$$\begin{aligned} q(\upsilon|\mu) &= \int q(\upsilon|\mu,\psi_e)q(\psi_e)d\psi_e \approx q(\upsilon|\mu,\Phi_e) \\ &= \mathcal{N}(\upsilon|\varphi\Phi_e,\varphi^2\Phi_e\sigma) \end{aligned} \quad (15)$$

where Φ_e is the sample from the Bernoulli distribution $q(\psi_e)$. The posterior $q(\mu^o)$ is similarly approximated with Φ^o_o sampled from the Bernoulli distribution $q(\psi^o_o)$.

$$\begin{aligned} q(\mu^o) &= \int q(\mu^o|\psi^o_o)q(\psi^o_o)d\psi^o_o \approx q(\mu^o|\Phi^o_o) \\ &= \mathcal{N}(\mu^o|w^o\Phi^o_o,(w^o)^2\Phi^o_o\delta^o) \end{aligned} \quad (16)$$

Subsequently, we adopt the annealed semi-implicit ARD prior to formulate the KL divergence for training.

$$\begin{aligned} &D_{KL}(q(\upsilon|\mu)||p(\upsilon|\mu) \approx D_{KL}(q(\upsilon|\mu,\Phi_e)||p(\upsilon|\mu,\Phi_e)) \\ &= \begin{cases} 0, & \Phi_e = 0 \\ 0.5\cdot\log[1+(1-\lambda_e)^2\sigma^{-1}], & \Phi_e = 1 \end{cases} \end{aligned} \quad (17)$$

Algorithm 1 Semi-implicit Variational Dropout NAS

Input: Data $\{x_n, y_n\}_{1:N}$, network parameter p, architecture parameter $(w, \varphi, \delta, \sigma)$: operation weight parameter w, edge weight parameter φ, dropout rate δ for operation and σ for edge, operation Bernoulli variable ψ_o and edge Bernoulli variable ψ_e, dropout noise $\xi \sim \mathcal{N}(\xi|\psi_o, \psi_o\delta)$ for w, the variational posterior distribution $\mathcal{N}(\mu|w\psi_o, w^2\psi_o\delta)$, dropout noise $\zeta \sim \mathcal{N}(\zeta|\psi_e, \psi_e\sigma)$ for φ, the variational posterior distribution $\mathcal{N}(\upsilon|\varphi\psi_e, \varphi^2\psi_e\sigma)$, prior distribution $p_{\lambda_o,\delta}(\mu)$ with the annealed temperature λ_o for operation and $p_{\lambda_e,\sigma}(\upsilon|\mu)$ with the annealed temperature λ_e for edge.

Output: The individual dropout rate δ for operation and σ for edge, the operation weight parameter w and the edge weight parameter φ, the sample of ψ_o and ψ_e.

Initialize w, φ, δ and σ.
while not converged **do**
  Sample Φ_o and Φ_e from ψ_o and ψ_e respectively.
  Approximate the semi-implicit posterior distribution as $q(\mu) \approx q(\mu|\Phi_o)$, $q(\upsilon|\mu) \approx q(\upsilon|\mu, \Phi_e)$.
  Approximate the semi-implicit prior as $p(\mu) \approx p(\mu|\Phi_o)$, $p(\upsilon|\mu) \approx p(\upsilon|\mu, \Phi_e)$.
  Calculate $D_{KL}(q(\upsilon|\mu, \Phi_e)\|p(\upsilon|\mu, \Phi_e))$ using Eq. (17) and $D_{KL}(q(\mu|\Phi_o)\|p(\mu|\Phi_o))$ using Eq. (18).
  Update w, φ, δ, and σ by gradient descent.
  Update network parameters p by gradient descent .
  $(1-\lambda_o^{new})^2 = 0.95 \cdot (1-\lambda_o^{old})^2$.
  $(1-\lambda_e^{new})^2 = 0.95 \cdot (1-\lambda_e^{old})^2$.
end while
return w, φ, δ, and σ and final samples Φ_o, Φ_e from ψ_o and ψ_e respectively.

$$D_{KL}(q(\mu^o)||p(\mu^o) \approx D_{KL}(q(\mu^o|\Phi_o^o)||p(\mu^o|\Phi_o^o)) = \begin{cases} 0, & \Phi_o^o = 0 \\ 0.5 \cdot \log[1 + (1-\lambda_o^o)^2(\delta^o)^{-1}], & \Phi_o^o = 1 \end{cases} \quad (18)$$

From Eq. (17) and (18), we obtain the objective KL divergence in Eq. (14). Algorithm 1 elaborates the implementation of SI-VDNAS.

4 Experiments

Following the pipeline of DARTS, our experiments can be divided into 3 stages. In the first stage, we search the architecture on CIFAR-10 with a simple network. The optimized architecture in search is stored and stacked to generate a complex network for new training processes from scratch. The new training processes consist of two stages, i.e., evaluation on CIFAR-10/100 and ImageNet.

4.1 Datasets

CIFAR-10/100 [Krizhevsky and Hinton, 2009] is a popular dataset consisting of 60K images, 50K training images and 10K test images. All these images share the same spatial resolution of 32×32, and are categorized into 10/100 classes.

ImageNet [Deng *et al.*, 2009] is a large-scale benchmark for image classification. It contains 1.3M training images and 50K test images that are equally distributed into 1000 classes. As in [Zoph *et al.*, 2018; Xie *et al.*, 2019; Liu *et al.*, 2019a], we adopt the *mobile setting* with a spatial resolution of 224×224 for input images to limit FLOPS by 600M during testing.

4.2 Architecture Search

Implementation Details

Following DARTS, the super-net is formed by stacking two kinds of basic cells. Specifically, 6 normal cells and 2 reduction cells are stacked to form the super-net. Each cell contains seven nodes, including two input nodes, four intermediate nodes, and one output node. The output of the four intermediate nodes are concatenated as the input for the output node. Each cell has 14 candidate edges, where a hybrid operation consisting of 7 candidate operations is assigned to each candidate edge.

For fair comparison, we adopt the same candidate operations as existing one-shot NAS, except for 'none' operation. We assign dropout rate to operation weight parameters w and edge weight parameters ϕ, respectively. So, we have two stages of our object function. In each stage, we use the semi-implicit distribution $q_\delta(\xi) = \int q_\delta(\xi|\psi)q_\delta(\psi)d\psi$ as the approximate posterior. The implicit distribution $q_\delta(\psi)$ is chosen to be the Bernoulli distribution parameterized by the dropout rate δ. Thus, we can sample the super-net by sampling from $q_\delta(\psi)$. The final structure is also determined by the samples of $q_\delta(\psi)$. The first part of semi-implicit distribution is the Gaussian distribution which is the same as that of VDNAS when $\psi = 1$. The prior distribution is chosen to be the annealed semi-implicit ARD distribution, which is the generalized version of ARD prior used in previous work [Liu *et al.*, 2019b]. For our training, we set the annealed temperature as $(1-\lambda^{new})^2 = 0.95 \cdot (1-\lambda^{old})^2$, and the second part of prior shares the same sample of $q_\delta(\psi)$.

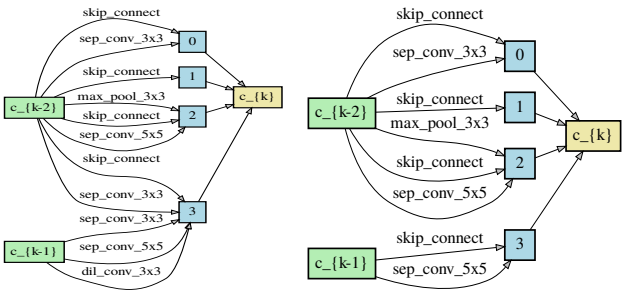
We utilize the bi-level update algorithm to update the architectural parameters and conventional network weights, respectively. For the training, we split the training images into two subsets with the same size. One subset is used for training network parameters, the other is used for architectural parameters. We can not only train the network for 50 epochs following DARTS to get the optimal structure, but also train 150 epochs for convergence(the number of epochs does not have to be 150, but it can also be 100 or 300. We can search to get a convergent result without the need to carefully design the search process for preventing degeneration), with the initial number of channels being 16. Following [Chen *et al.*, 2019], we freeze architectural parameters and only update the network parameters in first 15 epochs. The batch-size is set to 64 to enable the searching process on single GPU.

Search Results

The search process requires 8 GPU-hours for optimal structure within 50 epochs and 20 GPU-hours for a convergent result within 150 epochs on a single NVIDIA GTX 1080Ti GPU. The search time can be reduced by about 50% on a single Tesla V100 GPU. Figure 1 shows the optimal structure within 50 epochs and the convergent structure within 150 epochs. Due to space limitations, we only show the searched normal cells, as the number of reduction cells is small.

Architecture	Top-1 (Test) Error (%)		Params	Search Cost	Search Method
	CIFAR-10	CIFAR-100	(M)	(GPU-days)	
DenseNet-BC [Huang *et al.*, 2017]	3.46	17.18	25.6	-	manual
AmoebaNet-A + cutout [Real *et al.*, 2019]	3.34 ± 0.06	-	3.2	3150	evaluation
AmoebaNet-B + cutout [Real *et al.*, 2019]	2.55 ± 0.05	-	2.8	3150	evaluation
NASNet-A + cutout [Zoph *et al.*, 2018]	2.65	-	3.3	1800	RL
ENAS + cutout [Pham *et al.*, 2018]	2.89	-	4.6	0.5	RL
PNAS [Liu *et al.*, 2018]	3.41 ± 0.09	-	3.2	225	SMBO
NAONet-WS [Luo *et al.*, 2018]	3.53	-	3.1	0.4	NAO
MdeNAS [Zheng *et al.*, 2019]	2.55	-	3.6	0.16	MDL
DARTS (1st order) + cutout [Liu *et al.*, 2019a]	3.00 ± 0.14	17.76†	3.3	0.4	gradient-based
DARTS (2nd order) + cutout [Liu *et al.*, 2019a]	2.76 ± 0.09	17.54†	3.3	1	gradient-based
SNAS (mild) + cutout [Xie *et al.*, 2019]	2.98	-	2.9	1.5	gradient-based
SNAS (moderate) + cutout [Xie *et al.*, 2019]	2.85 ± 0.02	-	2.8	1.5	gradient-based
SNAS (aggressive) + cutout [Xie *et al.*, 2019]	3.10 ± 0.04	-	2.3	1.5	gradient-based
PC-DARTS + cutout [Xu *et al.*, 2019]	2.57 ± 0.07	-	3.6	0.1	gradient-based
P-DARTS + cutout [Chen *et al.*, 2019]	2.50	16.55†	3.4	0.3	gradient-based
BayesNAS + cutout [Zhou *et al.*, 2019]	2.81 ± 0.04	-	3.4	0.2	gradient-based
DARTS-EGS (M = 4) [Chang *et al.*, 2019]	3.01	-	2.6	1	gradient-based
DARTS-EGS (M = 7) [Chang *et al.*, 2019]	2.79	-	2.9	1	gradient-based
Amended-DARTS,S1 + cutout [Bi *et al.*, 2019]	2.81 ± 0.21	-	3.5	1.0	gradient-based
Amended-DARTS,S2 + cutout [Bi *et al.*, 2019]	2.60 ± 0.15	-	3.6	1.1	gradient-based
SI-VDNAS(base) + cutout	2.50 ± 0.06	15.98	3.6	0.3	gradient-based
SI-VDNAS(convergence) + cutout	2.60 ± 0.05	16.20	2.7	0.8	gradient-based

Table 1: Comparison with state-of-the-art NAS methods for image classification on CIFAR-10/100. For each method, top-1 test error (%), number of parameters (M) and search cost (GPU-days) are evaluated. Here, lower error rate stands for better performance and † indicates that the experiments are conducted by P-DARTS.



(a) Base normal cell (b) Convergent normal cell

Figure 1: Searched normal cells on CIFAR-10.

4.3 Architecture Evaluation

Evaluation on CIFAR-10/100

Evaluation on CIFAR-10/100 follows that of DARTS and P-DARTS [Chen *et al.*, 2019]. The network is composed of 20 cells (18 normal cells and 2 reduction cells). Each cell has the same architecture learned in our searching stage. The initial number of channels is set to 36. The network weights are trained from scratch using all the 50K training images with a batch size of 96. The network is trained for 600 epochs. We use the SGD optimizer with an initial learning rate of 0.025 (annealed down to zero following a cosine schedule without restart), a momentum of 0.9, a weight decay of $3 \times 10^{-4}/5 \times 10^{-4}$ and a norm gradient clipping at 5. We apply the drop-path trick with the probability of 0.3. Cutout is also used in our evaluation. Table 1 summarizes the results obtained by the state-of-the-arts and SI-VDNAS on CIFAR-10/100. SI-VDNAS outperforms all the benchmarks in terms of test accuracy. It can achieve 2.50% top-1 error rate with 3.6M parameters in 0.3 GPU-days. The convergent architecture of SI-VDNAS also achieves 2.60% top-1 error rate and 2.7M parameters in 0.8 GPU-days.

Evaluation on ImageNet

We further evaluate the searched architectures from CIFAR-10 on ImageNet to validate its generalizability. The evaluation process is similar to DARTS. The input images share the resolution of 224×224 for *Mobile Setting*. Three convolution layers of stride 2 are first employed to reduce the sizes of input images from 224×224 to 28×28. The network consists of 12 normal cells and 2 reduction cells. The initial number of channels is set to 48. The network is trained from scratch for 250 epochs using a batch size of 1024. We use the SGD optimizer with a momentum of 0.9, an initial learning rate of 0.5 (decayed down to zero linearly) and a weight decay of 3×10^{-5}. We also adopt label smoothing and an auxiliary loss tower during training. Learning rate warm-up is applied for the first 5 epochs.

Table 2 provides the evaluation results. The effectiveness of transferring SI-VDNAS is demonstrated. The basic architecture searched on CIFAR-10 can achieve 25.3% top-1 error rate and 8.0% top-5 error rate with 5.0 MB parameters and 577M FLOPS. The convergent architecture searched on

Architecture	Test Error (%)		Params	FLOPS	Search Cost	Search Method
	Top-1	Top-5	(M)	(M)	(GPU-days)	
Inception-v1 [Szegedy *et al.*, 2015]	30.2	10.1	6.6	1448	-	manual
MobileNet [Howard *et al.*, 2017]	29.4	10.5	4.2	569	-	manual
ShuffleNet 2× (v2) [Ma *et al.*, 2018]	25.1	-	~ 5	591	-	manual
AmoebaNet-C [Real *et al.*, 2019]	24.3	7.6	6.4	570	3150	evaluation
NASNet-A [Zoph *et al.*, 2018]	26.0	8.4	5.3	564	1800	RL
PNAS[Liu *et al.*, 2018]	25.8	8.1	5.1	588	225	SMBO
MdeNAS [Zheng *et al.*, 2019]	25.5	7.9	6.1	-	0.16	MDL
DARTS (2nd order) [Liu *et al.*, 2019a]	26.7	8.7	4.7	574	1	gradient-based
SNAS (mild) [Xie *et al.*, 2019]	27.3	9.2	4.3	522	1.5	gradient-based
PC-DARTS(CIFAR-10) [Xu *et al.*, 2019]	25.1	7.8	5.3	586	0.1	gradient-based
PC-DARTS† [Xu *et al.*, 2019]	24.2	7.3	5.3	597	3.8	gradient-based
P-DARTS [Chen *et al.*, 2019]	24.4	7.4	4.9	557	0.3	gradient-based
BayesNAS [Zhou *et al.*, 2019]	26.5	8.9	3.9	-	0.2	gradient-based
DARTS-EGS (M = 4) [Chang *et al.*, 2019]	25.7	8.5	4.3	-	1.5	gradient-based
DARTS-EGS (M = 7) [Chang *et al.*, 2019]	24.9	8.1	4.7	-	1.5	gradient-based
DARTS+† [Liang *et al.*, 2019]	23.9	7.4	5.1	582	6.8	gradient-based
FairDARTS-A [Chu *et al.*, 2019]	26.3	8.2	3.6	417	0.4	gradient-based
FairDARTS-B [Chu *et al.*, 2019]	24.9	7.5	4.8	541	0.4	gradient-based
Amended-DARTS, S2 [Bi *et al.*, 2019]	24.3	7.4	5.5	590	1.1	gradient-based
SI-VDNAS(base)	25.3	8.0	5.0	577	0.3	gradient-based
SI-VDNAS(convergence)	25.6	8.1	4.1	462	0.8	gradient-based

Table 2: Comparison with state-of-the-art NAS methods for image classification on ImageNet. For each method, top-1 and top-5 test errors (%), number of parameters (M), FLOPS (M) and search cost (GPU-days) are evaluated. Here, lower error rate stands for better performance and † indicates that the structure is directly searched on ImageNet.

CIFAR-10 can achieve 25.6% top-1 error rate and 8.2% top-5 error rate with only 4.1 MB parameters and 462M flops, which outperforms DARTS (26.7% top-1 error rate and 8.7% top-5 error rate). SI-VDNAS also reduces 1%-2% top-1 and top-5 error rate in comparison to SNAS. Note that Amended-DARTS and P-DARTS adopt deeper search space (more than 8 layers stacked as the super-net) than DARTS. Furthermore, PC-DARTS and DARTS+ directly performs NAS on ImageNet. Structures achieved by these approaches are usually deeper and naturally perform better on large dataset consisting of high-resolution images. However, these approaches actually change the search space and are time consuming.

4.4 Ablation Study

To verify the design of SI-VDNAS, we further evaluate the strategies excluding the semi-implicit distribution from SI-VDNAS. Two search strategies, namely H-VDNAS and V-VDNAS, are adopted, where V-VDNAS utilizes only vanilla variational dropout and H-VDNAS considers hierarchical structure over V-VDNAS. Table 3 shows the top-1 error rates on CIFAR-10/100 obtained by SI-VDNAS, V-VDNAS, H-VDNAS and DARTS, respectively. These results imply that both variational dropout and hierarchical structure contribute to the performance gain by SI-VDNAS. We also depict the searched normal cells by V-VDNAS and H-VDNAS in Figure 2. Moreover, H-VDNAS and V-VDNAS are also affected by the over-selection of skip-connect operations with a large number of epochs, as semi-implicit variational dropout is not adopted to preserve variable operations. Note that it is not desirable to directly separate the hierarchical structure from SI-VDNAS, as multiple operations are preserved in one edge.

Architecture	SI	HS	Top-1 (Test) Error (%)		Params
			CIFAR-10	CIFAR-100	(M)
SI-VDNAS	✓	✓	2.50±0.06	15.98	3.6
H-VDNAS	×	✓	2.54±0.04	16.68	3.7
V-VDNAS	×	×	2.62±0.08	16.74	3.3
DARTS(2nd)	-	-	2.76±0.09	17.54	3.3

Table 3: Comparison with DARTS, V-VDNAS and H-VDNAS for image classification on CIFAR-10/100 with 50 search epochs. For each method, top-1 test error (%) and number of parameters (M) are evaluated. SI and HS indicate the semi-implicit variation dropout and hierarchical structure proposed by SI-VDNAS. Here, lower error rate stands for better performance.

5 Related Work

In this section, we briefly introduce previous works paying attention to NAS. Evolutionary algorithms were adopted in [Elsken *et al.*, 2019; Miikkulainen *et al.*, 2019; Real *et al.*, 2017; Real *et al.*, 2019] to evolve one single network or a family of networks towards better performance. Reinforcement learning (RL) based methods [Zoph and Le, 2017; Zoph *et al.*, 2018; Bender *et al.*, 2018; Pham *et al.*, 2018] utilized a meta-controller to guide the search process in the huge space of architecture by optimizing the reward function for the inference accuracy of the selected network. To narrow the search space, ENAS [Pham *et al.*, 2018] stacked repeated cells to form the final structure, which was also adopted in

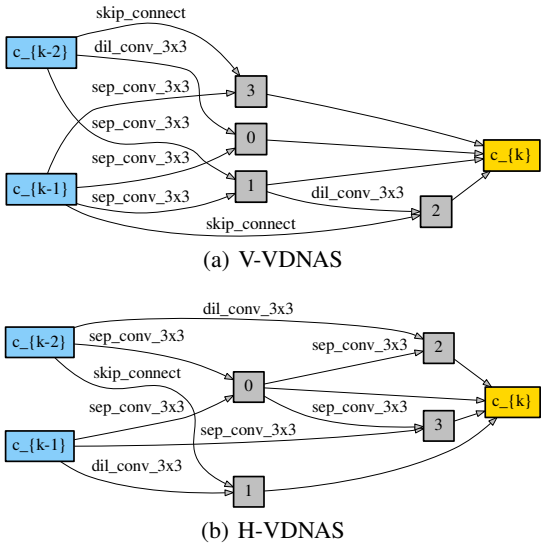


Figure 2: Base normal cells searched on CIFAR-10 by V-VDNAS and H-VDNAS, respectively.

gradient-based methods such as DARTS.

Despite the good performance, a large amount of search time and consumption of computing resources limit the development of evolution based methods and reinforcement based methods. Recently, Gradient based methods, especially one-shot models, have gradually become the mainstream method of NAS. Among them, DARTS [Liu *et al.*, 2019a] proposed a differentiable framework by introducing architectural parameters to measure the importance of candidate operations. Many works are then based on DARTS. SNAS [Xie *et al.*, 2019] introduced the concrete distribution to NAS and replaced softmax function in DARTS with Gumbel-softmax function. PC-DARTS proposed partial channel connection to reduce the memory cost in NAS and accelerate the search process. P-DARTS gradually improved the width and depth of search space to bridge the gap between search and evaluation. Recently, DARTS is discovered to degenerate when searched until convergence. DARTS+ [Liang *et al.*, 2019] introduced early stopping into NAS, Fair-DARTS [Chu *et al.*, 2019] replaced the softmax function with sigmoid function, Amended-DARTS added the hyperparameter to th gradient of DARTS. Too many artifacts and lack of interpretability are common to these methods, our proposed SI-VDNAS leverages variational dropout based on an annealed semi-implicit ARD prior, solving the degeneration and also improving the evaluation performance by enlarging the evaluation space.

6 Conclusion

In this paper, we proposed a probabilistic NAS approach, named Semi-Implicit Variational Dropout Neural Architecture Search (SI-VDNAS). The core idea of SI-VDNAS is to use the semi-implicit variational dropout and annealed semi-implicit ARD prior replace the vanilla variational dropout and ARD prior, hierarchical structure is also used to separate the edge selection and operation selection. SI-VDNAS can solve the degeneration occurred in previous one-shot NAS sharing the search space of DARTS. SI-VDNAS can also improve the evaluation performance by preserving variable operations and variable edge. In terms of performance, SI-VDNAS can approach the state-of-art result in CIFAR-10/100 and outperforms the benchmark when transformed to ImageNet.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grants 61971285, 61720106001, 61932022, 61931023, 61972256, 61871267 and 91838303, and in part by the Program of Shanghai Academic Research Leader under Grant 17XD1401900.

References

[Bender *et al.*, 2018] Gabriel Bender, Pieter-Jan Kindermans, Barret Zoph, Vijay Vasudevan, and Quoc Le. Understanding and simplifying one-shot architecture search. In *Proceedings of the 35th International Conference on Machine Learning*, pages 549–558, Stockholm, Sweden, July 2018.

[Bi *et al.*, 2019] Kaifeng Bi, Changping Hu, Lingxi Xie, Xin Chen, Longhui Wei, and Qi Tian. Stabilizing DARTS with amended gradient estimation on architectural parameters. *arXiv preprint arXiv:1910.11831*, 2019.

[Chang *et al.*, 2019] Jianlong Chang, Xinbang Zhang, Yiwen Guo, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. Differentiable architecture search with ensemble gumbel-softmax. *arXiv preprint arXiv:1905.01786*, 2019.

[Chen *et al.*, 2019] Xin Chen, Lingxi Xie, Jun Wu, and Qi Tian. Progressive differentiable architecture search: Bridging the depth gap between search and evaluation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1294–1303, Seoul, Korea, October 2019.

[Chu *et al.*, 2019] Xiangxiang Chu, Tianbao Zhou, Bo Zhang, and Jixiang Li. Fair DARTS: Eliminating unfair advantages in differentiable architecture search. *arXiv preprint arXiv:1911.12126*, 2019.

[Deng *et al.*, 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, Miami, FL, USA, June 2009.

[Elsken *et al.*, 2019] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Efficient multi-objective neural architecture search via Lamarckian evolution. In *7th International Conference on Learning Representations*, New Orleans, LA, USA, May 2019.

[Howard *et al.*, 2017] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[Huang *et al.*, 2017] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*, pages 4700–4708, Honolulu, HI, USA, July 2017.

[Krizhevsky and Hinton, 2009] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[Liang *et al.*, 2019] Hanwen Liang, Shifeng Zhang, Jiacheng Sun, Xingqiu He, Weiran Huang, Kechen Zhuang, and Zhenguo Li. Darts+: Improved differentiable architecture search with early stopping. *arXiv preprint arXiv:1909.06035*, 2019.

[Liu *et al.*, 2018] Chenxi Liu, Barret Zoph, Maxim Neumann, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy. Progressive neural architecture search. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 19–34, Munich, Germany, September 2018.

[Liu *et al.*, 2019a] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. In *7th International Conference on Learning Representations*, New Orleans, LA, USA, May 2019.

[Liu *et al.*, 2019b] Yuhang Liu, Wenyong Dong, Lei Zhang, Dong Gong, and Qinfeng Shi. Variational Bayesian dropout with a hierarchical prior. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7124–7133, Long Beach, CA, USA, June 2019.

[Luo *et al.*, 2018] Renqian Luo, Fei Tian, Tao Qin, Enhong Chen, and Tie-Yan Liu. Neural architecture optimization. In *Advances in Neural Information Processing Systems 31*, pages 7816–7827, Montreal, QC, USA, December 2018.

[Ma *et al.*, 2018] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. ShuffleNet V2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 116–131, Munich, Germany, September 2018.

[Miikkulainen *et al.*, 2019] Risto Miikkulainen, Jason Liang, Elliot Meyerson, Aditya Rawal, Daniel Fink, Olivier Francon, Bala Raju, Hormoz Shahrzad, Arshak Navruzyan, Nigel Duffy, and Babak Hodjat. Evolving deep neural networks. In *Artificial Intelligence in the Age of Neural Networks and Brain Computing*, pages 293–312. Academic Press, 2019.

[Molchanov *et al.*, 2017] Dmitry Molchanov, Arsenii Ashukha, and Dmitry Vetrov. Variational dropout sparsifies deep neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 2498–2507, Sydeny, NSW, Australia, August 2017.

[Pham *et al.*, 2018] Hieu Pham, Melody Y. Guan, Barret Zoph, Quoc V. Le, and Jeff Dean. Efficient neural architecture search via parameter sharing. In *Proceedings of the 35th International Conference on Machine Learning*, pages 4095–4104, Stockholm, Sweden, July 2018.

[Real *et al.*, 2017] Esteban Real, Sherry Moore, Andrew Selle, Saurabh Saxena, Yutaka Leon Suematsu, Jie Tan, Quoc V. Le, and Alexey Kurakin. Large-scale evolution of image classifiers. In *Proceedings of the 34th International Conference on Machine Learning*, pages 2902–2911, Sydney, NSW, Australia, August 2017.

[Real *et al.*, 2019] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V. Le. Regularized evolution for image classifier architecture search. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, pages 4780–4789, Honolulu, HI, USA, January 2019.

[Srivastava *et al.*, 2014] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[Szegedy *et al.*, 2015] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, Boston, MA, USA, June 2015.

[Xie *et al.*, 2019] Sirui Xie, Hehui Zheng, Chunxiao Liu, and Liang Lin. SNAS: stochastic neural architecture search. In *7th International Conference on Learning Representations*, New Orleans, LA, USA, May 2019.

[Xu *et al.*, 2019] Yuhui Xu, Lingxi Xie, Xiaopeng Zhang, Xin Chen, Guo-Jun Qi, Qi Tian, and Hongkai Xiong. Pc-darts: Partial channel connections for memory-efficient architecture search. In *International Conference on Learning Representations*, 2019.

[Zheng *et al.*, 2019] Xiawu Zheng, Rongrong Ji, Lang Tang, Baochang Zhang, Jianzhuang Liu, and Qi Tian. Multinomial distribution learning for effective neural architecture search. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1304–1313, Seoul, Korea, October 2019.

[Zhou *et al.*, 2019] Hongpeng Zhou, Minghao Yang, Jun Wang, and Wei Pan. BayesNAS: A Bayesian approach for neural architecture search. In *Proceedings of the 36th International Conference on Machine Learning*, pages 7603–7613, Long Beach, CA, USA, June 2019.

[Zoph and Le, 2017] Barret Zoph and Quoc V. Le. Neural architecture search with reinforcement learning. In *5th International Conference on Learning Representations*, Toulon, France, April 2017.

[Zoph *et al.*, 2018] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. Learning transferable architectures for scalable image recognition. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8697–8710, Salt Lake City, UT, USA, June 2018.