Prioritized Architecture Sampling with Monto-Carlo Tree Search

Xiu Su[1*], Tao Huang[2*], Yanxi Li[1], Shan You[2,3†], Fei Wang[2], Chen Qian[2], Changshui Zhang[3], Chang Xu[1†]

[1]School of Computer Science, Faculty of Engineering, The University of Sydney, Australia
[2]SenseTime Research [3]Department of Automation, Tsinghua University
[*]Equal contributions [†]Corresponding authors
{xisu5992,yali0722}@uni.sydney.edu.au {huangtao,youshan,wangfei,qianchen}@sensetime.com
zcs@mail.tsinghua.edu.cn c.xu@sydney.edu.au

Motivation

One-shot NAS: Based on the weight-sharing paradigm, One-shot NAS methods model NAS as a one-shot training process of an over-parameterized supernet, where various architectures can be directly derived.

Single Path Methods:

1. Iteratively train the paths (architectures) in the supernet.
2. Search architectures then return the one with the best performance.

Issues:

- Current methods select each operation independently without considering previous layers.
- The historical information obtained with huge computation cost is usually used only once and then discarded.
- The search cost is high since it usually searches a large number (e.g., 1000) of architectures for a good result.

Intuition

Modeling the search space as a Monte-Carlo tree (MCT), which can naturally

- capture the dependency among layers with a tree structure;
- store intermediate results for future decision and a better exploration-exploitation balance;
- bridge the training and search by searching on the MCT constructed in training.

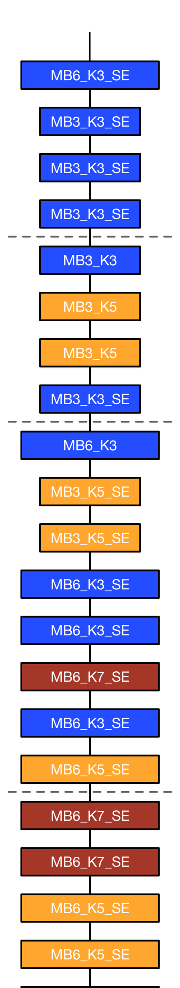
Problems:

1. Q: How to reward the operations in MCT?
   A: Use the training loss $\mathcal{L}_{tr}$ as the Q-value in UCT function.
2. Q: It's impossible to explore all the nodes since the number of nodes grows exponentially with the increment of depth.
   A: 1. We propose a node communication technique to share the rewards for nodes with the same operation and depth.
   2. We propose a hierarchical node selection method to select the node hierarchically and re-evaluate those less-visited nodes.

Experimental Settings

ImageNet:

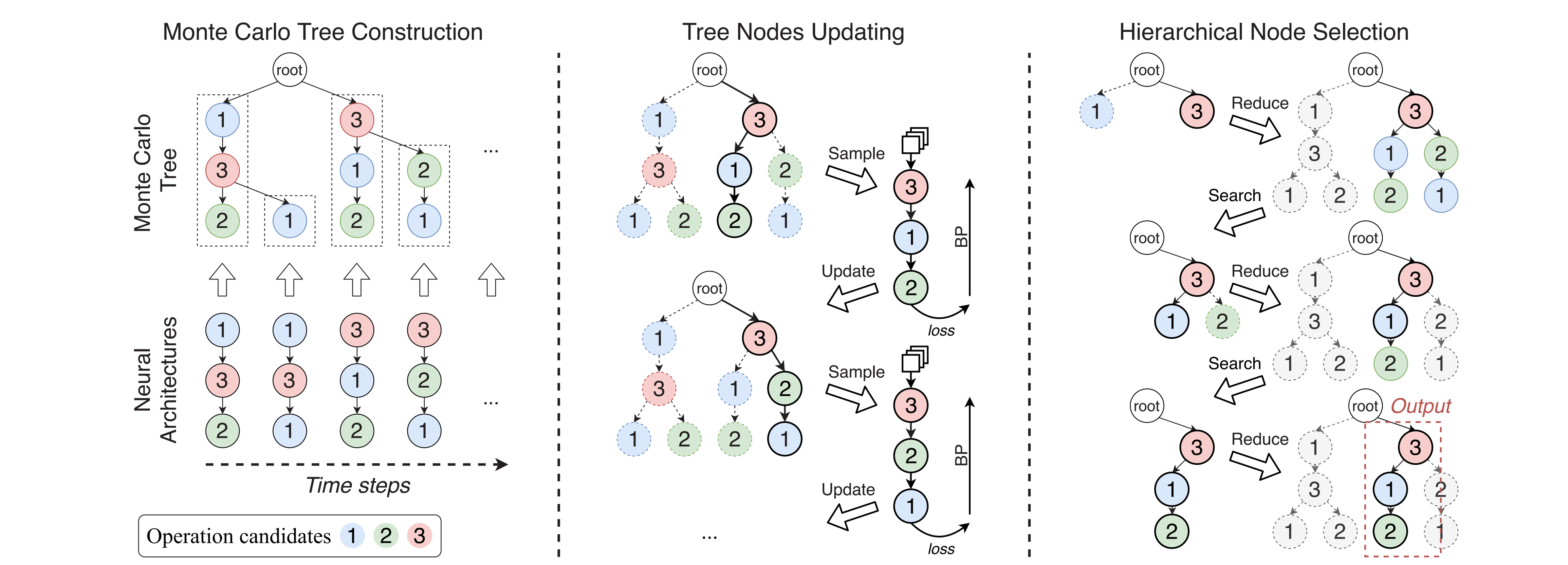
- Search space: MobileNetV2 inverted bottleneck with CNN kernel {3,5,7}, expansion ratio {3,6} and optional SE module. Size 13^{21} with identity.
- Supernet: train 60 epochs using uniform sampling for warm-up, 60 epochs with MCTS
- Search: 20 architectures in MCT
- Retraining: following Mnasnet.



CIFAR-10:

- Search space: MobileNetV2 inverted bottleneck with kernel size {3, 5} and expansion ratio {3, 6} Size 3^8 with identity.
- Supernet: train 100 epochs using uniform sampling for warm-up, 100 epochs with MCTS
- Search: 20 architectures in MCT

Framework of MCT-NAS



MCT-NAS models the search space into a MCT (left), then updates the tree with a prioritized sampling strategy during training (middle), finally searches the optimal architecture using hierarchical node selection (right).

Training with Prioritized Sampling

We use the training loss as the Q-value in UCT function, calculated as

$$Q(v_i^{(l)}) = \frac{\widetilde{\mathcal{L}}_t}{\mathcal{L}_{tr}(\boldsymbol{\alpha}_t)},$$

where $\widetilde{\mathcal{L}}_t$ denotes the training loss of the current architecture, $\boldsymbol{\alpha}_t$ is the moving average of training loss in previous t iterations.

The UCT function for the node $v_i^{(l)}$ in layer l with choice i is calculated by

$$\mathrm{UCT}(v_i^{(l)}) = \frac{Q(v_i^{(l)})}{n_i^{(l)}} + C_1\sqrt{\frac{\log(n_p^{(l-1)})}{n_i^{(l)}}},$$

where $n_p^{(l-1)}$ and $n_i^{(l)}$ denotes the visit times of parent node and this node, respectively.

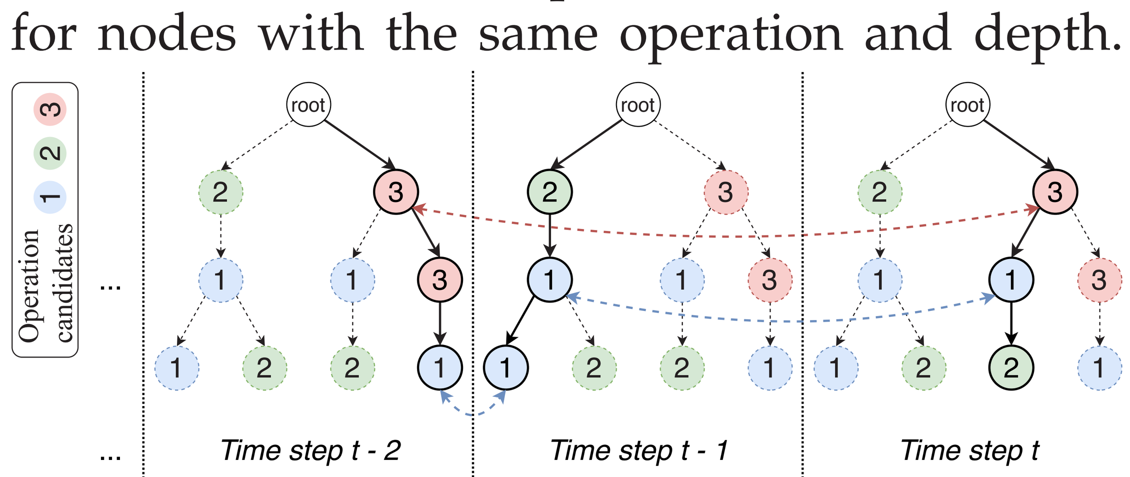
To make more nodes evaluated, we relax the operation selection in MCTS into a probabilistic distribution, formulated as

$$P_t(v_i^{(l)}) = \frac{\exp\left(\mathrm{UCT}(v_i^{(l)})/\tau\right)}{\sum_{j\le N^l}\exp\left(\mathrm{UCT}(v_j^{(l)})/\tau\right)}, \quad (1)$$

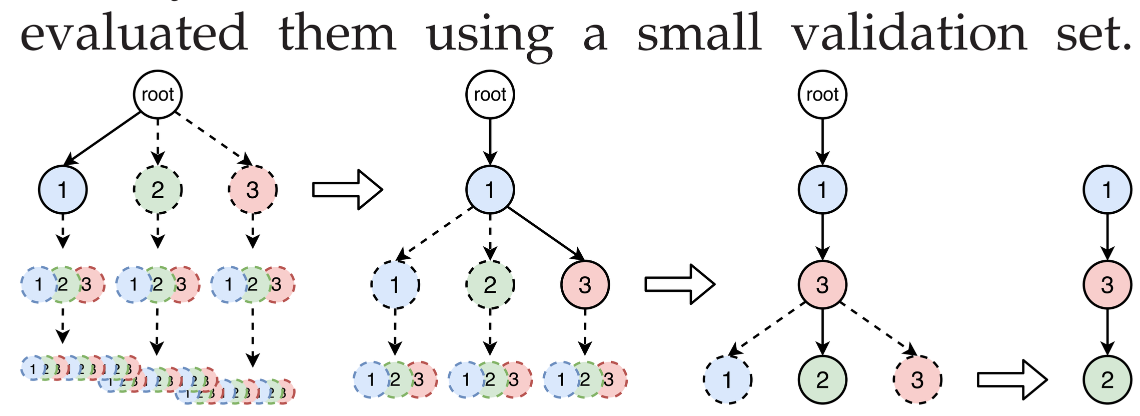
where τ is a temperature term. We set τ to 0.0025 in all of our experiments.

Node Communication and Hierarchical Node Selection

In supernet training: We propose a node communication technique to share the rewards for nodes with the same operation and depth.



In search: We propose a hierarchical node selection method to select the node hierarchically; for those less-visited nodes, we re-evaluated them using a small validation set.



NAS-Bench-Macro

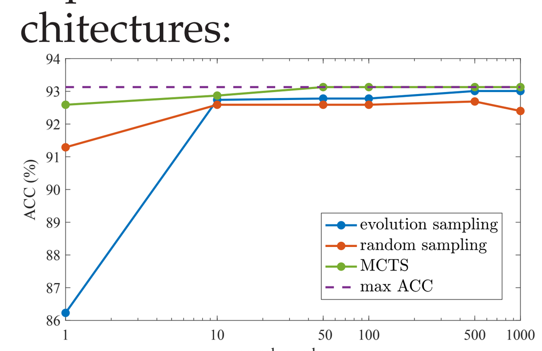
We propose a NAS benchmark on macro structures with CIFAR-10 dataset. The benchmark is avaliable at `https://github.com/xiusu/NAS-Bench-Macro`.

Our MCT-NAS can obtain better supernet with higher ranking correlation:

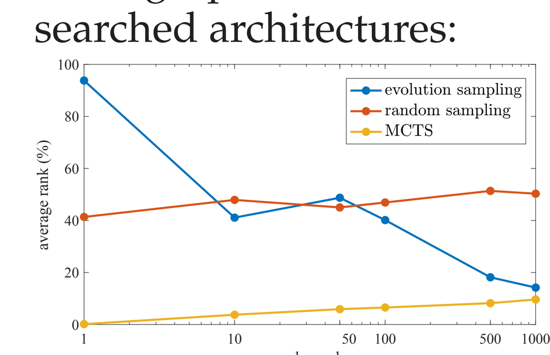
Methods	Spearman rho	Kendall tau
uniform	88.96%	72.41%
MCTS	90.63%	74.66%
uniform + MCTS	91.87%	76.22%

Our MCT-NAS can search better architectures with fewer search number:

Top ACCs of searched architectures:

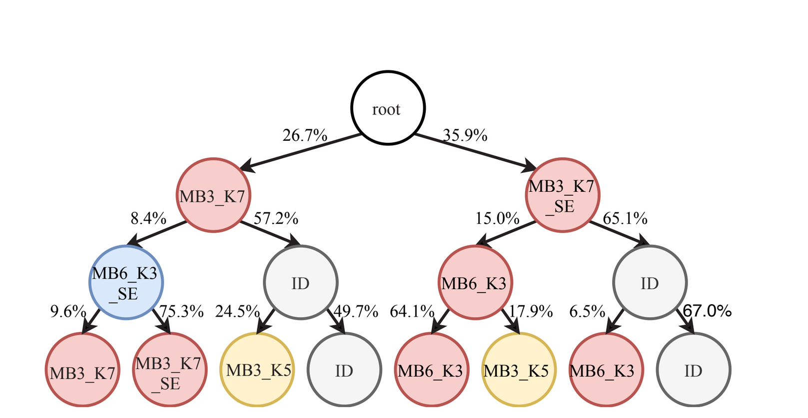


Average percentile rank of searched architectures:



Comparison with State-of-the-art NAS Methods on ImageNet

Methods	Top-1 (%)	FLOPs (M)	Params (M)	training (Gdays)	search number
SCARLET-C	75.6	280	6.0	10	8400
GreedyNAS-C	76.2	284	4.7	7	1000
MCT-NAS-C	76.3	280	4.9	12	20 × 5
Single-path	76.2	328	-	12	1000
ST-NAS-A	76.4	326	5.2	-	990
GreedyNAS-B	76.8	324	5.2	7	1000
MCT-NAS-B	76.9	327	6.3	12	20 × 5
EfficientNet-B0	76.3	390	5.3	-	-
ST-NAS-B	77.9	503	7.8	-	990
MCT-NAS-A	78.0	442	8.4	12	20 × 5



Visualization of first 3 layers of searched MCT.