BossNAS: Exploring Hybrid CNN-transformers with Block-wisely Self-supervised Neural Architecture Search

Supplementary Material

Changlin Li[1], Tao Tang[2], Guangrun Wang[3,4], Jiefeng Peng[3], Bing Wang[5], Xiaodan Liang[2]*, Xiaojun Chang[6]

[1]GORSE Lab, Dept. of DSAI, Monash University [2]Sun Yat-sen University [3]DarkMatter AI Research

[4]University of Oxford [5]Alibaba Group [6]RMIT University

changlin.li@monash.edu,

{trent.tangtao,wanggrun,jiefengpeng,xdliang328}@gmail.com,

fengquan.wb@alibaba-inc.com, xiaojun.chang@rmit.edu.au

A. Appendix

A.1. A brief review of NAS

NAS methods aim to automatically optimize neural network architectures by exploring search spaces with *search algorithms* and evaluating architectures by means of *rating schemes*. NAS methods can be divided into two categories depending on the rating scheme utilized, *i.e.* multi-trial NAS and weight-sharing NAS. **Multi-trial NAS** methods [92, 2, 54, 63, 42, 83] rate all sampled architectures by training them from scratch, making this process computationally prohibitive and difficult to deploy on large datasets. They either perform architecture rating by training on relatively small datasets (*e.g.* CIFAR-10) [92, 2, 54] or by training for the first few epochs (*e.g.* 5 epochs) [63] on ImageNet. To avoid repeated training of candidate networks, **weight-sharing NAS** methods [7, 43, 18, 1, 6, 13, 82] optimize a *supernet* that encodes the whole search space, then rate each candidate architecture according to its weights inherited from the supernet. Among them, *gradient-based* approaches [43, 7, 70] and *sampler-based* approaches [52, 61] jointly optimize the weight of the supernet and the factors (or agent) used to choose the architecture; for their part, *one-shot* approaches[1] [22, 15, 6, 4, 51] optimize the supernet before performing a search with the frozen supernet weights. We refer to [55] for a more comprehensive NAS review.

*Corresponding Author.

[1]In this paper, following the pioneering works SMASH [6] and One-shot [4], when we refer to one-shot NAS methods, we are discussing those incorporating two-stage (i.e., a supernet training stage and a searching stage) weight-sharing methods rather than the general weight-sharing NAS discussed in [80].

A.2. Implementation Details

Search spaces. We evaluate our method on three search spaces:

- **HyTra search space.** The beginning of the networks in this search space is the classic ResNet stem that reduces the spatial resolution by a factor of 4 with a strided 7×7 convolution layer and a max-pooling layer. It contains $L = 16$ choice block layers in total, as the same to ResNet50. Before the first choice block layer, the input can be further down-sampled to different scales. The downsampling module consists of multiple 3×3 convolutions with stride of 2. At each choice block layer, the spatial resolution can either stay unchanged or be reduced to half of its scale, unless reaching the smallest scale $1/32$. As introduced in Sec. 4, this search space contains two disparate candidate choices: $\{\texttt{ResConv}, \texttt{ResAtt}\}$. As transformer blocks are expensive in the first scales, we only enable the choice of **`ResAtt`** in the last two scales (*i.e.* $1/16$ and $1/32$). The total size of this challenging hybrid search space is roughly 2.8×10^6.

- **MBConv search space.** MobileNet-like search space and its variations are generally used as benchmarks for recent NAS methods [63, 29, 64, 7, 70, 15, 37, 46, 87]. Following Li *et. al.* [37], we use a search space with 18 layers and each layer contains 4 candidate MobileNet blocks (combination of kernel size $\{3, 5\}$ and reduction rate $\{3, 6\}$). This results in a large search space containing about $4^{18} \approx 6.9\times10^{10}$ architectures.

- **NATS-Bench $\mathcal{S}_S$.** The NATS-Bench *size* search space $\mathcal{S}_S$ [17] is a channel configuration search space built

upon a fixed cell-based architecture with 5 layers, where the 2-nd and 4-th layers have a down-sample rate of 2. Number of channels in each layer is chosen from $\{8, 16, 24, 32, 40, 48, 56, 64\}$. $\mathcal{S}_S$ has $8^5 = 32768$ architecture candidates in total. Candidates of different channel numbers in our supernet share the weights in a slimmable manner [78, 77, 76, 38, 9]. We divide the supernet into 3 blocks, according to spatial size.

Datasets. The datasets we use to evaluate and analyze our method include ImageNet [16], CIFAR-10 and CIFAR-100 [36]. **ImageNet** is a large-scale dataset containing 1.2 M `train` set images and 50 K `val` set images in 1000 classes. We randomly samples 50 K images from the original `train` set to form a `NAS-val` set for architecture rating and use the remainder as the `NAS-train` set for supernet training. No labels are used during training and searching of our NAS method. Finally, our searched architectures are retrained from scratch on `train` set and evaluated on `val` set. For **CIFAR-10** and **CIFAR-100** [36], we use the splits proposed in NATS-Bench [17]. CIFAR-10 is divided into 25 K `train` set, 25 K `val` set, and 10 K `test` set. CIFAR-100 is devided into 50 K `train` set, 5 K `val` set, and 5 K `test` set. The final accuracies of searched architectures are queried from NATS-Bench $\mathcal{S}_S$ [17].

Training details.

We train each block of the **BossNAS supernet** for 20 epochs including 1 linear warm-up epoch on ImageNet. For the relatively smaller CIFAR datasets, we extend it to 30 epochs. In each training step, we randomly sample 4 paths for the ensemble bootstrapping. Other hyperparameters for self-supervised training of the supernet follow closely to BYOL [21], we use the `LARS` optimizer [75] with a cosine decay learning rate schedule [44]. The base learning rate is set to 4.8 for a total batchsize of 4096.

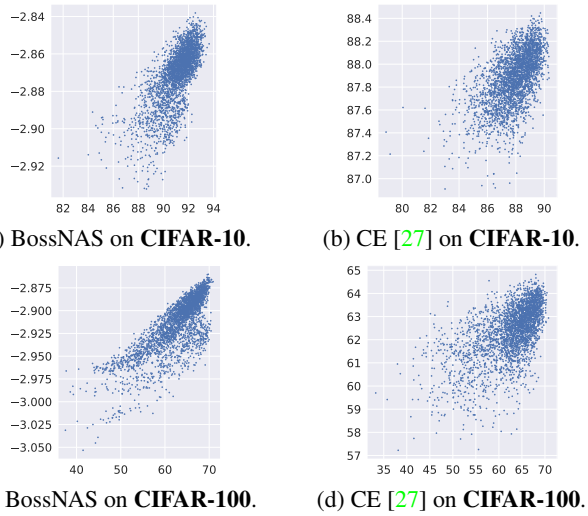
For ImageNet retraining of **BossNet-T models**, we follow similar with DeiT [66], as we found it robust for both CNNs and transformers. More specifically, we use AdamW optimizer with 1e-3 initial learning rate and cosine learning rate scheduler, for a total batch size of 1024. Weight decay is set to 0.05. We use model EMA with decay rate 0.99996 following [79]. Please refer to DeiT [66] for more details on data-augmentation and regularization.

For ImageNet retraining of **BossNet-M models**, we follow closely to EfficientNet [64]. We use batchsize 4096, RMSprop optimizer with momentum 0.9 and initial learning rate of 0.256 which decays by 0.97 every 2.4 epochs. Please refer to EfficientNet [64] for more details of other settings.

Re-implementation of other NAS methods on HyTra.

For DNA [37], we use ResNet-50 [25] as the teacher model. We divide the supernet into four blocks, with four layers in each block, and train each block for 20 epochs. The intermediate features of every block of the student supernet and the teacher are all downsampled with global pooling and projected with one fully-connected layer before calculating distillation loss, as the scale of different candidate block is not the same in HyTra search space. Other settings follow closely to DNA [37].

For UnNAS [41], we adopt *rotation prediction* [35] (`Rot`) pretext task, for its simplicity. Following [41], we use three extra stride-2 convolution layers at the beginning of the supernet to reduce spatial resolution. The supernet is trained for 2 epochs as in [41].



(a) BossNAS on **CIFAR-10**. (b) CE [27] on **CIFAR-10**.

(c) BossNAS on **CIFAR-100**. (d) CE [27] on **CIFAR-100**.

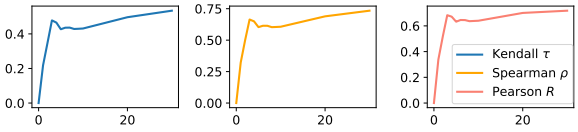
Figure 8: Comparison of architecture rating and its true accuracy of our BossNAS and CE [27] on **NATS-Bench $\mathcal{S}_S$** with **CIFAR datasets**.

Dataset	Method	τ	ρ	R
CIFAR-10	CE [27]	0.42	0.60	0.59
	BossNAS	**0.53**	**0.73**	**0.72**
CIFAR-100	CE [27]	0.43	0.60	0.60
	BossNAS	**0.59**	**0.76**	**0.79**

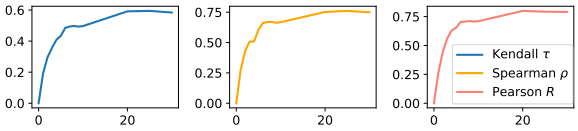
Table 6: Architecture rating accuracy on **NATS-Bench $\mathcal{S}_S$** with **CIFAR datasets**.

A.3. Additional Analysis on NATS-Bench $\mathcal{S}_S$

Architecture rating comparison. We compare with the predictor-based NAS method CE [27] by architecture rating accuracy on CIFAR-10 and CIFAR-100. As shown in Fig. 8, we compare the two NAS methods by plotting the correlation of the architecture rating and the true accuracy of 3000 randomly sampled architectures from NATS-Bench *size* search space $\mathcal{S}_S$ [17]. Architectures with BossNAS form denser and more spindly scatter pattern than CE on both of the two datasets. Moreover, as measured quantitatively in Tab. 6, BossNAS outperforms CE by a large margin (**0.11** and **0.16** τ) in both datasets.

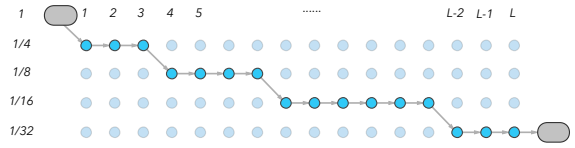


(a) Ranking correlations during supernet training on **CIFAR-10**.

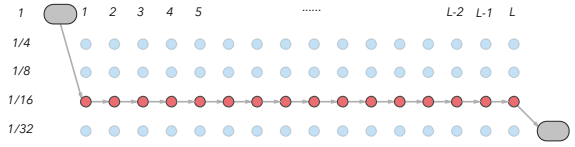


(b) Ranking correlations during supernet training with **CIFAR-100**.

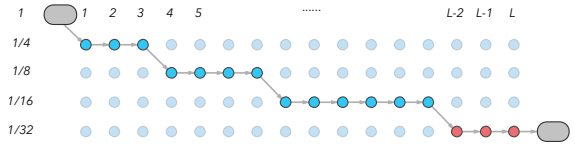
Figure 9: Convergence behavior of BossNAS on **NATS-Bench $\mathcal{S}_S$** and **CIFAR datasets**.



(a) Architecture of ResNet50-T.



(b) Architecture of ViT-T/16.



(c) Architecture of BoTNet-T.

Figure 10: Visualization of Human-designed Architectures in HyTra. Blue nodes denotes `ResConv` and red nodes denotes `ResAtt`.

Convergence Behavior. We illustrate the architecture rating accuracy of BossNAS during its 30 epoch supernet training phase on CIFAR datasets in Fig. 9. The architecture rating accuracy increases quickly and steadily with minor fluctuations, in a similar manner with that on MBConv search space (Fig. 7). In particular, architecture rating accuracy of our BossNAS converges to a satisfactory result, **0.76** ρ, smoothly and quickly within only 20 epochs on CIFAR-100, and continues to be stable for the subsequent 10 epochs.

A.4. Visualization of Human-designed Architectures in HyTra

The architectures of ResNet50-T, ViT-T/16 and BoTNet50-T from our HyTra search space are illustrated in Fig. 10. Their architectures follow as closely as possible to the architectures of their prototypes.