

APPENDIX: A SURGERY OF THE NEURAL ARCHITECTURE EVALUATORS

Anonymous authors

Paper under double-blind review

A ADDITIONAL RESULTS ABOUT PREDICTOR-BASED EVALUATORS

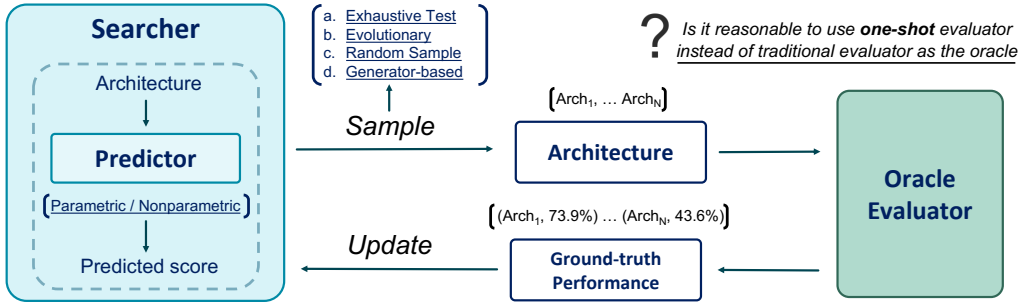


Figure 1: The overview of predictor-based neural architecture search (NAS). The underlined descriptions between the parenthesis denote different methods.

A.1 PREDICTOR PERFORMANCES RESULTS: SINGLE AND MULTIPLE STAGES

Table 1: The Kendall’s Tau of different predictors on 3 different randomly sampled training dataset of size 78

Training Loss	Ranking			Regression		
Dataset	1	2	3	1	2	3
MLP	0.1330±0.074	0.1560±0.0078	0.2481±0.0069	0.0111±0.0000	0.0548±0.0276	0.0467±0.0130
LSTM	0.5631±0.0060	0.6028±0.0457	0.5487±0.0150	0.6024±0.0039	0.5784±0.0180	0.4656±0.0176
GATES (Ning et al., 2020)	0.7597±0.0079	0.7750±0.0106	0.7645±0.0054	0.2067±0.0000	0.7240±0.0074	0.7135±0.0055
RF (Sun et al., 2019)	-	-	-	0.4329±0.0077	0.4123±0.0104	0.4218±0.0119

Table 2: The performance distribution, BR@K, Kendall’s Tau of 5 training stages. In each stage, $K = 78$ architectures are chosen, evaluated, and used to train the predictor along with previous architecture data. Note that in this table, K in BR@K is the absolute architecture number without normalization

	Stage	0	1	2	3	4
GATES	Perf. Range	[0.560, 0.938]	[0.921, 0.944]	[0.935, 0.944]	[0.933, 0.944]	[0.933, 0.944]
	Perf. Std	6.43e-2	4.59e-3	2.16e-3	2.18e-3	2.30e-3
	BR@11/BR@7/BR@1	1/2/306	1/1/3	1/1/2	1/1/3	1/1/3
	Kendall’s Tau	0.769	0.759	0.752	0.742	0.725
	Stage	0	1	2	3	4
LSTM	Perf. Range	[0.560, 0.938]	[0.922, 0.944]	[0.922, 0.944]	[0.932, 0.944]	[0.934, 0.944]
	Perf. Std	6.43e-2	4.52e-3	2.63e-3	2.42e-3	1.98e-3
	BR@11/BR@7/BR@1	99/268/393	2/2/9	1/1/6	1/2/5	1/1/3
	Kendall’s Tau	0.562	0.556	0.571	0.739	0.724

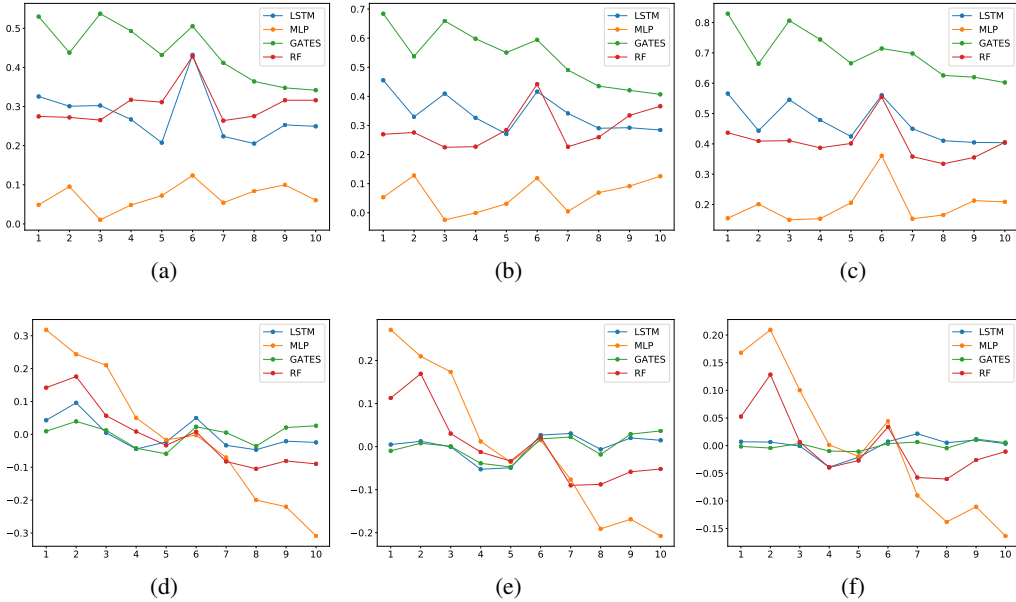


Figure 2: (a)(b)(c) Kendall-tau in different FLOPs groups, the training set size is 39, 78 and 390, respectively. (d)(e)(f) Average rank difference in different FLOPs groups, the training set size is 39, 78 and 390, respectively.

A.2 OVER- AND UNDER-ESTIMATION OF ARCHITECTURES

Fig. 2(d)(e)(f) illustrates the relationship between the FLOPs of architectures and how it is likely to be over-estimated. It seems that MLP and RF are more likely to overestimate the smaller architectures and underestimate the larger ones, while LSTM and GATES show no obvious preference on the architectures’ FLOPs. Fig. 2(a)(b)(c) shows that GATES can give more accurate rankings on smaller architectures than larger architectures, which indicates that GATES might still have trouble in comparing larger architectures that usually have good performances.

A.3 ONE-SHOT ORACLE EVALUATOR

Luo et al. (2018) made an attempt to use a parameter-sharing evaluator as the oracle evaluator in Fig. 1. That is to say, they use the noisy signals provided by the parameter sharing evaluator to train the predictor. This will significantly accelerate the NAS process, compared with using an expensive traditional evaluator. However, it is found to cause the NAS algorithm to fail to discover good architectures. Also, predictors have been used to accelerate parameter-sharing NAS methods (Li et al., 2020; Wang et al., 2020), since one predictor forward pass is faster than testing on the whole validation queue, even if no separate training phase is needed. In this section, we explore whether a predictor can recover from the noisy training signals provided by the parameter-sharing evaluator. Since GATES achieves consistently better results than other predictors, it is used in the following experiments. Specifically, we want to answer two questions:

1. Can sampling only a subset during supernet training help achieve better one-shot Kendall’s Tau on these architectures?
2. Can predictor training help recover from the noisy training signals provided by the one-shot evaluator?

We randomly sample 78 architectures from the search space. Two differently trained parameter-sharing evaluators are used to provide the one-shot instruction signal of these 78 architectures: 1) Uniformly sampling from the whole search space, 2) Uniformly sampling from the 78 architectures. We find that strategy 1 (sampling from the whole search space) can get a higher evaluation Kendall’s

Tau, no matter whether the evaluation is on the 78 architectures (0.657 V.S. 0.628) or the whole search space (0.701 V.S. 0.670). Thus the answer to Question 1 is “No”.

Then, to answer the second question, we utilize the one-shot instruction signal provided by the supernet trained with 15625 architectures to train the predictor¹. The Kendall’s Tau between the architecture scores given by the resulting predictor and the ground-truth performances is 0.718 on all the 15625 architectures, which is slightly worse than the one-shot instruction signals (0.719). More importantly, BR@1% degrades from 2.5% to 12.1%, thus the answer to Question 2 is “No”.

Thus, we conclude that although training a predictor using one-shot signals can bring acceleration, since no extra inference is needed during the search, it is not beneficial in the sense of evaluation quality (especially of good architectures). Perhaps, incorporating more manual prior knowledge and regularizations can increase the denoising effect, which might be worth future research.

REFERENCES

- Yanxi Li, Minjing Dong, Yunhe Wang, and Chang Xu. Neural architecture search in a proxy validation loss landscape. In *International Conference on Machine Learning*, 2020.
- Renqian Luo, Fei Tian, Tao Qin, Enhong Chen, and Tie-Yan Liu. Neural architecture optimization. In *Advances in Neural Information Processing Systems 31*, pp. 7816–7827. 2018.
- Xuefei Ning, Yin Zheng, Tianchen Zhao, Yu Wang, and Huazhong Yang. A generic graph-based neural architecture encoding scheme for predictor-based nas. In *The European Conference on Computer Vision (ECCV)*, 2020.
- Yanan Sun, Handing Wang, Bing Xue, Yaochu Jin, Gary G Yen, and Mengjie Zhang. Surrogate-assisted evolutionary deep learning using an end-to-end random forest-based performance predictor. *IEEE Transactions on Evolutionary Computation*, 2019.
- Tianzhe Wang, K. Wang, H. Cai, J. Lin, Zhijian Liu, and Song Han. Apq: Joint search for network architecture, pruning and quantization policy. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2075–2084, 2020.

¹The average of scores provided by 3 supernets trained with different seeds is used.