Lightweight Multi-Objective Evolutionary Neural Architecture Search with Low-Cost Proxy Metrics
(Supplementary Material)

Ngoc Hoang Luong[a,b,*], Quan Minh Phan[a,b], An Vo[a,b], Tan Ngoc Pham[a,b], Dzung Tri Bui[a,b]

[a] *University of Information Technology, Ho Chi Minh City, Vietnam*
[b] *Vietnam National University, Ho Chi Minh City, Vietnam*

In this Supplementary, we report the results of the following multi-objective evolutionary neural architecture search (MOENAS) approaches on the NAS-Bench-101 and NAS-Bench-201 benchmarks. The details of all methods are listed in Table 1:

Table 1: List of MOENAS methods.

Algorithm	Performance metric	Search method
NSGA-II (`val_error`)	Validation error rate at the 12-th epoch	Training-based
NSGA-II (`val_loss`)	Validation loss at the 12-th epoch	Training-based
NSGA-II (`train_loss`)	Training loss at the 12-th epoch [9]	Training-based
MOENAS-PSI	Validation error rate at the 12-th epoch	Training-based
MOENAS-TF-PSI	Validation error rate at the 12-th epoch	Training-based
ENAS-TFI	Validation error rate at the 12-th epoch	Training-based
NSGA-II (`synflow`)	Synaptic Flow (`synflow`) [10]	Training-free
NSGA-II (`jacov`)	Jacobian Covariance (`jacov`) [5]	Training-free
NSGA-II (`snip`)	SNIP [3]	Training-free
NSGA-II (`grad_norm`)	Gradient Norm [1]	Training-free
NSGA-II (`grasp`)	Gradient Signal Preservation [12]	Training-free
NSGA-II (`fisher`)	Fisher [11]	Training-free
NSGA-II (`synflow+jacov`)	Sum of `synflow` and `jacov`	Training-free
NSGA-II (`LS+LR+skip`)	Sum of: • Logarithm of `synflow` • Linear Regions (also known as `naswot` [4]) • $\frac{\#\text{skipped layers}}{\#\text{skip connections in a cell}}$ [2]	Training-free
E-TF-MOENAS (ours)	`synflow` and `jacov`	Training-free

1. Additional results on NAS-Bench-101

In this section, we present the performance of MOENAS approaches on the problem instances created on NAS-Bench-101. Regarding the evaluation of the obtained architectures in terms of test error rate and FLOPs, Figure 1 and Table 3 indicate that our method E-TF-MOENAS significantly outperforms other

*Corresponding author
Email addresses: hoangln@uit.edu.vn (Ngoc Hoang Luong), quanpm.16@grad.uit.edu.vn (Quan Minh Phan), 19520007@gm.uit.edu.vn (An Vo), 19520925@gm.uit.edu.vn (Tan Ngoc Pham), 19521386@gm.uit.edu.vn (Dzung Tri Bui)

training-free methods at both Inverted Generational Distance (IGD) and Hypervolume (HV) indicators. Compared to training-based competitors, E-TF-MOENAS is also significantly better in terms of IGD but is slightly worse in terms of HV. However, we note that the search cost of E-TF-MOENAS is less than approximately 30 times the search cost of training-based methods.

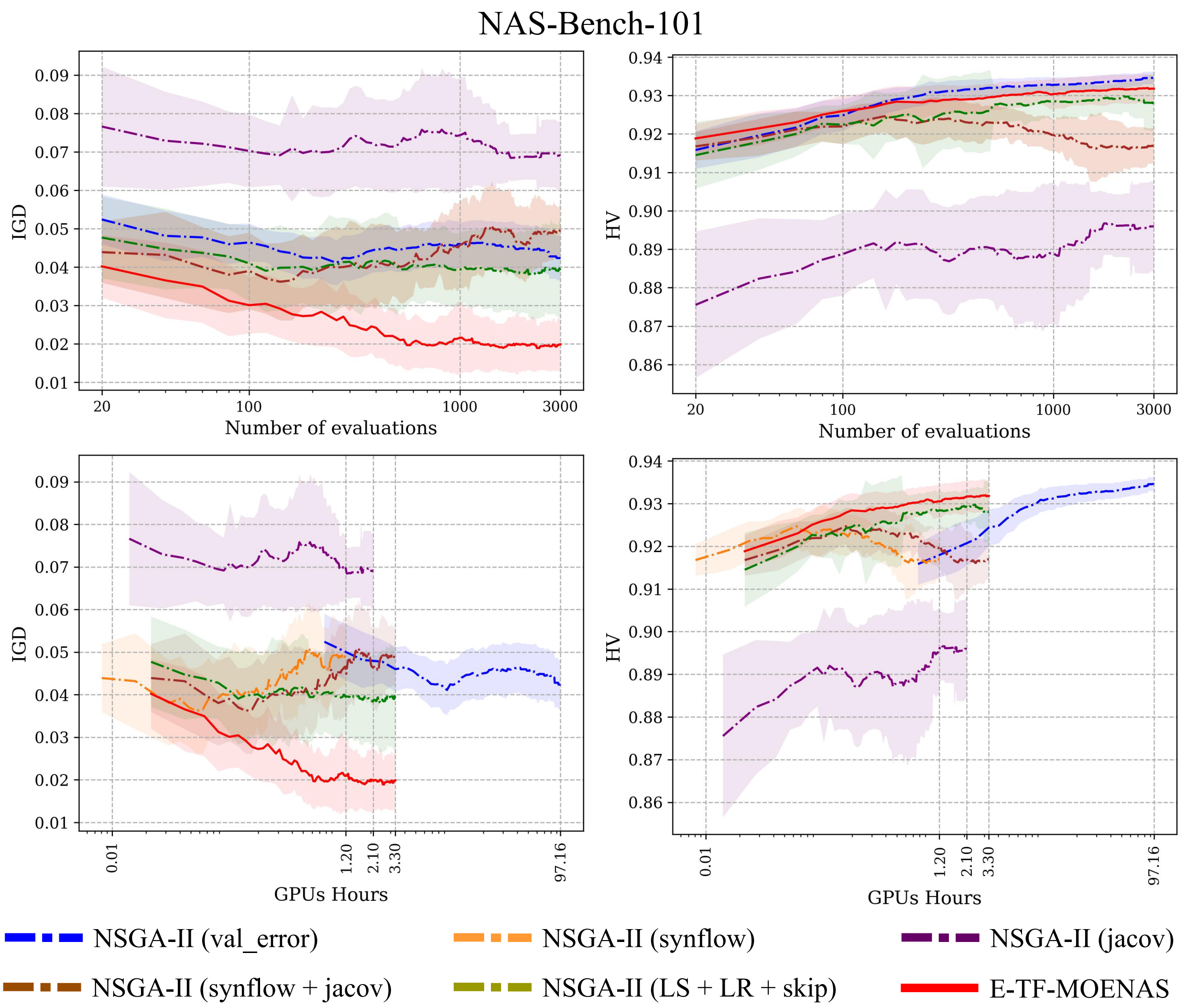

Figure 1: The mean and standard deviation IGD and HV indicator values of MOENAS methods the NAS-Bench-101 problem that minimizes the test error rate and FLOPs. The horizontal axis (log scale): the number of evaluations (Above); the GPUs hours (Below).

Table 2 contains the IGD and HV results for the obtained approximation fronts of architectures evaluated in terms of test error rate and the number of parameters. Table 3 contains the IGD and HV results for the obtained approximation fronts of architectures evaluated in terms of test error rate and FLOPs.

2. More results on NAS-Bench-201

On NAS-Bench-201, besides the MONAS problem formulation that optimizes the test error rate and FLOPs, we additionally create two problem instances by replacing the FLOPs with the number of parameters and latency:

- Figure 2 and Table 4 present the IGD and HV results of MOENAS methods optimizing test error rate and number of parameters.
- Table 5 presents the IGD and HV results of MOENAS methods optimizing test error rate and FLOPs.
- Figure 3 and Table 6 present the IGD and HV results of MOENAS methods optimizing test error rate and latency.

Similar to the results achieved on NAS-Bench-101, Figure 2, Figure 3, Table 4, and Table 6 indicate that E-TF-MOENAS also outperforms other training-free methods. For training-based methods, E-TF-MOENAS is only worse than NSGA-II (`train_loss`) at only the IGD value (for the problem optimizing the number of parameters, see Table 4) or the HV (for the problem optimizing the latency, see Table 6). On the other hand, our method is significantly better than the remaining training-based methods at both IGD and HV values.

Regarding the transferability of the obtained architectures, experimental results are reported in the following tables:

- Table 7 presents the transferred IGD and HV values with respect to test error rate and number of parameters.
- Table 8 presents the transferred IGD and HV values with respect to test error rate and FLOPs.
- Table 9 presents the transferred IGD and HV values with respect to test error rate and latency.

We verify the transferability of architectures obtained by MOENAS methods when searching on CIFAR-10 by re-evaluating them on CIFAR-100 and ImageNet16-120. Table 7 and Table 9 exhibit the superior transferability of E-TF-MOENAS since our method is significantly better than the other approaches at both IGD and HV metric values.

Table 2: The search time and the final IGD & HV values (mean ± standard deviation) of MOENAS methods on the MONAS problem instance created on NAS-Bench-101 that minimizes the test error rate and the number of parameters. Bold results indicate statistical significance.

Algorithm	IGD*	HV**	Search Cost (sec.)
Training-based			
NSGA-II (`val_error`)	0.0427 ± 0.0071	$\mathbf{0.8984 \pm 0.0024}$	349,760
MOENAS-PSI [6, 8]	0.0410 ± 0.0075	$\mathbf{0.8987 \pm 0.0026}$	355,304
MOENAS-TF-PSI [8]	0.0409 ± 0.0079	$\mathbf{0.8994 \pm 0.0027}$	346,471
ENAS-TFI [7]	0.0384 ± 0.0067	$\mathbf{0.9001 \pm 0.0021}^{\dagger}$	366,664
Training-free			
NSGA-II (`synflow`)	0.0479 ± 0.0075	0.8710 ± 0.0092	3347
NSGA-II (`jacov`)	0.0722 ± 0.0064	0.8354 ± 0.0242	2001
NSGA-II (`snip`)	0.0945 ± 0.0084	0.7632 ± 0.0124	1992
NSGA-II (`grad_norm`)	0.0974 ± 0.0072	0.7695 ± 0.0109	2124
NSGA-II (`grasp`)	0.0906 ± 0.0151	0.8064 ± 0.0261	11,322
NSGA-II (`fisher`)	0.0868 ± 0.0049	0.8042 ± 0.0138	2476
NSGA-II (`synflow+jacov`)	0.0480 ± 0.0079	0.8736 ± 0.0074	7268
NSGA-II (`LS+LR+skip`)	0.0389 ± 0.0109	0.8898 ± 0.0088	7583
E-TF-MOENAS (ours)	$\mathbf{0.0188 \pm 0.0052}^{\dagger}$	0.8952 ± 0.0040	5709

† indicates the best result of each problem instance.
* For IGD values, the lesser the better.
** For HV values, the greater the better.

Table 3: The search time and the final IGD & HV values (mean ± standard deviation) of MOENAS methods on the MONAS problem instance created on NAS-Bench-101 that minimizes the test error rate and FLOPs. Bold results indicate statistical significance.

Algorithm	IGD*	HV**	Search Cost (sec.)
Training-based			
NSGA-II (`val_error`)	0.0424 ± 0.0057	$\mathbf{0.9345 \pm 0.0016}$	$349,584$
MOENAS-PSI [6, 8]	0.0447 ± 0.0065	$\mathbf{0.9336 \pm 0.0016}$	$356,196$
MOENAS-TF-PSI [8]	0.0414 ± 0.0074	$\mathbf{0.9348 \pm 0.0019}$	$354,834$
ENAS-TFI [7]	0.0375 ± 0.0047	$\mathbf{0.9361 \pm 0.0008}^{\dagger}$	$370,156$
Training-free			
NSGA-II (`synflow`)	0.0495 ± 0.0070	0.9169 ± 0.0046	4316
NSGA-II (`jacov`)	0.0691 ± 0.0088	0.8959 ± 0.0117	7570
NSGA-II (`snip`)	0.0927 ± 0.0072	0.8416 ± 0.0068	6095
NSGA-II (`grad_norm`)	0.0961 ± 0.0078	0.8506 ± 0.0086	6117
NSGA-II (`grasp`)	0.0957 ± 0.0143	0.8674 ± 0.0170	$16,720$
NSGA-II (`fisher`)	0.0858 ± 0.0049	0.8721 ± 0.0096	7840
NSGA-II (`synflow+jacov`)	0.0495 ± 0.0070	0.9169 ± 0.0046	$11,879$
NSGA-II (`LS+LR+skip`)	0.0392 ± 0.0122	0.9281 ± 0.0072	$11,878$
E-TF-MOENAS (ours)	$\mathbf{0.0199 \pm 0.0070}^{\dagger}$	0.9318 ± 0.0034	$11,882$

† indicates the best result of each problem instance.
* For IGD values, the lesser the better.
** For HV values, the greater the better.

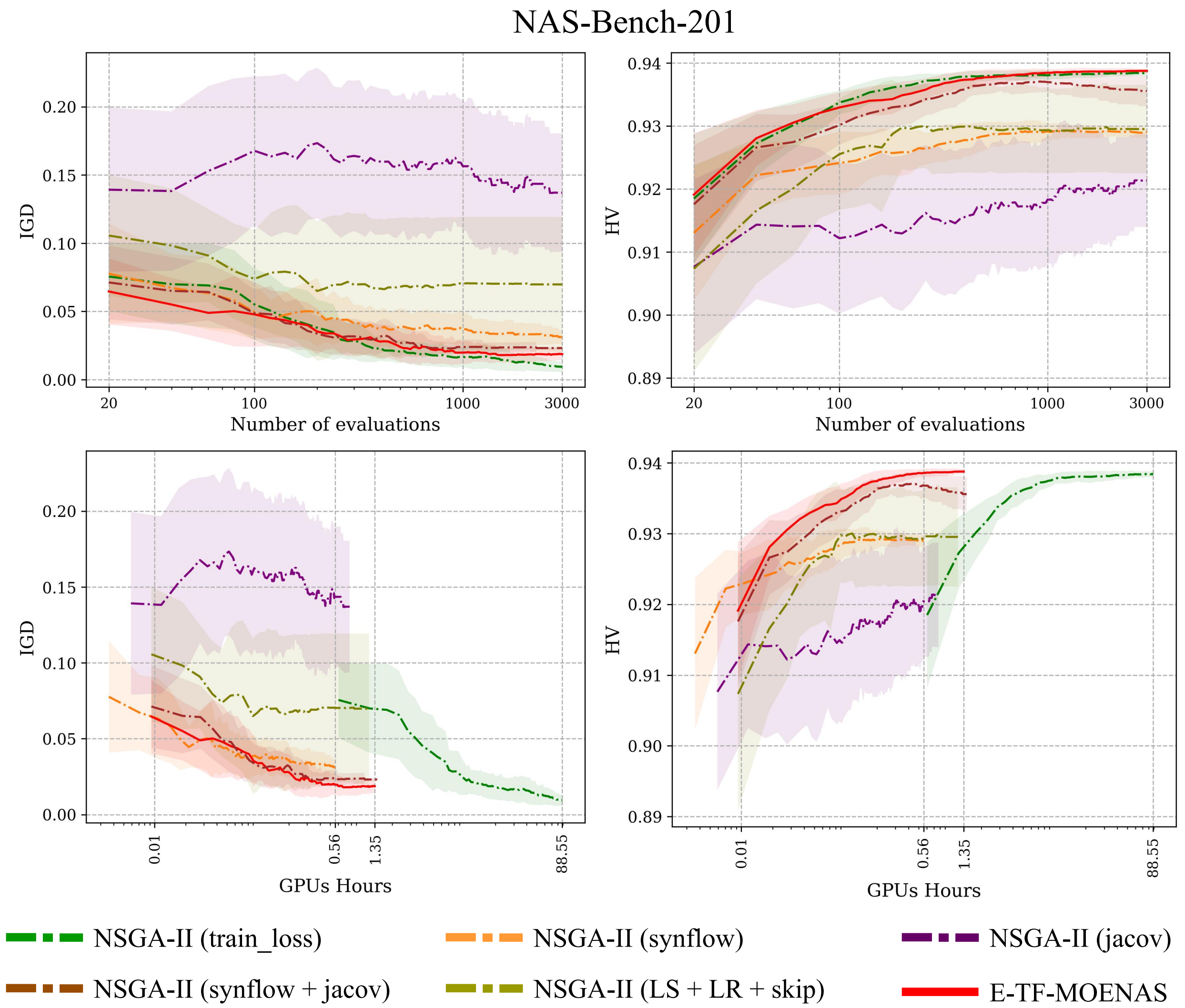

Figure 2: The mean and standard deviation IGD and HV indicator values of MOENAS methods on the NAS-Bench-201 problem instance that minimizes the test error rate and the number of parameters. The horizontal axis (log scale): the number of evaluations (Above); the GPUs hours (Below).

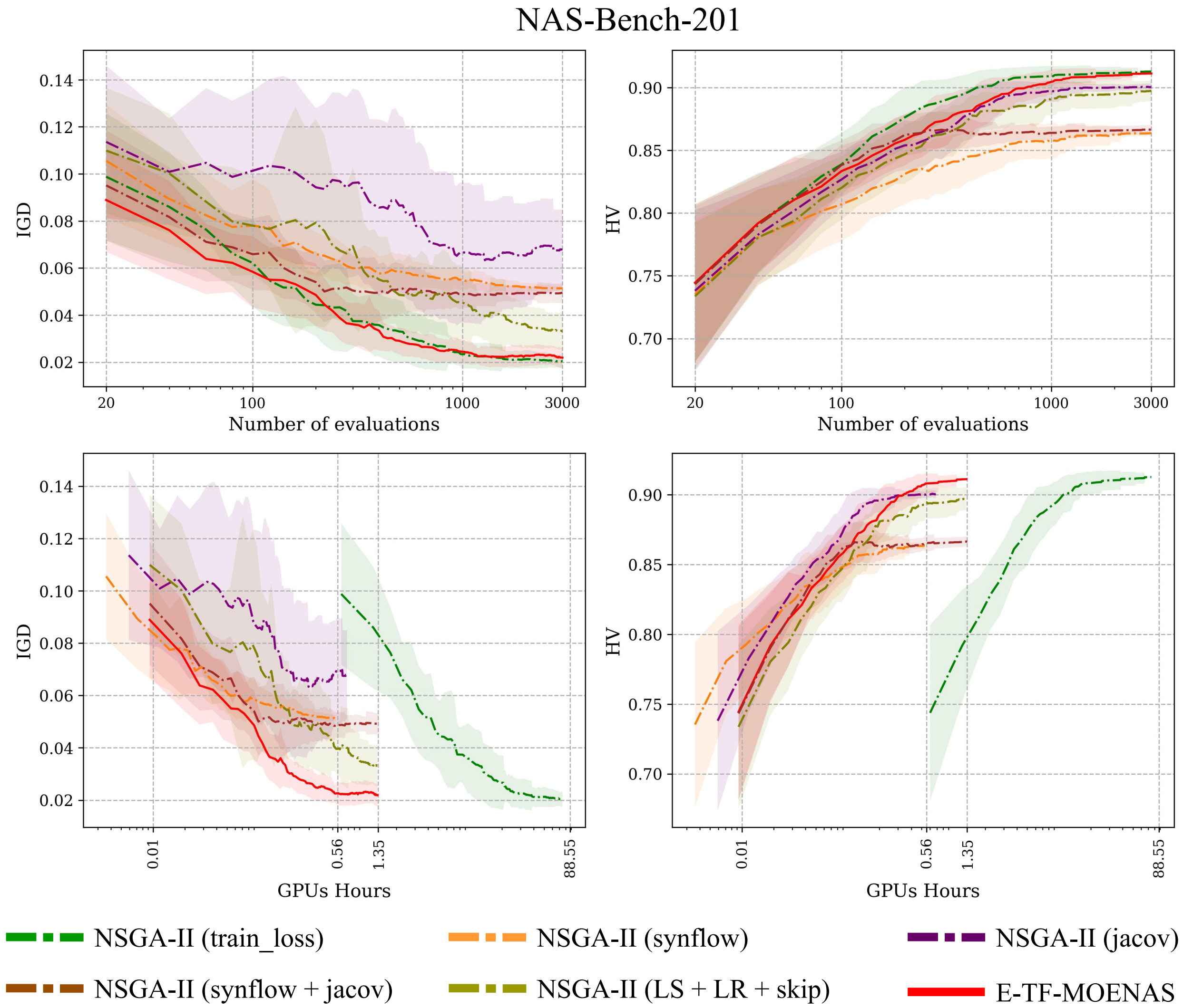

Figure 3: The mean and standard deviation IGD and HV indicator values of MOENAS methods on the NAS-Bench-201 problem instance that minimizes the test error rate and the latency. The horizontal axis (log scale): the number of evaluations (Above); the GPUs hours (Below).

Table 4: The search time and the final IGD & HV values (mean ± standard deviation) of MOENAS methods on the MONAS problem instance created on NAS-Bench-201 that minimizes the test error rate and the number of parameters. Bold results indicate statistical significance.

Algorithm	IGD*	HV**	Search Cost (sec.)
Training-based			
NSGA-II (`val_error`)	0.0453 ± 0.0130	0.9352 ± 0.0012	$275,802$
NSGA-II (`val_loss`)	0.0495 ± 0.0098	0.9346 ± 0.0011	$279,604$
NSGA-II (`train_loss`)	$\mathbf{0.0094 \pm 0.0037}^{\dagger}$	0.9384 ± 0.0004	$318,782$
MOENAS-PSI [6, 8]	0.0495 ± 0.0182	0.9347 ± 0.0024	$275,836$
MOENAS-TF-PSI [8]	0.0431 ± 0.0131	0.9353 ± 0.0011	$284,170$
ENAS-TFI [7]	0.0439 ± 0.0139	0.9353 ± 0.0013	$280,058$
Training-free			
NSGA-II (`synflow`)	0.0312 ± 0.0050	0.9290 ± 0.0008	2016
NSGA-II (`jacov`)	0.1370 ± 0.0434	0.9213 ± 0.0074	2785
NSGA-II (`snip`)	0.2402 ± 0.0009	0.8675 ± 0.0022	2906
NSGA-II (`grad_norm`)	0.2389 ± 0.0005	0.8708 ± 0.0012	2893
NSGA-II (`grasp`)	0.2387 ± 0.0006	0.8718 ± 0.0012	8317
NSGA-II (`fisher`)	0.2409 ± 0.0031	0.8680 ± 0.0012	2971
NSGA-II (`synflow+jacov`)	0.0232 ± 0.0010	0.9356 ± 0.0025	5199
NSGA-II (`LS+LR+skip`)	0.0698 ± 0.0494	0.9295 ± 0.0070	4290
E-TF-MOENAS (ours)	0.0187 ± 0.0046	$\mathbf{0.9388 \pm 0.0004}^{\dagger}$	4876

† indicates the best result of each problem instance.
* For IGD values, the lesser the better.
** For HV values, the greater the better.

Table 5: The search time and the final IGD & HV values (mean ± standard deviation) of MOENAS methods on the MONAS problem instance created on NAS-Bench-201 that minimizes the test error rate and the FLOPs. Bold results indicate statistical significance.

Algorithm	IGD*	HV**	Search Cost (sec.)
	Training-based		
NSGA-II (`val_error`)	0.0474 ± 0.0112	0.8891 ± 0.0013	276,168
NSGA-II (`val_loss`)	0.0502 ± 0.0082	0.8885 ± 0.0012	278,753
NSGA-II (`train_loss`)	$\mathbf{0.0083 \pm 0.0028}^{\dagger}$	$\mathbf{0.8930 \pm 0.0004}$	318,207
MOENAS-PSI [6, 8]	0.0490 ± 0.0180	0.8887 ± 0.0027	275,397
MOENAS-TF-PSI [8]	0.0464 ± 0.0126	0.8893 ± 0.0015	283,849
ENAS-TFI [7]	0.0454 ± 0.0117	0.8892 ± 0.0013	284,296
	Training-free		
NSGA-II (`synflow`)	0.0479 ± 0.0075	0.8710 ± 0.0092	3347
NSGA-II (`jacov`)	0.0722 ± 0.0064	0.8354 ± 0.0242	2001
NSGA-II (`snip`)	0.0945 ± 0.0084	0.7632 ± 0.0124	1992
NSGA-II (`grad_norm`)	0.0974 ± 0.0072	0.7695 ± 0.0109	2124
NSGA-II (`grasp`)	0.0906 ± 0.0151	0.8064 ± 0.0261	11,322
NSGA-II (`fisher`)	0.0868 ± 0.0049	0.8042 ± 0.0138	2476
NSGA-II (`synflow+jacov`)	0.0305 ± 0.0116	0.8828 ± 0.0021	9961
NSGA-II (`LS+LR+skip`)	0.0795 ± 0.0553	0.8821 ± 0.0084	8622
E-TF-MOENAS (ours)	0.0113 ± 0.0043	$\mathbf{0.8932 \pm 0.0006}^{\dagger}$	9671

† indicates the best result of each problem instance.
* For IGD values, the lesser the better.
** For HV values, the greater the better.

Table 6: The search time and the final IGD & HV values (mean ± standard deviation) of MOENAS methods on the MONAS problem instance created on NAS-Bench-201 that minimizes the test error rate and the latency. Bold results indicate statistical significance.

Algorithm	IGD*	HV**	Search Cost (sec.)
	Training-based		
NSGA-II (`val_error`)	0.0634 ± 0.0185	0.9088 ± 0.0008	$238,946$
NSGA-II (`val_loss`)	0.0666 ± 0.0165	0.9027 ± 0.0029	$239,030$
NSGA-II (`train_loss`)	$\mathbf{0.0204 \pm 0.0028}^{\dagger}$	$\mathbf{0.9127 \pm 0.0004}^{\dagger}$	$269,185$
MOENAS-PSI [6, 8]	0.0586 ± 0.0207	0.9091 ± 0.0009	$235,250$
MOENAS-TF-PSI [8]	0.0627 ± 0.0159	0.9091 ± 0.0005	$238,324$
ENAS-TFI [7]	0.0570 ± 0.0202	0.9092 ± 0.0007	$240,965$
	Training-free		
NSGA-II (`synflow`)	0.0514 ± 0.0011	0.8636 ± 0.0021	1935
NSGA-II (`jacov`)	0.0679 ± 0.0170	0.9003 ± 0.0027	2465
NSGA-II (`snip`)	0.0750 ± 0.0258	0.8855 ± 0.0128	2807
NSGA-II (`grad_norm`)	0.0683 ± 0.0269	0.8720 ± 0.0176	2818
NSGA-II (`grasp`)	0.0760 ± 0.0342	0.8870 ± 0.0106	7493
NSGA-II (`fisher`)	0.0761 ± 0.0360	0.8869 ± 0.0132	2844
NSGA-II (`synflow+jacov`)	0.0495 ± 0.0042	0.8665 ± 0.0035	4931
NSGA-II (`LS+LR+skip`)	0.0332 ± 0.0080	0.8972 ± 0.0081	4771
E-TF-MOENAS (ours)	$\mathbf{0.0219 \pm 0.0043}$	0.9112 ± 0.0007	4793

† indicates the best result of each problem instance.
* For IGD values, the lesser the better.
** For HV values, the greater the better.

Table 7: The transfer results on the NAS-Bench-201 problems with respect to the test error rate and the number of parameters. The NAS runs are conducted on the CIFAR-10 dataset and the found architectures are then transferred to the CIFAR-100 and ImageNet-16-120 datasets. Bold results indicate statistical significance.

Algorithm	**CIFAR-100**		**ImageNet16-120**	
	IGD*	HV**	IGD*	HV**
NSGA-II (`val_error`)	0.0732 ± 0.0238	0.7202 ± 0.0043	0.0535 ± 0.0147	0.4711 ± 0.0050
NSGA-II (`val_loss`)	0.0846 ± 0.0168	0.7192 ± 0.0038	0.0603 ± 0.0101	0.4693 ± 0.0050
NSGA-II (`train_loss`)	0.0288 ± 0.0092	0.7350 ± 0.0006	0.0319 ± 0.0041	0.4875 ± 0.0015
MOENAS-PSI	0.0781 ± 0.0254	0.7191 ± 0.0064	0.0555 ± 0.0190	0.4698 ± 0.0081
MOENAS-TF-PSI	0.0704 ± 0.0266	0.7207 ± 0.0041	0.0564 ± 0.0126	0.4704 ± 0.0050
ENAS-TFI	0.0732 ± 0.0241	0.7207 ± 0.0046	0.0541 ± 0.0164	0.4705 ± 0.0055
NSGA-II (`synflow`)	0.0320 ± 0.0028	0.7216 ± 0.0009	0.0249 ± 0.0045	0.4793 ± 0.0016
NSGA-II (`jacov`)	0.1587 ± 0.0382	0.7073 ± 0.0108	0.1115 ± 0.0331	0.4579 ± 0.0148
NSGA-II (`snip`)	0.2846 ± 0.0075	0.5993 ± 0.0111	0.3397 ± 0.0289	0.2335 ± 0.0279
NSGA-II (`grad_norm`)	0.2811 ± 0.0036	0.6047 ± 0.0055	0.2926 ± 0.0201	0.2824 ± 0.0231
NSGA-II (`grasp`)	0.2889 ± 0.0036	0.5956 ± 0.0059	0.3250 ± 0.0142	0.2478 ± 0.0182
NSGA-II (`fisher`)	0.2781 ± 0.0039	0.6093 ± 0.0059	0.3499 ± 0.0291	0.2263 ± 0.0313
NSGA-II (`synflow+jacov`)	0.0447 ± 0.0016	0.7299 ± 0.0033	0.0180 ± 0.0016	0.4842 ± 0.0010
NSGA-II (`LS+LR+skip`)	0.0908 ± 0.0600	0.7171 ± 0.0136	0.0679 ± 0.0476	0.4675 ± 0.0186
E-TF-MOENAS (ours)	$\mathbf{0.0089 \pm 0.0050}^{\dagger}$	$\mathbf{0.7365 \pm 0.0008}^{\dagger}$	$\mathbf{0.0141 \pm 0.0063}^{\dagger}$	$\mathbf{0.4887 \pm 0.0009}^{\dagger}$

† indicates the best result of each problem instance.
* For IGD values, the lesser the better.
** For HV values, the greater the better.

Table 8: The transfer results on the NAS-Bench-201 problems with respect to the test error rate and the FLOPs. The NAS runs are conducted on the CIFAR-10 dataset and the found architectures are then transferred to the CIFAR-100 and ImageNet-16-120 datasets. Bold results indicate statistical significance.

Algorithm	**CIFAR-100**		**ImageNet16-120**	
	IGD*	HV**	IGD*	HV**
NSGA-II (`val_error`)	0.0768 ± 0.0221	0.6679 ± 0.0040	0.0545 ± 0.0131	0.3999 ± 0.0045
NSGA-II (`val_loss`)	0.0853 ± 0.0123	0.6668 ± 0.0034	0.0609 ± 0.0067	0.3976 ± 0.0045
NSGA-II (`train_loss`)	0.0298 ± 0.0096	0.6835 ± 0.0006	0.0313 ± 0.0048	0.4169 ± 0.0014
MOENAS-PSI	0.0761 ± 0.0264	0.6675 ± 0.0067	0.0548 ± 0.0200	0.3988 ± 0.0085
MOENAS-TF-PSI	0.0749 ± 0.0246	0.6681 ± 0.0048	0.0533 ± 0.0140	0.3998 ± 0.0050
ENAS-TFI	0.0754 ± 0.0226	0.6688 ± 0.0040	0.0545 ± 0.0142	0.3992 ± 0.0051
NSGA-II (`synflow`)	0.0328 ± 0.0146	0.6698 ± 0.0028	0.0281 ± 0.0105	0.4084 ± 0.0032
NSGA-II (`jacov`)	0.1975 ± 0.0469	0.6452 ± 0.0168	0.1548 ± 0.0470	0.3700 ± 0.0254
NSGA-II (`snip`)	0.1716 ± 0.0415	0.5767 ± 0.0123	0.2296 ± 0.0397	0.2323 ± 0.0274
NSGA-II (`grad_norm`)	0.1780 ± 0.0496	0.5707 ± 0.0196	0.2356 ± 0.0482	0.2279 ± 0.0359
NSGA-II (`grasp`)	0.2329 ± 0.0580	0.5606 ± 0.0224	0.2479 ± 0.0413	0.2314 ± 0.0257
NSGA-II (`fisher`)	0.2480 ± 0.0331	0.5469 ± 0.0039	0.2428 ± 0.0273	0.2299 ± 0.0153
NSGA-II (`synflow+jacov`)	0.0319 ± 0.0107	0.6692 ± 0.0032	0.0270 ± 0.0094	0.4079 ± 0.0035
NSGA-II (`LS+LR+skip`)	0.0998 ± 0.0652	0.6641 ± 0.0154	0.0733 ± 0.0536	0.3947 ± 0.0195
E-TF-MOENAS (ours)	$\mathbf{0.0101 \pm 0.0076}^{\dagger}$	$\mathbf{0.6844 \pm 0.0012}^{\dagger}$	$\mathbf{0.0165 \pm 0.0081}^{\dagger}$	$\mathbf{0.4189 \pm 0.0018}^{\dagger}$

† indicates the best result of each problem instance.
* For IGD values, the lesser the better.
** For HV values, the greater the better.

Table 9: The transfer results on the NAS-Bench-201 problems with respect to the test error rate and the latency. The NAS runs are conducted on the CIFAR-10 dataset and the found architectures are then transferred to the CIFAR-100 and ImageNet-16-120 datasets. Bold results indicate statistical significance.

Algorithm	**CIFAR-100**		**ImageNet16-120**	
	IGD*	HV**	IGD*	HV**
NSGA-II (`val_error`)	0.0903 ± 0.0316	0.6778 ± 0.0030	0.0244 ± 0.0029	0.6455 ± 0.0066
NSGA-II (`val_loss`)	0.1051 ± 0.0321	0.6677 ± 0.0090	0.0302 ± 0.0056	0.6370 ± 0.0135
NSGA-II (`train_loss`)	0.0369 ± 0.0071	0.6941 ± 0.0020	0.0165 ± 0.0028	0.6737 ± 0.0029
MOENAS-PSI	0.0858 ± 0.0336	0.6791 ± 0.0027	0.0242 ± 0.0039	0.6472 ± 0.0077
MOENAS-TF-PSI	0.1024 ± 0.0295	0.6783 ± 0.0013	0.0240 ± 0.0027	0.6444 ± 0.0059
ENAS-TFI	0.0890 ± 0.0353	0.6794 ± 0.0025	0.0232 ± 0.0030	0.6462 ± 0.0064
NSGA-II (`synflow`)	0.0802 ± 0.0062	0.6331 ± 0.0071	0.0547 ± 0.0022	0.6558 ± 0.0010
NSGA-II (`jacov`)	0.0768 ± 0.0151	0.6605 ± 0.0048	0.0397 ± 0.0067	0.6159 ± 0.0171
NSGA-II (`snip`)	0.0755 ± 0.0143	0.6412 ± 0.0067	0.0578 ± 0.0053	0.5694 ± 0.0065
NSGA-II (`grad_norm`)	0.0885 ± 0.0180	0.6268 ± 0.0124	0.0807 ± 0.0256	0.5388 ± 0.0484
NSGA-II (`grasp`)	0.0857 ± 0.0224	0.6480 ± 0.0119	0.0709 ± 0.0194	0.5613 ± 0.0421
NSGA-II (`fisher`)	0.0901 ± 0.0263	0.6461 ± 0.0116	0.0636 ± 0.0205	0.5606 ± 0.0384
NSGA-II (`synflow+jacov`)	0.0608 ± 0.0064	0.6521 ± 0.0060	0.0376 ± 0.0061	0.6622 ± 0.0017
NSGA-II (`LS+LR+skip`)	0.0583 ± 0.0116	0.6671 ± 0.0121	0.0442 ± 0.0091	0.6651 ± 0.0052
E-TF-MOENAS (ours)	$\mathbf{0.0301 \pm 0.0036}^{\dagger}$	$\mathbf{0.6903 \pm 0.0017}^{\dagger}$	$\mathbf{0.0155 \pm 0.0026}^{\dagger}$	$\mathbf{0.6741 \pm 0.0010}^{\dagger}$

† indicates the best result of each problem instance.
* For IGD values, the lesser the better.
** For HV values, the greater the better.

References

[1] M. S. Abdelfattah, A. Mehrotra, L. Dudziak, and N. D. Lane. Zero-cost proxies for lightweight NAS. In *ICLR 2021*, Virtual Event, Austria, 2021. OpenReview.net.

[2] N. Cavagnero, L. Robbiano, B. Caputo, and G. Averta. FreeREA: Training-free evolution-based architecture search. In *WACV 2023*, pages 1493–1502, Waikoloa, HI, USA, 2023. IEEE. doi: 10.1109/WACV56688.2023.00154.

[3] N. Lee, T. Ajanthan, and P. H. S. Torr. Snip: single-shot network pruning based on connection sensitivity. In *ICLR 2019*, New Orleans, LA, USA, 2019. OpenReview.net.

[4] J. Mellor, J. Turner, A. J. Storkey, and E. J. Crowley. Neural architecture search without training. In *International Conference on Machine Learning, ICML 2021*, volume 139 of *Proceedings of Machine Learning Research*, pages 7588–7598. PMLR, 2021.

[5] J. C. Mellor, J. Turner, A. J. Storkey, and E. J. Crowley. Neural architecture search without training. *CoRR*, abs/2006.04647, 2020. URL `https://arxiv.org/abs/2006.04647v1`.

[6] Q. M. Phan and N. H. Luong. Enhancing multi-objective evolutionary neural architecture search with surrogate models and potential point-guided local searches. In *IEA/AIE 2021*, volume 12798, pages 460–472, Kuala Lumpur, Malaysia, 2021. doi: 10.1007/978-3-030-79457-6_39.

[7] Q. M. Phan and N. H. Luong. Efficiency enhancement of evolutionary neural architecture search via training-free initialization. In *2021 8th NAFOSTED Conference on Information and Computer Science (NICS)*, pages 138–143, 2021. doi: 10.1109/NICS54270.2021.9701573.

[8] Q. M. Phan and N. H. Luong. Enhancing multi-objective evolutionary neural architecture search with training-free pareto local search. *Appl. Intell.*, 2022. doi: 10.1007/s10489-022-04032-y.

[9] R. Ru, C. Lyle, L. Schut, M. Fil, M. van der Wilk, and Y. Gal. Speedy performance estimation for neural architecture search. In *NeurIPS 2021*, pages 4079–4092, 2021.

[10] H. Tanaka, D. Kunin, D. L. K. Yamins, and S. Ganguli. Pruning neural networks without any data by iteratively conserving synaptic flow. In *NeurIPS 2020*, 2020.

[11] L. Theis, I. Korshunova, A. Tejani, and F. Huszár. Faster gaze prediction with dense networks and fisher pruning. *CoRR*, abs/1801.05787, 2018. URL `http://arxiv.org/abs/1801.05787`.

[12] C. Wang, G. Zhang, and R. B. Grosse. Picking winning tickets before training by preserving gradient flow. In *ICLR 2020*, Addis Ababa, Ethiopia, 2020. OpenReview.net.