Supplementary Materials for Arch-Graph: Acyclic Architecture Relation Predictor for Task-Transferable Neural Architecture Search

Minbin Huang[1], Zhijian Huang[1], Changlin Li[3], Xin Chen[4], Hang Xu[2], Zhenguo Li[2], Xiaodan Liang[1*]
[1]Shenzhen Campus of Sun Yat-sen University [2]Huawei Noah's Ark Lab
[3]ReLER, AAII, UTS [4]The University of Hong Kong
{huangmb5,huangzhj56}@mail2.sysu.edu.cn, changlin.li@monash.edu, cyn0531@connect.hku.hk,
chromexbjxh@gmail.com, li.zhenguo@huawei.com, xdliang328@gmail.com

1. TransNAS-Bench-101

1.1. Tasks

There are 7 tasks in TransNAS-Bench-101 [3] (TB101), namely Object Classification (Cls.O), Scene Classification (Cls.S), Autoencoding (Auto.), Surface Normal (Normal), Semantic Segmentation (Sem.Seg.), Room Layout (Room.) and Jigsaw Puzzle (jigsaw). These tasks are carefully chosen to ensure both diversity and similarity across tasks from Taskonomy [11]. More analysis of similarities of these tasks can be found in [3].

1.2. Search Spaces

There are two search spaces in TB101: Macro-level Search Space and Cell-level Search Space.

Macro-level Search Space. It contains architectures with different depths (the total number of blocks), locations to down-sample feature maps, and locations to raise channels. Two residual blocks are first grouped to form a module. Each architecture then contains 4 to 6 such modules. The module positions can be chosen to downsample the input feature maps 1 to 4 times, and each time the spatial size will shrink by a factor of 2. The network can double its channel 1 to 3 times at chosen locations. This search space thus consists of 3,256 unique architectures.

Micro-level Search Space. It is similar to NAS-Bench-201 that is obtained by assigning different operations (as edges) transforming the feature map from the source to the target node. The predefined operation set has $L = 4$ representative operations: zeroize, skip-connection, 1×1 convolution and 3×3 convolution. The macro-level skeleton is fixed, which contains five modules with doubling channel and down-sampling feature map operations at the 1st, 3rd, and 5th modules. This search space thus contains $4^6 = 4,096$ architectures.

*Corresponding author.

1.3. Architecture Encoding

Previous works [4, 9] have empirically shown that using Graph Convolutional Network (GCN) is helpful for better representing cell-based structures. Therefore, for micro-level search space, we follow [4] to transform the operation-on-edge setting to operation-on-node setting. Hence, a cell-based architecture can be represented uniquely by a node feature matrix where a one-hot vector represents the operation. More details can be found in Fig. 1.

For Macro-level search space, we also use graph representation for architectures. We use one-hot vectors to encode what operation (downsample, double channels, or both) is performed in each module. Take Fig. 1(right) for example, X is the node feature matrix of the architecture where we double the channel in the 2nd and the 3rd module, downsample the feature map in the 4th module and both double the channel and downsample the feature map in the 5th module.

2. Maximal Weighted Acyclic Subgraph

Maximal Acyclic Subgraph (MAS) problem was included by R.Karp in his list of 21 NP-complete problems [6], which is defined in Definition 1.

Definition 1 *The Maximal Acyclic Subgraph (MAS) of $\mathcal{G}$ is a graph $\hat{\mathcal{G}} = (\mathcal{V}, \hat{\mathcal{E}})$ such that $\hat{\mathcal{E}} \subseteq \mathcal{E}$, $\hat{\mathcal{G}}$ has no cycles and the number of edges $|\hat{\mathcal{E}}|$ is maximal.*

This concept is very helpful after we obtain the Architecture Relation Graph, since we want to drop as few edges as possible to make the Relation Graph acyclic. Cvetkovic *et al*. proposed an algorithmic solution [2] to the max-MAS problem[2], which is closely related to the MAS problem by treating it as the following optimization problem. ρ is the

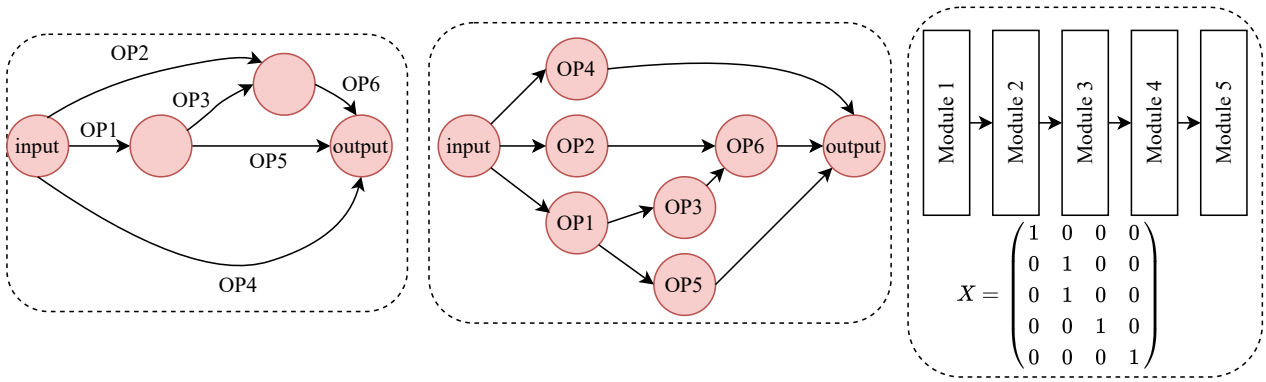

Figure 1. (Left, operation on edge) Graph representation of structures in micro-level search space. (Middle, operation on node) Equivalent representation of structures in micro-level search space. (Right) Graph representation of structures in macro-level search space. X is the feature matrix of an architecture where it doubles its channel in the 2nd and the 3rd module, downsample the feature map in the 4th module and both double the channel and downsample the feature map in the 5th module.

spectral radius and $\mathcal{B}(A, r)$ is the L_1 ball of radius r centered at A.

$$\begin{aligned} &\min \ \rho(X) \\ &s.t. \quad X \in \mathcal{B}(A, r) \end{aligned} \tag{1}$$

Definition 2 *(The max-MAS problem) Finding the minimal integer r such that a given graph $\mathcal{G}$ with adjacency matrix A can be made acyclic by cutting at most r incoming edges from each vertex.*

The reason why we want to minimize the spectral radius of X is the following proposition.

Proposition 1 *Let $\mathcal{G}$ be a directed graph of interest and $A \in \mathbb{R}^{n\times n}$ be its adjacency matrix, then $\mathcal{G}$ is acyclic* if and only if *the spectral radius of A is zero.*

Proof 1 ($\Leftarrow$) *Let $A = VJV^{-1}$ be the Jordan Normal Form of A, where $J = diag(J_{m_1}(\lambda_1), J_{m_2}(\lambda_2) \dots J_{m_s}(\lambda_s))$ and $J_{m_i}(\lambda_i)$ is a Jordan block. Then $A^n = VJ^nV^{-1} = 0$ by the property of a Jordan block and $\rho(A) = 0$. Hence, G has no walks of length n, and therefore it can't have cycles.*

($\Rightarrow$) *If G is acyclic, then it can't have walks of length $n + 1$. This is equivalent to $A^{n+1} = 0$, which implies that $\rho(A) = 0$.*

The solution of max-MAS problem gives us a good approximation of the original graph. However, what we want is not only a subgraph without cycles, but also a subgraph which we can trust those edges in it. This naturally leads to the definition of Maximal Weighted Acyclic Subgraph that minimizes the distance between the subgraph and the original graph, while maximizing edge weights inside it.

When constructing the edge weight matrix, the d in the definition of Trust Score can be computed with respect to any representation of the data. For example, it can be the raw input, an unsupervised embedding of the space, or activations of intermediate representations of the classifier. In our work, we simply follow [5] and use the nearest neighbor distance.

3. Multi-task NAS setting

To further illustrate the effectiveness of Arch-Graph, we conduct an experiment on a more difficult setting by considering finding an architecture that is good on certain task combination $\tau^{(i)} = \{\tau_1, \tau_2, ..., \tau_{i_n}\}$, simultaneously. We implement weakNAS and BONAS on $\tau^{(i)}$ respectively and get a predicted average rank within $\tau^{(i)}$ for each candidate. For Arch-Graph, we can construct a graph with i_n weighted directed edges between nodes. To consider the influence from multiple tasks, we simply add up the (signed) weights to get a final relational graph and apply Algorithm 1 on it. The results are shown in Tab. 1 where $\tau^{(1)} = \{Auto., Normal\}$, $\tau^{(2)} = \{Cls.O., Cls.S.\}$, $\tau^{(3)} = \{Cls.O., Cls.S., Jigsaw\}$. Arch-Graph outperforms other methods in this setting and both BONAS and weakNAS witness significance performance gap with global best. Remarkably, the average rank of our method over the three different multi-task settings outperforms weakNAS and BONAS by **41.5** and **117.7**, proving the superiority of Arch-Graph on more difficult settings.

	Methods	$\tau^{(1)}$	$\tau^{(2)}$	$\tau^{(3)}$	Avg.
TB101	BONAS	285	42	111.3	146.1
	weakNAS	**11.5**	151	47.3	69.9
	Arch-Graph	**11.5**	**32.5**	**41.3**	**28.4**
	Global Best	7	13	23.3	14.4

Table 1. **Arch-Graph on multi-task NAS.** Average rank given by the ground truth ranking of the best architectures among the top **10** architecures predicted by the methods.

4. Visualization Results

We provide in Figs. 2 and 3 more visualizations results. In both figures, the upper part is the result on the pretrain task, jigsaw, after we pretrain the Pairwise Relation Predictor for a budget of 50 models. Note that we can always find the optimal architecture on jigsaw after the pretraining is completed. The lower part is the result on object classification and semantic segmentation. On both tasks, Arch-Graph can successfully adapt to each task, finding reasonably-good architectures.

5. More Ablation Study

Early Stopping. As many previous works [1, 7] alluded to, early stopping can help NAS reduce computational costs and mitigate the overfitting problem. An architecture's final performance is in fact highly correlated with its partially trained performance [7]. It is therefore possible to further reduce computational costs, specifically by using the early stopping performance of sampled architectures as training samples of the pairwise relation predictor.

	Methods	$\tau^{\uparrow}$	$\rho^{\uparrow}$	Avg. Rank$^{\downarrow}$
TB101	Insertion Sort	0.55	0.75	7.83
	Selection Sort	0.57	0.76	10.49
	Bubble Sort	0.58	0.78	6.33
	Arch-Graph	**0.61**	**0.79**	**5.24**
NB201	Insertion Sort	0.63	0.71	3.60
	Selection Sort	0.65	0.76	3.20
	Bubble Sort	0.60	0.73	3.80
	Arch-Graph	**0.67**	**0.79**	**2.40**

Table 2. **MWAS on single-task NAS.** Comparison of different comparator-based sorting algorithms on macro level search space of TransNAS-Bench-101 and NAS-Bench-201.

Summary of early stopping settings on different tasks can be seen in Tab. 3. Comparisons among Arch-Graph and other predictor-based NAS methods using early stopping metric can be found in Tab. 4. We first notice that almost all methods perform poorly on room layout. This is because the correlation coefficients between early stopping architectures and fully trained architectures on room layout is quite small, with a Kendall' τ of 0.28 and Pearson's R of -0.02.

Arch-Graph-zero is least influenced by the change of early stopping. Arch-Graph also show comparable results to it. In comparison, other methods all suffer from great performance drop except for BONAS-t. This shows that compared with previous predictor-based NAS methods, Arch-Graph has the potential to be combined with more early stopping techniques.

Comparator-based Sorting Algorithms.

In addition to Arch-Graph-zero that uses Insertion Sort to get a rough ranking, we also include other comparator-based sorting algorithms such as Selection Sort and Bubble Sort in Tab. 2. The experiments are conducted on TransNAS-Bench-101 (TB101) and NAS-Bench-201 (NB201). Here, the results on TB101 is the average of the single task results on the all 7 tasks. As shown in Tab. 2, Arch-Graph outperforms simple comparator-based sorting algorithms in all metrics, proving the necessity and effectiveness of MWAS.

References

[1] Bowen Baker, Otkrist Gupta, Ramesh Raskar, and Nikhil Naik. Accelerating neural architecture search using performance prediction. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings*. OpenReview.net, 2018. 3

[2] Aleksandar Cvetkovic and Vladimir Yu. Protasov. Maximal acyclic subgraphs and closest stable matrices. *SIAM J. Matrix Anal. Appl.*, 41(3):1167–1182, 2020. 1

[3] Yawen Duan, Xin Chen, Hang Xu, Zewei Chen, Xiaodan Liang, Tong Zhang, and Zhenguo Li. Transnas-bench-101: Improving transferability and generalizability of cross-task neural architecture search. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 5251–5260. Computer Vision Foundation / IEEE, 2021. 1

[4] Lukasz Dudziak, Thomas Chau, Mohamed Abdelfattah, Royson Lee, Hyeji Kim, and Nicholas Lane. Brp-nas: Prediction-based nas using gcns. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 10480–10490. Curran Associates, Inc., 2020. 1

[5] Heinrich Jiang, Been Kim, Melody Y. Guan, and Maya R. Gupta. To trust or not to trust A classifier. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 5546–5557, 2018. 2

[6] Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller and James W. Thatcher, editors, *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA*, The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972. 1

[7] Hanwen Liang, Shifeng Zhang, Jiacheng Sun, Xingqiu He, Weiran Huang, Kechen Zhuang, and Zhenguo Li. DARTS+: improved differentiable architecture search with early stopping. *CoRR*, abs/1909.06035, 2019. 3

[8] Zhichao Lu, Kalyanmoy Deb, Erik D. Goodman, Wolfgang Banzhaf, and Vishnu Naresh Boddeti. Nsganetv2: Evolutionary multi-objective surrogate-assisted neural architecture search. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020*

Tasks	Cls.O.	Cls.S.	Auto.	Normal	Sem. Seg.	Room.	Jigsaw
Fully trained #epoch	25	25	30	30	30	25	10
Early stopping #epoch	15	15	15	15	15	15	5

Table 3. Epoch numbers used in fully trained and early stopping settings.

Tasks		Cls.O.	Cls.S.	Auto.	Normal	Sem. Seg.	Room.	Jigsaw	Avg. Rank	Drop
Metric		Rank	Rank	Rank	Rank	Rank	Rank	Rank		
Single NAS	BONAS [9]	32.8	17.5	75.2	50.0	53.0	46.2	22.4	42.4	9.1
	weakNAS [10]	65.2	22.0	18.2	23.6	**10.6**	39.8	54.4	33.4	22.9
Transfer NAS	BONAS-t [9]	38.8	23.0	16.0	15.4	30.6	63.8	-	31.3	3.4
	nsganetv2 [8]	47.0	83.2	37.8	52.4	42.4	87.6	-	58.4	24.0
	weakNAS-t [10]	51.0	45.2	15.0	74.6	21.4	60.2	-	44.6	29.1
	Arch-Graph-zero	7.3	3.0	11.8	3.5	13.3	23.5	-	10.4	**2.6**
	Arch-Graph	**4.5**	**3.0**	**5.5**	**2.5**	12.3	**23.3**	-	**8.5**	3.2
	Global Best	1	1	1	1	1	1	1	1	-

bold indicates the best result.

Table 4. Performance of NAS methods when training predictors with early stopping on Macro level search space. Drop means average ranking dropped compared to training predictors using fully trained performance.

- *16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I*, volume 12346 of *Lecture Notes in Computer Science*, pages 35–51. Springer, 2020. 4

[9] Han Shi, Renjie Pi, Hang Xu, Zhenguo Li, James T. Kwok, and Tong Zhang. Bridging the gap between sample-based and one-shot neural architecture search with BONAS. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 1, 4

[10] Junru Wu, Xiyang Dai, Dongdong Chen, Yinpeng Chen, Mengchen Liu, Ye Yu, Zhangyang Wang, Zicheng Liu, Mei Chen, and Lu Yuan. Stronger nas with weaker predictors. *arXiv preprint arXiv:2102.10490*, 2021. 4

[11] Amir Roshan Zamir, Alexander Sax, William B. Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 3712–3722. Computer Vision Foundation / IEEE Computer Society, 2018. 1

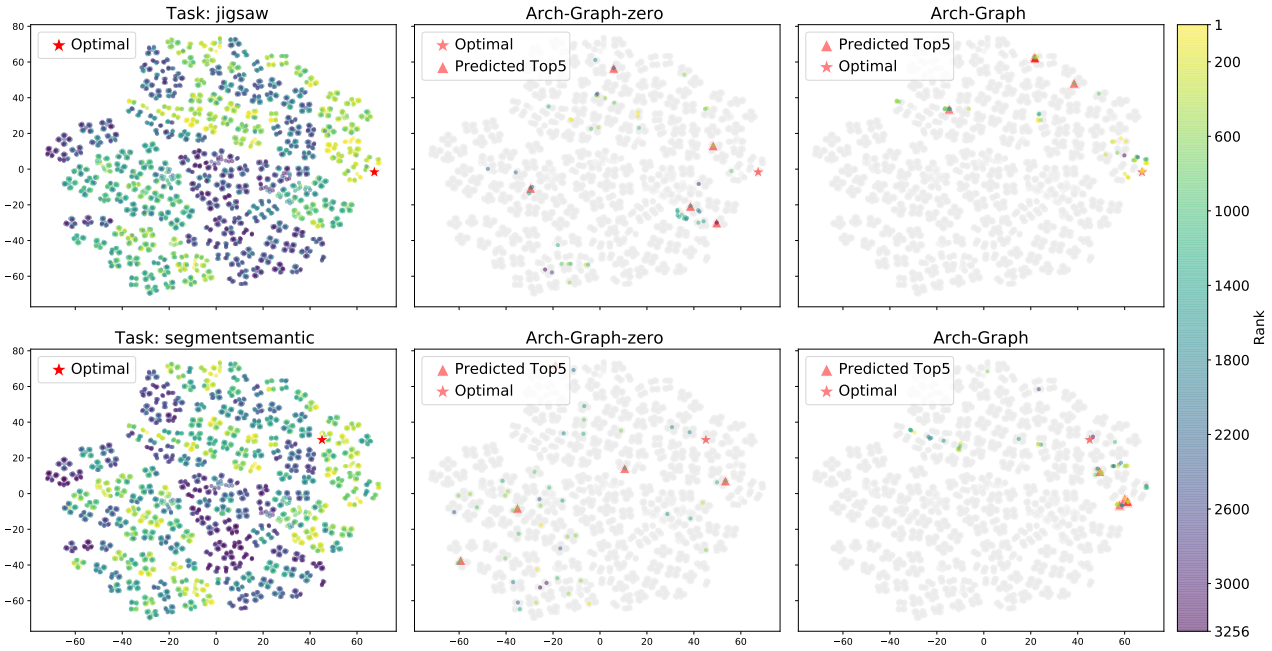

Figure 2. Comparison of searched results of jigsaw and semantic segmentation.

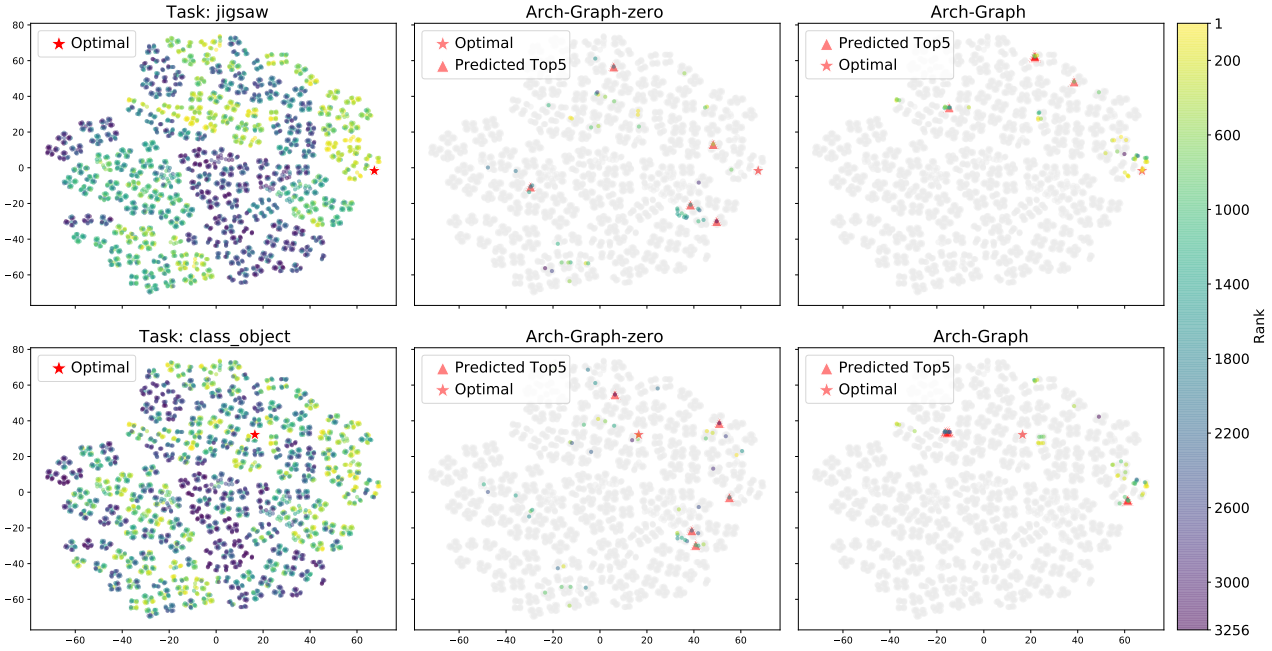

Figure 3. Search results on jigsaw and object classification.