NSGANetV2: Evolutionary Multi-Objective Surrogate-Assisted Neural Architecture Search (Supplementary Material)

Zhichao Lu, Kalyanmoy Deb, Erik Goodman, Wolfgang Banzhaf, and Vishnu Naresh Boddeti

Michigan State University, East Lansing, MI 48824, USA
{luzhicha,kdeb,goodman,banzhafw,vishnu}@msu.edu

Recall that Neural Architecture Search (NAS) is formulated as a bi-level optimization problem in the original paper. The key idea of MSuNAS is to adopt a surrogate model at both the upper and lower level in order to improve the efficiency of solving the NAS bi-level problem. In this supplementary section, we include the following material:

1. Further analysis on the upper level surrogate model of MSuNAS in Section 1.
2. Search performance of MSuNAS on additional three non-standard datasets in Section 2.
3. Post-search analysis in terms of mining for ***architectural design insights*** in Section 3.1.
4. ***"Objective transfer"*** in Section 3.2. Here we seek to quickly search for architectures optimized for target objectives by initializing the search with architectures sampled from *insights* gained by searching on source objectives.
5. The visualization of the final architectures on the six datasets that we searched in Section 4.

1 Correlation Between Search Performance and Surrogate Model

In MSuNAS, we use a surrogate model at the upper architecture level to reduce the number of architectures sent to the lower level for weight learning. There are at least two desired properties of a surrogate model, namely:

- a high rank-order correlation between the performance predicted by the surrogate model and the true performance
- a high sample-efficiency such that the number of architectures, that are fully trained and evaluated, for constructing the surrogate model is as low as possible

In this section, we aim to quantify the correlation between the surrogate model's rank-order correlation (Kendall' Tau [3]) and MSuNAS's search performance. On ImageNet dataset, we run MSuNAS with four different surrogate models, including Multi-Layer Perceptron (MLP), Classification And Regression

Trees (CART), Radial Basis Function (RBF) and Gaussian Processes (GP). We record the accumulative hypervolume [9] and calculate the rank-order correlation on all architectures evaluated during the search. The results are provided in Fig. 1. In MSuNAS, we iteratively fit and refine surrogate models using only architectures that are close to the Pareto frontier. Hence, surrogate models can focus on interpolating across a much restricted region (models close to the current pareto front) in the search space, leading to a significant better rank-order correlation achieved as opposed to existing methods [4,2], i.e., ~ 0.9 for MSuNAS vs 0.476 for ProgressiveNas [4]. Furthermore, we empirically observe that high rank-order correlation in a surrogate model translates into better search performance (lower sample complexity), measured by hypervolume [9], when paired with MSuNAS. On ImageNet, RBF outperforms the other three surrogate models considered. However, to improve generalization to other datasets, we follow an adaptive switching routine that compares all four surrogate models and selects the best based on cross validation (see Section 3.3 in the main paper).

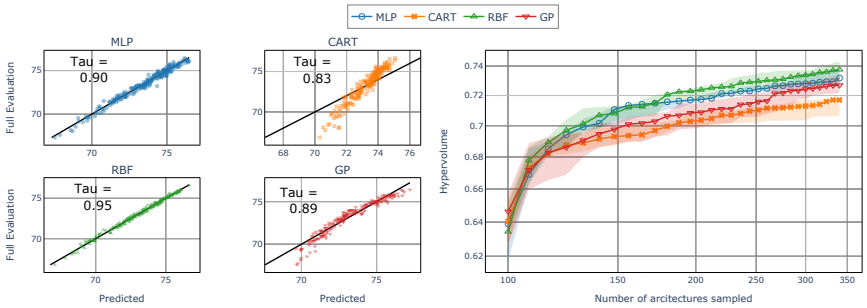


Fig. 1: Left: Kendall's Tau [3] rank-order correlation comparison among different surrogate models. For each model, we calculate the correlation on 350 architectures fully trained and evaluated during the search and we report the mean value over five runs. Right: MSuNAS search performance, measured by hypervolume [9], with different surrogate models. Empirically, we observe a positive correlation between rank-order correlation in surrogate model predictions and the search performance when paired with MSuNAS. All experiments are performed on ImageNet [8]. In general, the rank-order correlation of MSuNAS (~ 0.9) is significantly better than that achieved by ProgressiveNAS [4] (0.476).

2 Search Results on Additional Datasets

As previous mentioned in the main paper, existing NAS methods are rarely evaluated beyond standard datasets, where there exists sufficient training images and well-established training hyperparameters. In addition to the three non-standard datasets shown in the main paper, we provide the search performance

of MSuNAS on three more datasets (see Table 1) in this section. We follow a similar search setting as in the case of CIFAR datasets, outlined in the main paper under Section 3.

Datasets	Type	#Classes	#Train	#Test
Pets [7]	fine-grained	37	3,680	3,369
DTD [1]	fine-grained	47	3,760	1,880
Aircraft [5]	fine-grained	100	6,667	3,333

Table 1: Additional Datasets Searched

Fig. 2 compares the performance of the NSGANetV2 obtained by directly searching on the respective datasets to models from other transfer architectures (with weights fine-tuned) from ImageNet. In general, NSGANetV2 significantly outperforms other both manually and algorithmically designed models on all three datasets.

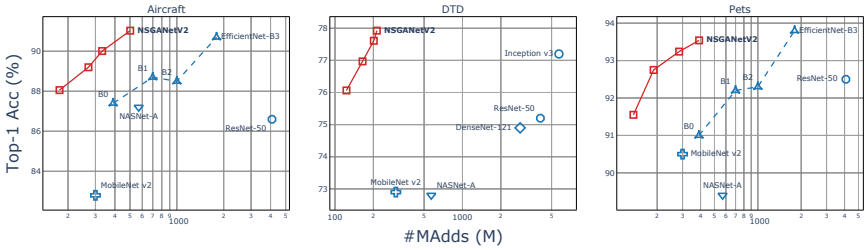
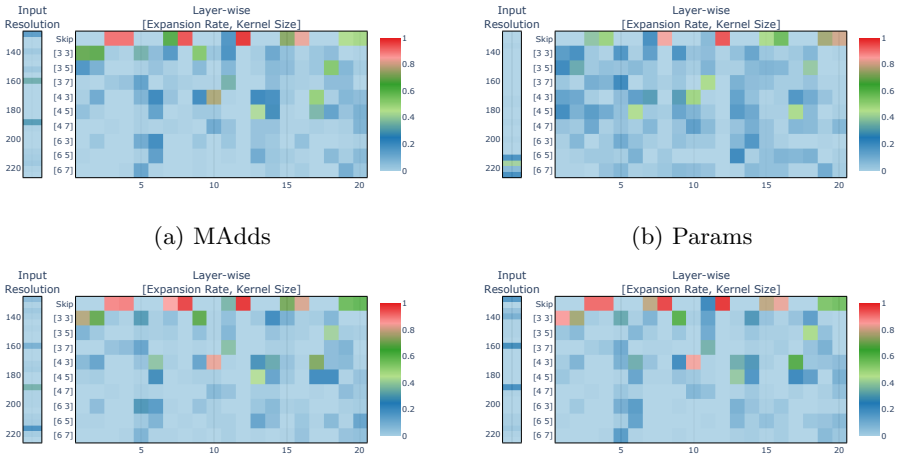


Fig. 2: Performance of the set of task-specific models, i.e. NSGANetV2s, on three additional non-standard datasets, comparing to various baselines from transfer learning (fine-tune from ImageNet).

3 Post Search Analysis

3.1 Mining for Insights

Every single run of MSuNAS generates a set of architectures. Mining the information that is generated through that process allows practitioners to choose a suitable architecture *a posteriori* to the search. To demonstrate one such scenario, we ran MSuNAS to optimize the predictive performance along with one of four different efficiency related measurements, namely MAdds, Params, CPU and GPU latency. At the end of the evolution, we identify the non-dominated architectures and visualize their architectural choices in Fig. 3a - 3d. We observe that the efficient architectures under MAdds, CPU and GPU latency requirements are similar, indicating positive correlation among them, which is not the case with Params. We notice that MSuNAS implicitly exploits the fact that Params is agnostic to the image resolution, and choose to use input images at highest allowed resolution to improve predictive performance (see the Input Resolution heatmap in Fig. 3b).



(a) MAdds (b) Params

(c) CPU latency (d) GPU latency

Fig. 3: The layer-wise architectural choice frequency of the non-dominated architectures obtained by MSuNAS when optimizing predictive performance and MAdds (a) / Params (b) / CPU (c) / GPU latency (d).

3.2 Transfer Across Objectives

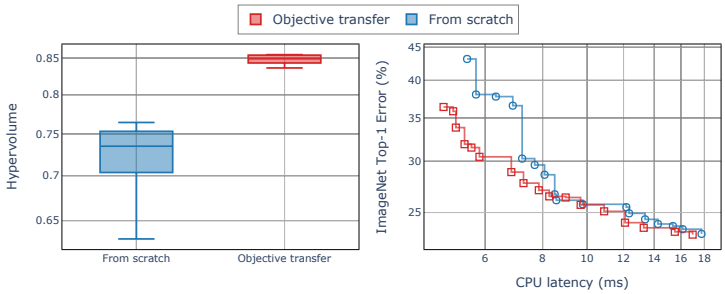


Fig. 4: Comparing MSuNAS's search performance when initialized (1) from randomly sampled architectures and (2) by sampling from distribution constructed from efficient architectures on a related task (MAdds).

Further post-optimal analysis of the set of non-dominated architectures often times reveals valuable design principles, referred to as derived heuristics [6]. Such derived heuristics can be utilized for novel tasks. Here we consider one such example, transferring architectures and associated weights from models that were searched with respect to one pair of objectives to architectures that are optimal

with respect to a different pair of objectives. The idea is that if the objectives that we want to transfer across are related but not identical, for instance, MAdds and Latency, we can improve search efficiency by exploiting such correlations. More specifically, we can search for optimal architectures with respect to a target set of objectives by initializing the search with architectures from a source set of objectives much more efficiently, compared to starting the search from scratch. As a demonstration of this property, we conduct the following experiment:

- Target Objectives: predictive performance and CPU latency.
- Approach 1 ("from scratch"): MSuNAS from randomly (uniformly) initialized architectures.
- Approach 2 ("from objective transfer"): MSuNAS from architectures sampled from distribution constructed from non-dominated architectures of source objectives, namely, predictive performance and MAdds (Fig. 3a).

In Approach 1, we initialize the search process for the target objectives from randomly sampled architectures (uniformly on the search space). In contrast, in Approach 2, we initialize the search process for the target objectives by architectures sampled from the *insights* obtained by searching on a related pair of source objectives (predictive performance and MAdds) i.e., from the distribution in Fig. 3b(a).

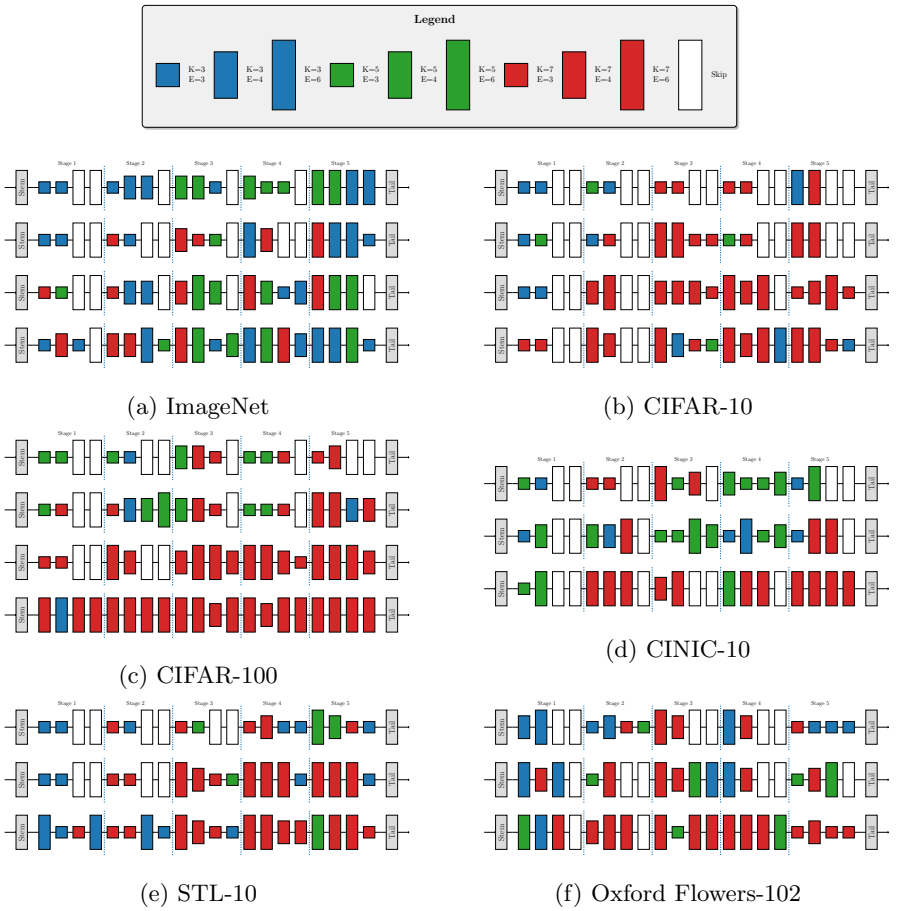
In Fig. 4, we first compare the hypervolume achieved by these two approaches over five runs. We visualize the obtained Pareto front (from the run with median hypervolume) in Fig. 4 (Right). We observe that utilizing *insights* from searching on related objectives can significantly improve search performance. In general, we believe that heuristics can be derived and utilized to improve search performance on related tasks (e.g. MAdds and CPU latency), which is another desirable property of MSuNAS, where a set of architectures are obtained in a single run. The efficiency gains from the objective transfer (Approach 2) we demonstrate here are directly proportional to the correlation between the source and target objectives. However, if the source and target objectives are not related, then Approach 2 may not be more efficient than Approach 1.

4 Evolved Architectures

In this section, we visualize the obtained architectures in Fig. 5. All architectures are found by simultaneously maximizing predictive performance and minimizing MAdds. We observe that different datasets require different architectures for an efficient trade-off between MAdds and performance. Finding such architectures is only possible by directly searching on the target dataset, which is the case in MSuNAS.

References

1. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., Vedaldi, A.: Describing textures in the wild. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2014)



(a) ImageNet

(b) CIFAR-10

(c) CIFAR-100

(d) CINIC-10

(e) STL-10

(f) Oxford Flowers-102

Fig. 5: The architectures of NSGANetV2s referred to in Fig. 1 (main paper) and Fig. 6 (main paper). The *stem* layers in all architectures are the same and not searched. All architectures consist of five blocks, denoted with dashed lines. The first layer in blocks 1, 2, 3, and 5 use stride 2. We use color to denote kernel size and height to denote expansion ratio (legends). For each dataset, architectures are arranged in ascending #MAdds order from top to bottom, i.e. architectures on the top rows have smaller MAdds than those on the bottom rows.

2. Dong, J.D., Cheng, A.C., Juan, D.C., Wei, W., Sun, M.: Dpp-net: Device-aware progressive search for pareto-optimal neural architectures. In: European Conference on Computer Vision (ECCV) (2018)
3. KENDALL, M.G.: A NEW MEASURE OF RANK CORRELATION. Biometrika **30**(1-2), 81–93 (06 1938). https://doi.org/10.1093/biomet/30.1-2.81, `https://doi.org/10.1093/biomet/30.1-2.81`
4. Liu, C., Zoph, B., Neumann, M., Shlens, J., Hua, W., Li, L.J., Fei-Fei, L., Yuille, A., Huang, J., Murphy, K.: Progressive neural architecture search. In: European

Conference on Computer Vision (ECCV) (2018)
5. Maji, S., Kannala, J., Rahtu, E., Blaschko, M., Vedaldi, A.: Fine-grained visual classification of aircraft. Tech. rep. (2013)
6. Myburgh, C., Deb, K.: Derived heuristics-based consistent optimization of material flow in a gold processing plant. Engineering Optimization **50**(1), 1–18 (2018). https://doi.org/10.1080/0305215X.2017.1296436
7. Parkhi, O.M., Vedaldi, A., Zisserman, A., Jawahar, C.: Cats and dogs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2012)
8. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision **115**(3), 211–252 (2015)
9. Zitzler, E., Thiele, L.: Multiobjective optimization using evolutionary algorithms — a comparative case study. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.P. (eds.) Parallel Problem Solving from Nature — PPSN V. pp. 292–301. Springer Berlin Heidelberg, Berlin, Heidelberg (1998)