Network Space Search for Pareto-Efficient Spaces

Min-Fong Hong, Hao-Yun Chen, Min-Hung Chen, Yu-Syuan Xu, Hsien-Kai Kuo, Yi-Min Tsai, Hung-Jen Chen, and Kevin Jou
MediaTek Inc.
{romulus.hong, hao-yun.chen, mh.chen, Yu-Syuan.Xu}@mediatek.com

Abstract

Network spaces have been known as a critical factor in both handcrafted network designs or defining search spaces for Neural Architecture Search (NAS). However, an effective space involves tremendous prior knowledge and/or manual effort, and additional constraints are required to discover efficiency-aware architectures. In this paper, we define a new problem, Network Space Search (NSS), as searching for favorable network spaces instead of a single architecture. We propose an NSS method to directly search for efficient-aware network spaces automatically, reducing the manual effort and immense cost in discovering satisfactory ones. The resultant network spaces, named Elite Spaces, are discovered from Expanded Search Space with minimal human expertise imposed. The Pareto-efficient Elite Spaces are aligned with the Pareto front under various complexity constraints and can be further utilized as NAS search spaces to find satisfactory architectures. Moreover, our NSS approach is capable of searching for superior spaces in future unexplored spaces, revealing great potential in searching for network spaces automatically.

1. Introduction

The recent architectural advance of deep convolutional neural networks [16, 27, 8, 10] considers several factors for network designs (e.g. types of convolution, network depths, filter sizes, etc.), which are combined to form a network space. One can leverage such network spaces to design favorable networks [34, 26] or utilize them as the search spaces for Neural Architecture Search (NAS) [20, 30, 33]. In industry, efficiency considerations for architectures are also required to be considered to deploy products under various platforms, such as mobile, AR, and VR devices.

Design spaces [22, 23] have lately been demonstrated to be a decisive factor for designing networks. Accordingly, several design principles are proposed to deliver promising networks [23]. However, these design principles are based on human expertise and require extensive experiments for validation. In contrast to handcrafted designs, NAS automatically searches for favorable architectures within a predefined search space, which has shown to be a critical factor affecting the performance and efficiency of NAS approaches [17]. Recently, it is common to reuse the tailored search spaces (e.g. DARTS [20] and MobileNet [10] search spaces) in previous works [20, 30, 33, 32]. However, these approaches ignore the potential of exploring untailored spaces, which may cause a gap between tailored NAS search spaces and untailored ones. Furthermore, defining a new search space requires tremendous prior knowledge to best include superior networks. Hence, it is beneficial to
discover superior network spaces automatically, improving the performance of NAS by bridging the gap and reducing human expertise for manually defining NAS search spaces.

In this paper, we propose to **Search for Network Spaces Automatically**. In order to relax the tremendous prior knowledge imposed in the search spaces, we exclude the prevalently used DARTS or MobileNet search spaces and instead search for network spaces on **Expanded Search Space**, which contains scalability with minimal assumptions in network designs. We then define a new problem, **Network Space Search (NSS)**, as searching for favorable network spaces instead of a single architecture. To obtain industry-favorable network spaces, efficiency and practical computation trade-offs are essential factors. Therefore, we propose our NSS method upon differentiable approaches and incorporate multi-objectives into the searching process to search for network spaces under various complexity constraints. The network spaces obtained by our NSS method, named **Elite Spaces**, are Pareto-efficient spaces aligned with the Pareto front [5] with respect to performance and complexity. Moreover, **Elite Spaces** can be further served as NAS search spaces for benefiting current NAS approaches to improve performance (e.g. in CIFAR-100, an averagely 2.3% lower error rate and 3.7% closer to target complexity than the baseline with around 90% fewer samples required to find satisfactory networks). Finally, our NSS method is capable of searching for superior spaces from various search spaces with different complexity, showing the applicability in unexplored and untailored spaces.

Our contributions are summarized below:

- We propose a whole new Auto-ML framework, **Network Space Search (NSS)**, to automatically search for favorable network spaces instead of a single architecture, reducing the human expertise involved in both designing network designs and defining NAS search spaces. To facilitate the NSS framework, we also define **Expanded Search Space** as a large-scale search space to search for favorable network spaces.

- We further incorporate multi-objectives into NSS to search for network spaces under various complexity constraints, and the discovered network spaces, named **Elite Spaces**, deliver satisfactory performance and are aligned with the Pareto front of **Expanded Search Space**. **Elite Spaces** can further be served as NAS search spaces to improve the effectiveness of differentiable NAS methods.

- Our NSS approach is capable of being exploited in unexplored network spaces with various complexity, demonstrating considerable potential in searching for network spaces in an automatic fashion.

### 2. Related Work

**Network Design.** Since the great success achieved by [16] on ILSVRC 2012 [25], several variants of network architectures [27, 28, 29, 8, 34, 13, 10, 26] are proposed, and the significance of network designs to the performance has been demonstrated. In addition, several design principles are proposed to efficiently discover high-performance networks [23], indicating the importance of network design spaces. However, discovering promising design choices still largely involves prior knowledge and human expertise. In this paper, our proposed Network Space Search (NSS) can automate the process of designing networks.

**Neural Architecture Search.** In order to reduce the manual effort required in exploring network architectures, Neural Architecture Search (NAS) is proposed to automate this high-demanding searching process. NAS has achieved impressive results on image classification [36, 31, 9], objection detection [7, 3], semantic segmentation [18], etc. Early works adopt reinforcement learning (RL) [35, 36, 21] and evolutionary algorithms (EA) [19, 24] to perform the architecture search. To improve computational efficiency, gradient-based methods [20, 6, 33, 32] are proposed and more favored. In addition to the single objective of accuracy, recent NAS methods search for architectures with better trade-offs between accuracy and latency [30, 33, 32, 1], FLOPs [31], and power consumption [11]. Unlike previous NAS methods targeting a single architecture, our proposed NSS incorporates multiple objectives to search for promising search spaces with better trade-offs.

**NAS Search Space.** Search space has been shown to be critical to NAS methods [17], and there are two mostly adopted ones: 1) DARTS search spaces [20], which are widely used in early research [35, 36, 21, 19, 24, 20, 6], can be considered as a directed-acyclic-graph by viewing nodes and edges as latent representations and feature extraction operations (e.g. convolutions), respectively, and NAS searches for the graph topology and the corresponding operation types on each edge. 2) MobileNet search spaces [26], which recently gain more attention regarding a small computation regime for deploying on edge devices (e.g. mobiles) [30, 33, 31, 9, 32], are composed of inverted residual blocks [26], where the combinations of kernel sizes, expansion ratios, squeeze-and-excitation [12] ratios, filter sizes, and the number of identical layers in each block are searched during the searching process. Despite the ubiquity, the above search spaces are tailored and involved with human expertise. Instead of tailoring the search spaces beforehand, we propose NSS to search for search spaces with minimal prior knowledge imposed, and the searched spaces can be served as NAS search spaces to benefit current NAS approaches to further improve performance.
3. Network Space Search

In this section, we introduce our proposed Network Space Search (NSS) framework, which searches for network spaces automatically, delivering promising network spaces and can be further served as NAS search spaces.

We first briefly review Neural Architecture Search (NAS) and compare it with our proposed NSS problem (Sec. 3.1). We then introduce Expanded Search Space, which involves minimal human expertise in network designs, as a generalized search space for NSS (Sec. 3.2). Thirdly, we formulate the NSS problem formally and propose our approach to search for network spaces (Sec. 3.3). Finally, we incorporate multi-objectives into our searching process to obtain network spaces under various complexity constraints (Sec. 3.4). The whole NSS framework is summarized and illustrated in Figure 2.

3.1. Preliminary: Neural Architecture Search

Differentiable neural architecture search [20, 6], which recently shows significant improvement in NAS efficiency to find superior architectures within a fixed search space, has drawn lots of attention. Besides, to reduce the computational cost, probability sampling of super networks [33, 32] is recently utilized for optimization, where Gumbel-Softmax function [14, 6] is often utilized to perform probability sampling while considering differentiability.

The performance of the optimal architecture greatly depends on the designs of the given NAS search space. Since discovering design principles is resource-consuming [23] and defining NAS search spaces greatly involves human expertise, we are interested in automatically acquiring promising network spaces. Therefore, we propose a novel Network Space Search (NSS) framework to discover favorable network spaces instead of searching for a single architecture.

3.2. Expanded Search Space

The ultimate goal of Auto-ML is finding satisfactory networks in an automatic fashion, and we aim to facilitate Auto-ML with our proposed NSS framework. Therefore, the two main goals of NSS are: 1) searching for promising network spaces automatically, and 2) the searched network spaces can be further served as NAS search spaces to obtain superior networks. To achieve the above goals, we first require a large-scale Space with two properties: automatability (i.e. minimal human expertise) and scalability (i.e. capability of scaling networks). Thus, instead of directly adopting the common ones [20, 10], which are used to search for network architectures, we define Expanded Search Space as a search space for NSS to search for network spaces.

A network in Expanded Search Space consists of a stem network, a body network, and a final prediction network. The network body, determining network computational cost, probability sampling of super networks [33, 32] is recently utilized for optimization, where Gumbel-Softmax function [14, 6] is often utilized to perform probability sampling while considering differentiability.

The performance of the optimal architecture greatly depends on the designs of the given NAS search space. Since discovering design principles is resource-consuming [23] and defining NAS search spaces greatly involves human expertise, we are interested in automatically acquiring promising network spaces. Therefore, we propose a novel Network Space Search (NSS) framework to discover favorable network spaces instead of searching for a single architecture.

Figure 2: An overview of the proposed Network Space Search (NSS) framework and Elite Spaces. During the searching process, our NSS method searches spaces from Expanded Search Space based on the feedback from the space evaluation, where we propose a novel paradigm to estimate the space performance by evaluating the comprised architectures. The discovered network spaces, named Elite Spaces, can be further utilized for designing favorable networks and served as search spaces for NAS approaches.
3.3. Searching Network Spaces

After defining Expanded Search Space, we would like to address the question: how to effectively search for network spaces given Expanded Search Space? To answer this, we formulate NSS as a differentiable problem to search for an entire network space:

$$\min_{A \in \mathcal{A}} \min_{w_A} \mathcal{L}(A, w_A)$$  \hspace{1cm} (1)

where the optimal network space $A^* \in \mathcal{A}$ is obtained from $\mathcal{A}$ along with its weights $w_{A^*}$, achieving minimal loss $\mathcal{L}(A^*, w_{A^*})$. Here $\mathcal{A}$ is a space without imposing any prior knowledge in network designs (e.g. Expanded Search Space introduced in Section 3.2). To reduce the computational cost, we also adopt probability sampling and the objective is rewritten to:

$$\min_{\Theta} \min_{w_A} \mathbb{E}_{A \sim P_{\Theta}, A \in \mathcal{A}}[\mathcal{L}(A, w_A)]$$  \hspace{1cm} (2)

where $\Theta$ contains parameters for sampling spaces $A \in \mathcal{A}$. Although we can exploit Eq. 2, which is relaxed from Eq. 1, for optimization, the estimation of expected loss for each space $\mathcal{A}$ is still lacking. To solve this, we adopt distribution sampling to practically optimizing Eq. 2 for the inference of super networks. More specifically, from a sampled space $A \in \mathcal{A}$ in Eq. 2, architectures $a \in A$ are sampled to evaluate the expected loss of $A$. Therefore, our goal formulated in Eq. 2 is further extended accordingly:

$$\min_{\Theta} \min_{w_A} \mathbb{E}_{A \sim P_{\Theta}, A \in \mathcal{A}}[\mathbb{E}_{a \sim P_{\Theta}} \mathcal{L}(a, w_a)]$$  \hspace{1cm} (3)

where $P_\Theta$ is a uniform distribution, and $\Theta$ contains parameters that determine the sampling probability $P_\Theta$ of each architecture $a$. Finally, Eq. 3 is our objective to be optimized for searching network spaces, and the evaluation of expected loss of a sampled space is as well based on it. We further theoretically validate the assumption of uniform distribution for $P_\Theta$ in the supplementary.

Instead of regarding a network space $A$ as a set of individual architectures, we represent it with the components in Expanded Search Space. Recalling that Expanded Search Space is composed of searchable network depths $d_i$ and widths $w_i$, a network space $A$ can therefore be viewed as a subset of all possible numbers of blocks and channels. More formally, it is expressed as $A = \{d_i \subseteq d, w_i \subseteq w\}_{i=1}^N$ where $d = \{1, 2, ..., d_{max}\}$, $w = \{1, 2, ..., w_{max}\}$, and $d_i$ and $w_i$ respectively denote the set of possible numbers of blocks and channels in $\mathcal{A}$. After the searching process, $d_i^A$ and $w_i^A$ are retained to represent the discovered network space.

To improve the efficiency of our NSS framework, we adopt the standard weight sharing techniques in our implementation from two aspects: 1) we adopt the masking techniques to simulate various numbers of blocks and channels by sharing a portion of the super components. 2) To ensure well-trained super networks, we apply the warmup techniques to both block and channel search. Please refer to the supplementary for more details.

3.4. Searching with Multi-Objectives

Although we can search for network spaces by optimizing Eq. 3, a single objective may not be able to discover satisfactory networks meeting the practical trade-offs [1, 30, 31]. Thus, it is preferable for NSS to search for network spaces satisfying multi-objectives for further use of designing networks or defining NAS search spaces.

In this way, the searched spaces allow downstream tasks to reduce the effort made on refining trade-offs and concentrate on fine-grained objectives instead. We focus on discovering networks with satisfactory trade-offs between accuracy and model complexity since its importance in practice for the industry. We incorporate model complexity in terms of FLOPs into our objective (Eq. 1) to search for network spaces fulfilling the constraints. Inspired by the absolute reward function [1], we define our FLOPs loss as:

$$\mathcal{L}_{FLOPs}(A) = |FLOPs(A)|/FLOPs_{target} - 1$$  \hspace{1cm} (4)

where $|\cdot|$ denotes the absolute function and $FLOPs_{target}$ is the FLOPs constraint to be satisfied. We combine the multi-objective losses by weighted summation, and therefore $\mathcal{L}$ in Eq. 1 can be replaced with the following equation:

$$\mathcal{L}(A, w_A) = \mathcal{L}_{task}(A, w_A) + \lambda \mathcal{L}_{FLOPs}(A)$$  \hspace{1cm} (5)

where $\mathcal{L}_{task}$ is the ordinary task-specific loss (Eq. 1, and can be optimized with Eq. 3 in practice) and $\lambda$ is the hyperparameter controlling the strength of FLOPs constraint. After searching by optimizing Eq. 5, we can obtain the network spaces satisfying multi-objectives, where the searched spaces $A^*$ are named Elite Spaces. More concretely, Elite Spaces are derived from the optimized probability distribution $P_\Theta$ after the searching process. We sample $n$ spaces from $P_\Theta$ and identify the one closest to the FLOPs constraint as our Elite Space. Unlike handcrafted network spaces or commonly adopted NAS search spaces [20, 26], Elite Spaces are obtained without prior knowledge or human expertise on network designs. Moreover, Elite Spaces can be further exploited in designing promising networks and served as search spaces, benefiting NAS approaches to improve performance.
4. Experiments

In this section, we present extensive experimental results to demonstrate the effectiveness of Network Space Search (NSS), which is aiming to search for promising network spaces in an automatic fashion. We first illustrate that our NSS approach is able to discover satisfactory network spaces, Elite Spaces, under different FLOPs constraints. Then, we show the discovered Elite Spaces can significantly benefit differentiable NAS methods to improve the effectiveness in searching for network architectures. We further reveal the capability of our approach being exploited in unexplored spaces by demonstrating that our NSS method bridges the gap between manually-defined search spaces and increasingly complex spaces for NAS. Finally, we discuss the considerable potential of NSS to search for network spaces automatically based on our generalized observations. More detailed results can be found in the supplementary.

4.1. Experimental Setups

Although we have adopted weight sharing techniques to improve the efficiency, Expanded Search Space requires the constructed super network to have $d_{\text{max}}$ blocks in each stage and $w_{\text{max}}$ channels in each convolutional kernel since it provides the maximum flexibility in searching for network designs. To fit in our computation resources, we set $d_{\text{max}} = 16$ and $w_{\text{max}} = 512$ in all 3 stages, and define each Elite Space as a continuous range of network depths and widths for simplicity. More concretely, each Elite Space consists of 4 and 32 possible blocks and channels, respectively, and therefore Expanded Search Space results in $(\frac{16}{3})^3 \times (\frac{512}{32})^3 = 2^{18}$ possible network spaces. After the searching process, we sample $n = 5$ spaces and retain the one closest to the FLOPs constraint as our Elite Space. To fundamentally investigate the behaviors of NSS, we focus on CIFAR-10 and CIFAR-100 datasets [15]. Following the convention in previous NAS works [20], we equally split the training images into a training set and a validation set. These two sets are used for training the super network and searching for network spaces, respectively. In addition, we follow the hyperparameter settings in [32] and reserve the beginning epochs for warmup. The batch size is set to 64 to fit in 4 1080Ti GPUs. Please refer to the supplementary for more implementation details.

4.2. Searching Network Spaces

Super Network-based Evaluation. We estimate the performance of Elite Spaces by evaluating their comprised architectures. Following the concept of [2], we train two stand-alone super networks for evaluation to reduce the
tremendous cost of training numerous architectures. Note that these super networks are distinguished from the ones used in searching, and their training cost is not included in the search cost. The weight sharing and warmup techniques mentioned in Section 3.3 are as well adopted to improve the quality of super network weights. We random sample 1000 architectures from each Elite Space and obtain their corresponding accuracy by reusing the weights of the well-trained super networks. The stand-alone super networks are carefully trained with state-of-the-art methods [32, 1] to deliver reliable feedback to estimate the performance of each architecture. Elite Spaces can thus be fairly evaluated.

Performance of Elite Spaces. Following the FLOPs configurations listed in [23], we select several representative settings to demonstrate the capability of our NSS method in obtaining promising network spaces under different FLOPs constraints. We target various FLOPs regimes from the mobile setting (i.e. 600MF) to extremely large models (i.e. 24GF) along with several intermediate sizes. In order to show the evident superiority of Elite Spaces, we randomly sample the same amount of architectures from Expanded Search Space as in Elite Spaces and evaluate them based on the aforementioned protocols for comparison. The results are illustrated in Figures 3 and 4 where blue and orange dots represent the randomly sampled architectures from Expanded Search Space and Elite Spaces, respectively. It can be observed that our NSS method sustainably discovers promising network spaces across different FLOPs constraints in both CIFAR-10 and CIFAR-100 datasets. Elite Spaces achieve satisfactory trade-offs between the error rates and meeting the FLOPs constraints, and are aligned with the Pareto front of Expanded Search Space. Since Elite Spaces discovered by our NSS method are guaranteed to consist of superior networks provided in various FLOPs regimes, they can be utilized for designing promising networks. More importantly, Elite Spaces are searched by NSS automatically, therefore the human effort involved in network designs is significantly reduced.

4.3. Served as NAS Search Spaces

Next, we exploit Elite Spaces discovered in Section 4.2 as the NAS search spaces to demonstrate our NSS method can benefit architecture search. We employ the state-of-the-art differentiable-based NAS method [32] to search for a single architecture fulfilling the FLOPs constraint. For comparison, we perform NAS in [32] directly on Expanded Search Space as our baseline. Since Expanded Search Space is too complex to obtain satisfactory architectures close enough to the constraint within 5 samples, we relax the sampling criterion for the baseline by keeping sampling
### 4.4. Bridging the Gap for NAS

We argue that the NAS built on human-defined search spaces may not work well on the spaces with different complexity. On the contrary, when much more complicated or unexplored network spaces arise, our NSS method is capable of bridging the gap between manually-defined search spaces and increasingly complex spaces for NAS. In order to simulate the trend of increasingly complex spaces, we vary the possible network depths and widths of Expanded Search Space to obtain several spaces with different complexity. Besides the original Expanded Search Space with the highest complexity, there are eight additional spaces grouped into three levels according to their maximum FLOPs. We perform NAS on these spaces as in Section 4.3 and target at the most rigorous 600MF constraint, which is commonly applicable for industry. Rather than reporting the error rate of a single architecture, we follow the evaluation procedure in Section 4.2 except that the samples are drawn from the final probability distribution instead of random sampling to demonstrate the performance of NAS in these increasingly complex spaces.

The results are illustrated in Figure 5a to Figure 5b in ascending order of space complexity, and the figures are truncated at 1000MF for better comprehension. It is observed that the samples are dense and concentrated on the FLOPs constraints in Figure 5a, indicating the searching process able to explore the architectures surrounding the target constraint and potentially discover a superior one. However, NAS struggles to obtain qualified architectures with more...
observed some valuable discrepancies in network designs from Elite Spaces. The observations from Elite Spaces show the trend as \( \{d_1 \leq d_3 \leq d_2, w_1 \leq w_3 \leq w_2\} \), which are in contrast to \( \{d_1 \leq d_2 \leq d_3, w_1 \leq w_2 \leq w_3\} \) from design principles. In addition, [23] conducts the experiments under 360MF to 400MF constraints which are merely a small portion of the whole FLOPs regimes. Therefore, to compare the performance of design principles and our approach, we plot the error EDF as in [23], and 1000 networks for both settings are randomly sampled with and without FLOPs constraints, as depicted in Figure 6a and Figure 6b, respectively. Although the curve corresponding to design principles is slightly left under the sampling constraints, our approach performs better in the full FLOPs regimes and has a higher probability of possessing superior networks. The results demonstrate the considerable potential of NSS in searching for promising network spaces in an automatic fashion, significantly reducing immense computational cost and involved human effort.

5. Conclusion

In this paper, we propose Network Space Search (NSS), a whole new Auto-ML paradigm focusing on automatically searching for Pareto-Efficient network spaces, introducing the great potential of exploring efficiency-aware network spaces. The discovered Elite Spaces deliver favorable spaces aligned with the Pareto front, and benefit the NAS approaches by served as search spaces, improving the model accuracy and searching speed. For future works, we plan to contain more aspects of network designs into our Expanded Search Space, including types of operations or different building blocks, to broaden the generalizability of NSS. We also plan to incorporate more constraints into multi-objectives to fulfill different industrial needs.
References


[25] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition chal-


