Training-Free Transformer Architecture Search With Zero-Cost Proxy Guided Evolution

Qinqin Zhou, Kekai Sheng, Xiawu Zheng, Ke Li, Yonghong Tian, *Fellow, IEEE*, Jie Chen, *Member, IEEE*, and Rongrong Ji, *Senior Member, IEEE*

***Abstract*—Transformers have shown remarkable performance, however, their architecture design is a time-consuming process that demands expertise and trial-and-error. Thus, it is worthwhile to investigate efficient methods for automatically searching high-performance Transformers via Transformer Architecture Search (TAS). In order to improve the search efficiency, training-free proxy based methods have been widely adopted in Neural Architecture Search (NAS). Whereas, these proxies have been found to be inadequate in generalizing well to Transformer search spaces, as confirmed by several studies and our own experiments. This paper presents an effective scheme for TAS called *TR*ansformer *A*rchitecture search with *Zer*O-cost p*R*oxy guided evolution (T-Razor) that achieves exceptional efficiency. First, through theoretical analysis, we discover that the synaptic diversity of multi-head self-attention (MSA) and the saliency of multi-layer perceptron (MLP) are correlated with the performance of corresponding Transformers. The properties of synaptic diversity and synaptic saliency motivate us to introduce the ranks of synaptic diversity and saliency that denoted as DSS++ for evaluating and ranking Transformers. DSS++ incorporates correlation information among sampled Transformers to provide unified scores for both synaptic diversity and synaptic saliency. We then propose a block-wise evolution search guided by DSS++ to find optimal Transformers. DSS++ determines the positions for mutation and crossover, enhancing the exploration ability. Experimental results demonstrate that our T-Razor performs competitively against the state-of-the-art manually or automatically designed Transformer architectures across four popular Transformer search spaces. Significantly, T-Razor improves the searching efficiency across different Transformer search spaces, e.g., reducing required GPU days from more than 24 to less than 0.4 and outperforming existing zero-cost approaches. We also apply T-Razor to the BERT search space and find that the searched Transformers achieve competitive GLUE results on several Neural Language Processing (NLP) datasets. This work provides insights into training-free TAS, revealing the usefulness of evaluating Transformers based on the properties of their different blocks.**

***Index Terms*—Evolution, neural architecture search, training-free proxy, transformer.**

Manuscript received 31 May 2023; revised 10 February 2024; accepted 15 March 2024. Date of publication 19 March 2024; date of current version 5 September 2024. This work was supported in part by National Science and Technology Major Project under Grant 2022ZD0118202, in part by the National Science Fund for Distinguished Young Scholars under Grant 62025603, in part by the National Natural Science Foundation of China under Grant U21B2037, Grant U22B2051, Grant 62176222, Grant 62176223, Grant 62176226, Grant 62072386, Grant 62072387, Grant 62072389, Grant 62002305, and Grant 62272401, and in part by the Natural Science Foundation of Fujian Province of China under Grant 2021J01002, and Grant 2022J06001. Recommended for acceptance by C. Xu. *(Qinqin Zhou, Kekai Sheng, and Xiawu Zheng contributed equally to this work.) (Corresponding author: Rongrong Ji.)*

Qinqin Zhou and Xiawu Zheng are with the Media Analytics and Computing Lab, School of Informatics, Xiamen University, Xiamen 361005, China (e-mail: qinqinzhou@stu.xmu.edu.cn; zhengxiawu@stu.xmu.edu.cn).

Kekai Sheng is with the Institute of Automation, Chinese Academy of Science, Beijing 100190, China (e-mail: shengkekai_D@163.com).

Ke Li is with YouTu Laboratory, Tencent Company, Ltd., Shanghai 518064, China (e-mail: tristanli.sh@gmail.com).

Yonghong Tian is with Peng Cheng Laboratory, Shenzhen 518066, China, and also with the National Engineering Laboratory for Video Technology (NELVT), School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China (e-mail: yhtian@pku.edu.cn).

Jie Chen is with the School of Electronic and Computer Engineering, Peking University, Beijing 100871, China, and also with Peng Cheng Laboratory, Shenzhen 518066, China (e-mail: chenj@pcl.ac.cn).

Rongrong Ji is with the Key Laboratory of Multimedia Trusted Perception and Efficient Computing, Ministry of Education of China, Xiamen University, Xiamen 361005, China (e-mail: rrji@xmu.edu.cn).

Digital Object Identifier 10.1109/TPAMI.2024.3378781

I. Introduction

TRANSFORMER is originally built for Natural Language Processing (NLP) tasks, and Vision Transformer (ViT) [3], [5], [6], [7] has shown its competitiveness in the computer vision community recently. Along with the emergence of manually-designed advanced Transformer architectures [8], [9], [10], Transformer Architecture Search (TAS) [1], [11], [12], [13], [14] makes its grand debut, which aims to search for multiple configurations of Transformer architecture in an automated way. Although existing TAS leverages the one-shot NAS scheme [15], [16], [17], [18], [19] to accelerate the search process, it still requires a high computational cost (e.g., larger than 24 GPU days) to train a supernet that can provide reliable performance estimations on various Transformer architectures. Furthermore, since the magnitude of Transformer search spaces (e.g., $\sim 10^{30}$ in GLiT [13]) far exceeds that of CNN search spaces (e.g., $\sim 10^{18}$ in DARTS [20]) and Transformers usually require more training epochs (e.g., 300), the search efficiency of one-shot based TAS is still unsatisfying for practical scenarios.

It is worth noting that, to enhance the searching efficiency on CNN search spaces, several proxies (e.g., GraSP [21], TE-score [22], and NASWOT [23]) are proposed to evaluate the ranks of different CNN architectures in a zero-cost manner. Typically, a CNN is composed of convolution layers, whereas the fundamental building blocks of a Transformer are multi-head self-attention (MSA) and multi-layer perceptron (MLP), which primarily consist of linear layers. The differences between these architectures raise concerns about the application of existing zero-cost proxies that are verified on CNN directly to the Transformer search space. Thus, it is essential and worthwhile to explore the possibility of developing an effective zero-cost

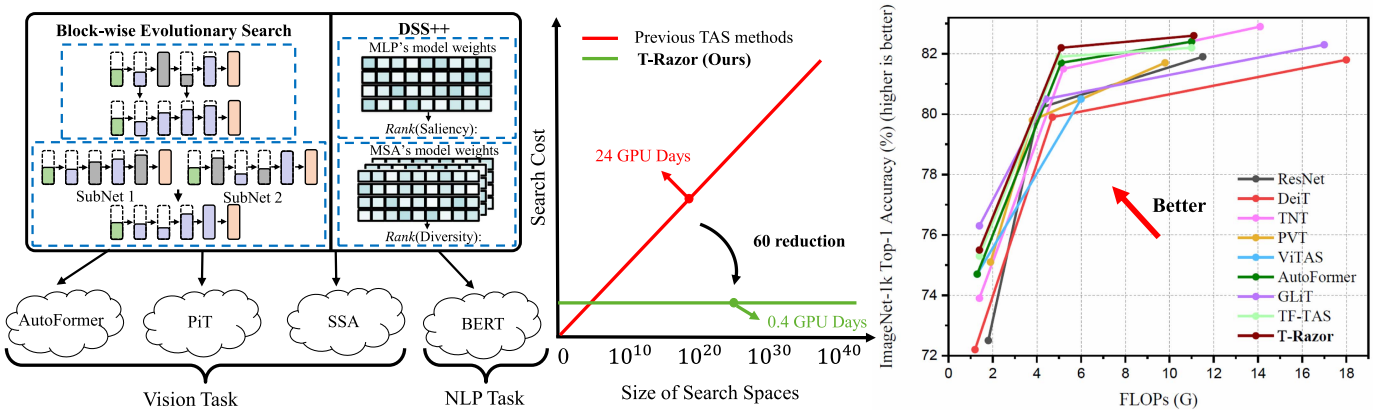


Fig. 1. *Left:* The proposed TRasformer Architecture search with ZerO-cost pRoxy guided evolution (T-Razor), which can be applied to different popular Transformer search spaces (i.e., AutoFormer [1], PiT [2], Shunted ViT [3], and BERT [4]) for both vision and language task scenarios. *Middle:* The proposed method greatly enhances the searching efficiency in different Transformer search spaces: from more than 24 GPU days to less than 0.4 GPU day. *Right:* We compare the architectures searched by T-Razor with state-of-the-art Transformers on ImageNet benchmark, and the results demonstrate that our T-Razor achieves superior accuracy-latency trade-offs.

proxy that is more suitable for ranking Transformers and facilitates the training efficiency of TAS. This problem motivates us to delve deeper into the Transformer architecture to propose an effective approach to conduct TAS in a training-free manner.

To develop effective performance indicators for ranking Transformers, we conduct a modular investigation on MSA and MLP in typical Transformer architectures. Our theoretical analyses and extensive experiments reveal that the two key components, i.e., MSA and MLP, in Transformers possess distinct properties for indicating model performance. Specifically, the *synaptic diversity* of MSA is based on the rank of its weight and gradient matrices, which reflects the representation abilities of MSA. And the *synaptic saliency* of MLP includes its weight and gradient matrices, which implies the weight importance of MLP. As demonstrate in Section III, higher synaptic diversity and synaptic saliency scores generally correspond to better performance of the corresponding Transformer. The 'synaptic' means the connections between neurons in the biomedical field, which is a metaphor for the connections between the blocks inside one model in this paper.

Inspired by the significant insights, we attempt to comprehensively evaluate various Transformers by exploiting the properties of the MSA and MLP mentioned above. Initially, we directly sum the proxy scores of the MSA and MLP in one Transformer as its proxy indicator. However, the different magnitudes of the diversity and saliency scores make it challenging to determine a balanced combination of these two scores. Considering that the synaptic diversity of MSA and the synaptic saliency of MLP are relatively independent, we utilize the ranks of diversity and saliency score obtained from the MSA and MLP to construct the indicator of the corresponding Transformer called DSS++. The DSS++ introduces the correlation between sampled Transformers to eliminate the effect of the different magnitudes of diversity and saliency scores. To further improve the efficiency and reduce the randomness in search process, we propose a block-wise evolution strategy based on DSS++ to search Transformer on different search spaces as shown in the left side of Fig. 1. To the best of our knowledge, this is the first time to propose a block-wise evolution strategy based on the synaptic diversity of MSA and synaptic saliency of MLP in searching Transformer architectures. Furthermore, it is worth emphasizing that our DSS++ is independent of the Transformer search space design and weight sharing strategy. Thus, it is flexible to combine DSS++ with popular state-of-the-art Transformer search space or TAS methods to enhance the searching efficiency. Compared with the manually designed Transformers [6], [24], [25], [26] and the automatically searched ones [1], [13], [27], our T-Razor achieves superior accuracy-latency trade-offs and accelerates the searching procedure from around 24 GPU days to less than 0.4 GPU days, about 60 times faster (see the *Middle* and *Right* of Fig. 1).

To ensure a fair comparison and thorough investigation, we also present a reliable test-bed to evaluate *state-of-the-art* zero-cost proxies (e.g., TE-score [22], NASWOT [23], NTK-trace [28], Gradsign [29] and MGM [30]) in the Transformer search space. We construct a large proxy TAS benchmark based on several pre-trained supernets of AutoFormer [1]. This allows us to compare the relative performance of alternative zero-cost proxies on Transformer architectures. Through our numerical observations of these zero-cost proxies, we are able to empirically verify the relative rankings of different zero-cost proxies in TAS, and our DSS++ outperforms its counterparts. Additionally, we draw some practical insights in designing a better proxy for ranking Transformer architectures.

Overall, our primary contributions are as follows:

- We propose a novel *TR*ansformer *A*rchitecture search with *ZerO*-cost p*R*oxy guided evolution (T-Razor) framework to efficiently search optimal Transformer architectures.
- We introduce a training-free indicator, the ranks of synaptic diversity and saliency denoted as DSS++ that combines the ranks of synaptic diversity and synaptic saliency to form a unified evaluation metric for Transformer architectures.

- Building on the DSS++, we further propose a block-wise evolution strategy to mutate and crossover Transformers in a block-wise manner.
- We conduct comprehensive experiments, including a series of controlled experiments, to analyze the proposed T-Razor. The results demonstrate that T-Razor not only achieves a competitive search performance, but also improves the search efficiency in searching Transformers. These findings provide empirical insights into designing optimal proxy metrics for evaluating Transformer networks.

This work builds upon our previous work [31] with several innovative improvements.

First, we further improved the training-free proxy by refining the rank of a Transformer into the ranks of the proxy scores obtained at the MSA and MLP layers of the corresponding Transformer architecture. Considering ranking relationships at the block level, rather than at the structural level, the proposed proxy further enhances the correlation of the evaluation result by combining the correlation relationships between different Transformers in advance. Second, we propose a block-wise evolution search method to enhance the fine-grained search for Transformer architectures in T-Razor. In this approach, mutation and crossover positions are determined by the DSS++. Third, to further evaluate the generalization ability of our T-Razor, we expanded the search space to cover computer vision and language processing tasks. For the computer vision task, we verify the effectiveness of T-Razor on Shunted ViT search space [3], one of the latest ViT backbones. For language processing tasks, we conduct experiments on BERT search space [4], which show that the proposed DSS++ achieves a competitive performance on several NLP scenarios. Fourth, we conduct several ablation studies to verify the effectiveness of the DSS++ and block-wise evolution search strategy.

The remainder of this paper is organized as follows: Section II provides an overview of the related research. Section III presents a detailed description of our proposed T-Razor. The experimental results and analyses are explicated in Section IV. Ultimately, we draw conclusion in Section V.

II. Preliminary

A. Design & Search Transformer Architectures

Since Transformer [5], the computer vision community has witnessed the emergence of many manually designed advanced Transformer architectures [2], [3], [8], [9], [32]. Technically, most of them consist of the same basic blocks that include MSA, Layer Normalization (LN), and MLP. Existing TAS approaches [1], [13], [14], [19], [27], [33], [34], [35], [36] search different dimensions in MSAs and MLPs, such as the number of heads in MSAs, the ratio of MSAs or MLPs. These methods are generally built on the one-shot NAS framework [15], [16], [17], [18]: train the supernet by training a subnet path in each epoch. The typical one-shot based TAS method AutoFormer [1] includes an entanglement strategy and divides the search space into three sub-supernets, each of which are trained in a one-shot manner for 500 epochs (about 24 GPU days) and requires 8 NVIDIA V100 GPUs. The size of AutoFormer search space is 1.7×10^{16}, which makes it time-consuming in training supernet. GLiT [13] applies the one-shot manner to train modules of Transformer with different granularities in two stages. Each stage need to train for 100 epochs and the size of GLiT search space is about 1.3×10^{30}, which is still time-consuming.

In general, how to reduce the cost of searching the Transformer architectures and ensuring the performance of the searched networks is a fundamental and challenging problem. In this paper, we try to find a way to maintain the performance of TAS and accelerate the searching efficiency.

B. Performance Evaluation Via Zero-Cost Proxy

There are two mainstreams of zero-cost proxy to reduce the cost of performance estimation and enhance searching efficiency. The first one, inspired by the pruning community, sums up the saliency value of each model weight as the proxy of the corresponding CNN architecture with a single forward/backward propagation. The popular methods include Grad-norm [37], SNIP [38], and GraSP [21]. They follow a default assumption: the more salient the weight value is, the more important it is to the model; and the more salient weight one network has, the better performance the model does. The second one, such as TE-score [22], NASWOT [23], Zen-Score [39], NTK-trace [28], GradSign [29], and MGM [30], is designed specifically for CNNs. They analyze the important properties (e.g., expressivity) of the representations of CNNs. Mellor et al. [23] propose jacobian covariance to sum up the saliency of each weight to rank CNNs. Chen et al. [22] apply two theory-inspired indicators as the proxy to find the best subnet. Shu et al. [28] propose NASI with a training-free metric called NTK-trace to approximate the trace norm of Neural Tangent Kernel to characterize the performance of infinite-wide DNNs at initialization. Shu et al. [40] propose a novel framework named hybrid NAS (HNAS) based on several existing training-free metrics including Grad-norm [37], SNIP [38], GraSP [21] and NTK-trace [28]. Zhang et al. [29] introduce GradSign to approximate the optimization landscape of various CNNs by evaluating the gradient signs for a mini-batch of samples, assuming a nearly convex and semi-smooth landscape for a large neighborhood in a randomly initialized network. Xu et al. [30] propose KNAS to select top-k architectures with the largest MGM (the mean of the Gram matrix of gradients) as candidates, which are then trained to choose the best one.

Different from the existing literature, in this paper, for the first time, we identify the shortcoming of directly applying existing proxies in the Transformer search space. Then, we propose a simple yet effective proxy to generate better performance and facilitate the searching efficiency of TAS.

C. Search Strategy for TAS

The formulation of an effective search strategy is a crucial component in Transformer architecture search. Inspired by NAS [15], [22], [41], several Transformer architecture search (TAS) [1], [13], [27], [35] methods have adopted an evolution-based search strategy. These methods require a supernet training

process that generates weights shared with subnets to accelerate the evaluation of sampled subnets. However, the supernet training process is often time-consuming, requiring more than 300 epochs. Additionally, the evolution-based search process for subnet sampling involves randomly determined mutation and crossover positions for architecture. That means the directions of structural mutation and crossover are uncertain, which brings instability into the search process. Furthermore, there is no guarantee of a correlation between the weights of the supernet and subnet, potentially might cause a biased evaluation of the sampled architecture for the evolution search process.

III. Methodology

A. Motivation

The existing TAS methods [1], [13], [27] are relatively time-consuming, especially in performance estimation (e.g., 300 training epochs on 8 GPUs on AutoFormer search space [1] or 3 GPU days to complete the pre-training process on BERT search space [4]). Then, it is worthwhile to leverage the zero-cost proxies [21], [22], [23] to rank Transformers and reduce the computation cost in performance estimation. Nevertheless, the existing zero-cost proxies are specifically designed for the CNN search spaces (e.g., DARTS [20] and NAS-Bench 201 [42]). Obviously, the search space of Transformer is quite different from that of CNN, then the existing proxies could not promise generalization on the Transformer search space (see the experimental results in Section IV-C). It motivates us to explore and exploit the useful properties of MSA and MLP within Transformer, and design an effective Transformer-oriented training-free proxy. Different from our first version, we note that the diversity of MSAs and the saliency of MLPs are two different metrics to measure two dimension of architectures. Although, the simple summation of these two metrics provide good enough results in estimating architectures, it still requires a proper manner to combine these two metrics. Therefore, we propose to use these two metrics to respectively rank architectures and then sum the ranks of these two metrics as the final rank of one architecture.

The training-free indicator makes extremely fast evaluation of Transformers (e.g., less than one GPU day) possible. However, evaluating each Transformer in a search space with over 10^{30} possibilities is impractical. To improve search efficiency, we combine the training-free indicator with a search strategy. One practice of search strategy is to use random sampling to evaluate a certain number of Transformers and select the top-performing one as the searched result. However, this approach can be unstable, and increasing the number of samples reduces search efficiency. Another strategy is to leverage evolution algorithm, which crosses and mutates Transformers based on excellent candidates found so far. However, the locations for mutation and crossover are chosen randomly, also introducing instability into the search process. To address this problem, we propose a block-wise evolution search method inspired by Particle Swarm Optimization (PSO). The presented method adopts the training-free indicator to determine the locations of mutation and crossover in a block-wise manner, enhancing stability during the search.

In this section, we propose an effective method to calculate the synaptic diversity of MSA and generate evaluation results that are positively correlated with the classification accuracies of Transformers. Additionally, we find that when the MLP module has more important weight parameters, i.e., higher synaptic saliency value, the corresponding Transformer network yields better classification performance. Furthermore, we propose a training-free proxy guided evolution search for Transformer architecture, termed as T-Razor. T-Razor employs the ranks of Diversity and Saliency as DSS++ in ranking various Transformers efficiently. Based on DSS++, T-Razor incorporates a block-wise evolution search strategy to improve the search results.

B. Synaptic Diversity in MSA

1) Theoretical Analysis: MSA is a basic fundamental component of Transformers. Several works unveil one important property of MSA: its *diversity* [43], [44]. Dong et al. [44] pointed out that the MSA causes *rank collapse* in the learned representations. In specific, as the input propagates forward in the network and the depth continues to deepen, the outputs of MSAs in Transformers gradually converge to rank-1. And eventually, the output degenerates into a matrix with a rank of 1, the value of each row becomes the same, i.e. the scarcity of diversity. Such rank collapse reduces the diversity of MSAs, which severely degenerates the performance of corresponding Transformer. Intuitively, the degree of the rank collapse could be used as a proxy to evaluate the diversity of MSAs in one Transformer. However, estimating the rank collapse in the high dimension representation space requires a huge computation cost. Actually, Fazel et al. [45] demonstrated that the rank of a matrix contains representative cues of the diversity information within the features. Building on the understandings, the rank of the weight parameters in the MSA module could be adopted as a substitute for the rank collapse to evaluate the Transformer architecture.

2) Synaptic Diversity: For the MSA module, the huge dimension of its weight matrix makes it computationally complex to directly measure the rank of its weight matrix, which hinders practical applications. Thus, it is necessary to devise another way too accelerate the calculation of synaptic diversity in MSA module. Based on the properties of the weight matrix, we prove theoretically that the rank of weight matrix could be approximated using the Nuclear-norm. Therefore, we propose to leverage the Nuclear-norm of the MSA's weight matrix to approximate its rank as the diversity indicator.

Theoretically, the Nuclear-norm of a weight matrix can be treated as an equivalent substitution for its rank, when the Frobenius-norm of the weight matrix meets certain conditions. Specifically, we denote the weight parameter matrix of an MSA module as W_m. m indicates the m-th linear layer in an MSA module, which is usually set to 4 to represent one of the linear Transformations of Query/Key/Value and the subsequent linear layer. The Frobenius-norm of W_m is formulated as:

$$\|W_m\|_F = \sqrt{\sum_{i=1}^{U}\sum_{j=1}^{V}|w_{i,j}|^2}, \tag{1}$$

where U, V are the dimension of W_m, and $w_{i,j}$ denotes the the element in the i-th row and j-th column of W_m.

In our case, W_m is the initialized weight matrix from a linear layer in the MSA module. Typically, in the initial state of a model, none of the weight values reach either infinite or infinitesimally small magnitudes. In other words, each column sum of W_m in the linear layers of MSA is less than a certain constant. According to inequality of arithmetic and geometric means, the upper-bound of $\|W_m\|_F$ is calculated as:

$$\begin{aligned}\|W_m\|_F &\leq \sqrt{\sum_{i=1}^{U}\left(\sum_{j=1}^{V} w_{i,j}\right)\cdot\left(\sum_{j=1}^{V} w_{i,j}\right)} \\ &\leq \sqrt{\sum_{i}^{U}\eta\cdot\eta} = \eta\sqrt{U},\end{aligned} \tag{2}$$

where η denotes the absolute value of the largest constant value of the column sums in W_m. And the upper-bound of $\|W_m\|_F$ could be the largest number of linear independent vectors of W_m, i.e., the matrix rank. We further provide the connection between the Nuclear-norm of W_m and the Frobenius-norm of W_m. For the matrix W_m, the Nuclear-norm could be calculated as follows:

$$\|W_m\|_{nuc} = \sum_{i=1}^{D}\sigma_i, \tag{3}$$

where σ^i denotes the i-th largest singular value and D denotes the number of singular values. Thus, the upper-bound of $\|W_m\|_{nuc}$ could be obtained as:

$$\|W_m\|_{nuc} = \sqrt{\left(\sum_{i=1}^{D}\sigma_i\right)^2} \leq \sqrt{D\sum_{i=1}^{D}\sigma_i^2} = \sqrt{D}\,\|W_m\|_F\,, \tag{4}$$

Similarly, the lower-bound of $\|W_m\|_{nuc}$ should be:

$$\|W_m\|_{nuc} = \sqrt{\left(\sum_{i=1}^{D}\sigma_i\right)^2} \geq \sqrt{\sum_{i=1}^{D}\sigma_i^2} = \|W_m\|_F\,, \tag{5}$$

Consequently, the connection between the Nuclear-norm of W_m and the Frobenius-norm of W_m could be depicted as:

$$\frac{1}{\sqrt{D}}\|W_m\|_{nuc} \leq \|W_m\|_F \leq \|W_m\|_{nuc} \leq \sqrt{D}\,\|W_m\|_F\,, \tag{6}$$

Thus, $\|W_m\|_{nuc}$ is the upper-bound of $\|W_m\|_F$, which is also limited by $\sqrt{D}\|W_m\|_F$. This indicates that, for a well-conditioned W_m in the MSA module, the nuclear norm can closely approximate the rank with a bounded error, especially under the derived constraints on $\|W_m\|_F$ shown in (2).

We further investigate the connection between the Nuclear-norm of W_m and the rank of W_m. Given two randomly selected vectors w_m^i and w_m^j in W_m, when w_m^i and w_m^j are linearly independent, $\|W_m\|_F$ approaches its upper-bound (i.e. $\|W_m\|_{nuc}$). The largest number of linear independent vectors is called the matrix rank. This indicates that: the larger the Frobenius-norm of W_m goes, the closer the rank of W_m is to the diversity of W_m. And according to the *Theorem* proved by Fazel et al. [45] that when $\|W_m\|_F \leq 1$, the convex envelope of the rank of W_m is the Nuclear-norm of W_m. This *Theorem* motivates us to try to approximate the rank of W_m with the Nuclear-norm of W_m in the MSA module. Formally, the Nuclear-norm of W_m is also defined as:

$$\|W_m\|_{nuc} = tr(\sqrt{W_m^T W_m}), \tag{7}$$

where $tr(*)$ denotes the trace of the corresponding matrix. In our case, we always have $\|W_m\|_F \leq \eta\sqrt{U}$, thus the convex envelope of the rank of W_m could be $\|W_m\|_{nuc}/\eta\sqrt{U}$. Theoretically, $\|W_m\|_{nuc}$ is proportional to $\|W_m\|_{nuc}/\eta\sqrt{U}$, which indicates the Nuclear-norm of W_m is an important part of the convex envelope of the rank of W_m. Based on this observation, we propose to use $\|W_m\|_{nuc}$ to measure the diversity of W_m. To better estimate the synaptic diversity of MSA modules from one Transformer network that the weights are randomly initialized, we further consider the aforementioned procedure on the gradient matrix $\partial\mathcal{L}/\partial W_m$ ($\mathcal{L}$ is the loss function) of each MSA module.

Overall, we define the synaptic diversity of the weight parameter in the l-th MSA module as follows:

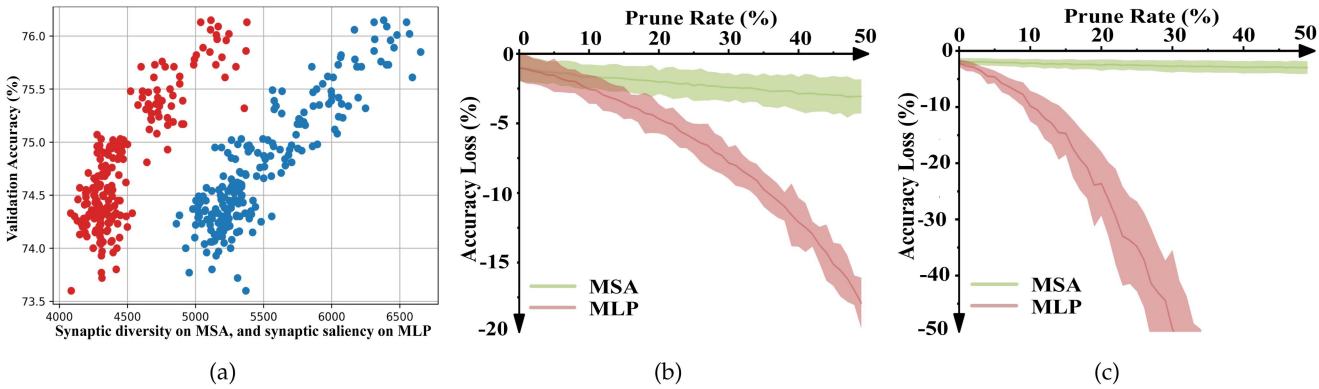
$$D_{MSA}^l = \sum_m \left\|\frac{\partial\mathcal{L}}{\partial W_m}\right\|_{nuc} \cdot \|W_m\|_{nuc}\,. \tag{8}$$

where $\|\frac{\partial\mathcal{L}}{\partial W_m}\|_{nuc}$ and $\|W_m\|_{nuc}$ are scalars. Consider that both model weights and model gradients contain important information about the corresponding model, we use $\|\frac{\partial\mathcal{L}}{\partial W_m}\|_{nuc}$ and $\|W_m\|_{nuc}$ to calculate the diversity of MSA module. We also conduct an experiment on the proposed tiny TAS benchmark, which indicates the Kendall's τ of using $\|\frac{\partial\mathcal{L}}{\partial W_m}\|_{nuc}$ alone is 0.457, and that the Kendall's τ of using $\|W_m\|_{nuc}$ alone is 0.542, neither of which can reach the Kendall's τ of 0.641 obtained by multiplying $\|\frac{\partial\mathcal{L}}{\partial W_m}\|_{nuc}$ and $\|W_m\|_{nuc}$.

To verify the positive correlation between the synaptic diversity of MSA and the test accuracy of the given Transformers, we re-train 100 Transformer networks sampled from AutoFormer [1] and obtain their corresponding classification performance as a tiny TAS benchmark. This benchmark is limited to a small interval of model parameter, which reduces the effecting of the model parameter to a large extent. The Kendall's τ between the performances of the networks and the corresponding synaptic diversity of MSA modules is 0.65 as shown in Fig. 2(a). The results presented in Section IV also demonstrate the positive correlation between the evaluation score of (8) and the validation accuracy of each input Transformer architecture.

C. Synaptic Saliency in MLP

1) Theoretical Analysis: Network pruning [46], [47], [48] has made significant strides in the context of CNNs, and has recently begun to demonstrate its power on Transformer [49], [50], [51]. Several effective CNN punning methods [46], [52], [53] have been proposed to measure the importance of the model weights during the initialization stage. Tanaka et al. [46] introduce the concept of synaptic saliency to measure the importance of weights in pruning CNNs without training. Wang et



(a) (b) (c)

Fig. 2. (a) Illustrations of the positive connection between the diversity and saliency scores (D^l_{MSA} blue points & S^l_{MLP} red points) and the accuracy of the ViT model. (b) & (c) The sensitivity analysis of MSA and MLP to pruning on a flat ViT [1] and a deep-narrow ViT [2], respectively.

al. [49] find that different modules in Transformers exhibit varying degrees of redundancy, even during the initialization stage, and attempt to prune the different dimensions of Transformers. Similarly, Paul et al. [50] observe that a considerable percentage of attention heads can be removed during testing without incurring a significant performance degradation. TAS, likewise, concentrates on searching several important dimensions, which include the number of attention heads, MSA and MLP ratio, etc. In light of these advancements, we attempt to employ the synaptic saliency as a means of assessing various Transformers. Nevertheless, it has been validated [49], [50], [51] that the sensitivities of MSA and MLP to pruning are different. A substantial proportion of weight in the MSA is redundant [50], [51], which has minimal impact on the performance during the test time. This implies that synaptic saliency might demonstrate distinct properties in MSA and MLP.

In order to further support these findings, we demonstrate some quantitative results from a pruning-sensitive experiment. As illustrated in Fig. 2(b), we randomly sample 5 Transformer architectures from AutoFormer search space to analyze the sensitivity of the MSA and MLP to pruning. We observe that the MLP is much more sensitive to pruning than the MSA. We also conduct the same experiment on the deep-narrow Transformer networks (e.g., PiT [2]), and obtain similar observations (see Fig. 2(c)). Moreover, we adopt synaptic saliency on MSA and MLP modules as proxies to calculate the Kendall's τ on the proposed tiny TAS benchmark, respectively. The Kendall's τ of synaptic saliency on the MLP is 0.47, which is better than on the MSA (0.24) and both of the MLP and MSA (0.41). These results coincide with the conclusion drawn above in [49], [50], [51].

Due to the summation-based computation of synaptic saliency, the presence of redundant weight parameters has a cumulative effect. In particular, the MSA module is found to be insensitive to pruning, which indicates a higher level of redundancy in the weight parameters of the MSAs. Previous research [37] in the field of pruning suggests that the redundant weight parameters exhibit significantly smaller values compared to their non-redundant counterparts. While the values of these redundant parameters are relatively small, redundancies exceeding 50 percent tend to have a substantial cumulative effect, especially when distinguishing between similar architectures. Considering the cumulative effect of redundant weight parameters in the Multi-head Self-Attention (MSA) module, it is important to account for such parameters in the zero-cost proxies used to measure saliency. Failing to do so may lead to a cumulative effect in the zero-cost proxies, which in turn, could potentially give a higher rank to a poor-performing network. Meanwhile, the synaptic saliency of the MLP modules is less affected by weight redundancy, making it a suitable proxy for indicating the performance of the corresponding Transformer.

2) Synaptic Saliency: To evaluate the performance of MLPs in Transformer, we utilize the concept of *synaptic saliency*, which has been extensively studied in network pruning. This measure is used to determine the importance of model weights. There are several pruning-based zero-cost proxies [21], [37], [38] that can be directly used to measure the synaptic saliency of CNNs, which are mainly composed of convolution layers. In contrast, Transformers mainly comprise MLP and MSA modules, which exhibit different pruning properties. Through the pruning sensitivity analysis of the MSA and MLP modules in Section III-C, we validate that the MLP modules are highly sensitive to pruning compared to MSA modules. Therefore, the synaptic saliency can better reflect the discrepancies in weight importance within the MLP module. Conversely, the MSA modules are relatively insensitive to the punning, the synaptic saliency of which is often influenced by the redundant weights.

Building upon the pruning sensitivity of MLP, we propose to measure the synaptic saliency in a block-wise manner. Specifically, this block-wise manner measures the synaptic saliency of MLPs as a part of the indicator for a Transformer architecture. Formally, given a Transformer architecture, the saliency score of the l-th MLP module is:

$$S^l_{MLP} = \sum_n \frac{\partial \mathcal{L}}{\partial W_n} : W_n. \tag{9}$$

where : denotes the Frobenius inner product of matrices and n denotes the number of linear layers in the l-th MLP in a specified Transformer network, which is usually set to 2. Fig. 2(a) shows some qualitative results to verify the effectiveness of S_{MLP} in

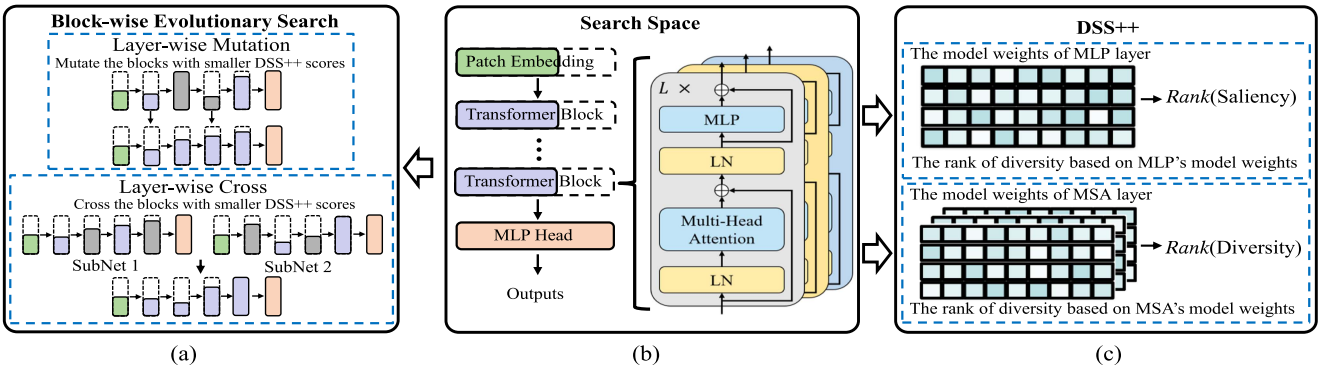


Fig. 3. Framework of our TRasformer Architecture search with ZerO-cost pRoxy guided evolution (T-Razor). Given a Transformer search space (b), T-Razor includes a proposed block-wise evolution search process that utilizes our DSS++ (c) to identify the mutation and crossover positions of each sampled subnet (the gray blocks in (a) denote that these block have the smaller DSS++ scores). Our DSS++ calculates the ranks of diversity and saliency scores as proxies for each Transformer.

evaluating Transformer architectures. On tiny TAS benchmark, a higher value of S_{MLP} typically corresponds to a higher level of validation accuracy for the corresponding Transformer model.

D. Training-Free Proxy Guided Evolution Search for Transformer Architecture

The diagram of our *TR*ansformer *A*rchitecture search with *Z*er*O*-cost p*R*oxy guided evolution (T-Razor) is shown in Fig. 3. T-Razor includes a new training-free indicator denoted as DSS++ to evaluate a Transformer network without training it. Based on DSS++, a block-wise evolution search method is designed to further explore the search spaces.

1) The Ranks of Diversity and Saliency: As the synaptic diversity and synaptic saliency are relatively independent, directly summing these two proxies as the indicator can breaks the relationship between different Transformers, which might be hidden by dimensional units of these two proxies. This issue is demonstrated in Fig. 2(a), which highlights the dimensional difference between the diversity and saliency proxies. Typically, the saliency score of a Transformer is larger than its diversity score. This discrepancy between diversity and saliency affects the evaluation of Transformers. Therefore, we propose to preserve the relationship between diversity and saliency proxies of different Transformers, and eliminate the dimensional differences between diversity and saliency. First, we measure the ranks of the diversity and saliency proxies, respectively, which include the ranking relationships among different Transformers. Then, we sum them up as the ranks of diversity and saliency denoted as DSS++. The rank of a Transformer is refined into the ranks of the proxy scores obtained at the MSA and MLP layers. In this manner, the Transformers that demonstrate high ranks in both saliency and diversity are assigned a relatively stable final rank, whereas those that exhibit inconsistencies in the ranks of saliency and diversity are filtered out.

Compared to our previous work [31], which directly aggregates the proxy scores of diversity and saliency, DSS++ addresses the issue of disorder between diversity and saliency by integrating their ranks. Consequently, Transformers with more consistent ranks of diversity and saliency are easier to outperform other Transformers. Specifically, the DSS++ establishes the relationship between the ranks of diversity and saliency in Transformers. Transformers that exhibit higher consistency in the ranks of these two factors receive higher indicator scores.

Transformers typically consist of patch embedding, MSA block and MLP block. Since the patch embedding block primarily focus on spatial information and provides limited insights into the overall performance of Transformers, then we construct DSS++ without considering the patch embedding block. Based on the various blocks of Transformer, the DSS++ can be decomposed in a block-wise manner. The decomposition allows for a more in-depth analysis of the diversity and saliency rankings within each block.

Integrating the ranks of the diversity of MSA and the saliency of MLP, we formulate the DSS++ as follows:

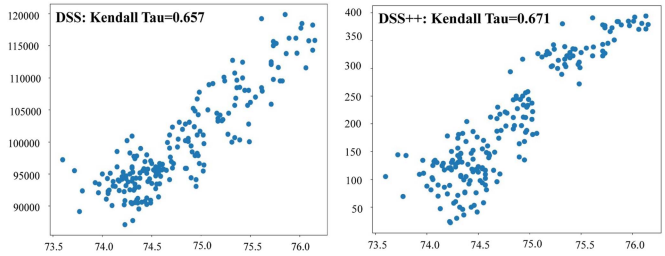
$$S_{DSS++}(\mathcal{A}) = Rank\left(\sum_{l} D^{l}_{MSA}\right) + Rank\left(\sum_{k} S^{k}_{MLP}\right). \tag{10}$$

Overall, the DSS++ evaluates each Transformer architecture from two different perspectives. T-Razor calculates S_{DSS++} after a forward and backward as the indicator of a specified Transformer architecture. We keep each pixel of the input data being 1 to eliminate the affection of input data. Thus, the model outputs are the model weights of each blocks, which are summed to back-propagating. Thus, S_{DSS++} is invariant to random seed. Moreover, the loss at first iteration is formulated as:

$$\mathcal{L} = \mathbb{I}^{T}\left(\prod_{l}^{L}\left|\omega^{[l]}\right|\right)\mathbb{I}, \tag{11}$$

where $\mathbb{I}$ is the all ones vector. It enables (10) to take only the inter-layer interactions of weight parameters into account to measure the diversity of MSAs and the saliency of MLPs.

The DSS++ incorporates the diversity and saliency rankings into the same dimension, making it possible to directly sum them up. It is worth noting that the DSS++ degenerates to the indicator proposed in [31] when sampling only one Transformer. During



(a) DSS (b) DSS++

Fig. 4. Ranking correlation results of (a) DSS and (b) DSS++ on the collected tiny TAS benchmark. The part of Transformers that exhibit higher accuracy present a greater degree of compactness.

a search process, DSS++ takes into account the relationship between the sampled Transformers that have been searched so far by calculating the ranks of diversity and saliency, respectively. Since measuring diversity on MSAs and salience on MLPs is relatively independent, there is often inconsistency between diversity scores and saliency score. As illustrated in Fig. 2, the saliency score of a Transformer is often higher than its diversity scores. For two Transformers, it is possible that one has a higher saliency score than the other, while the converse holds for their diversity score. Such inconsistency among Transformers often interfere with the search process, leading to divergence in the evaluation results of the Transformer. To filter out the Transformers with inconsistent diversity and saliency ranks, we propose an indicator that combines the diversity and saliency ranks to produce a comprehensive Transformer indicator. Based on the proposed DSS++, the Transformers with a more consistent rank between the diversity and the saliency are more likely to obtain high rank. We further compare the evaluation results of DSS++ with our first version (DSS-indicator [31]) on the proposed tiny TAS benchmark. As demonstrated in the Fig. 4, the Kendall τ of DSS-indicator and our DSS++ on the proposed tiny TAS benchmark are 0.657 and 0.671, respectively. It can be observed that the Transformer associated with the top DSS++ outperforms that associated with the DSS-indicator, which coincides with the aforementioned conclusion.

Based on the above analysis, we propose a *TR*ansformer *A*rchitecture search with *Z*er*O*-cost p*R*oxy guided evolution (T-Razor) to further enhances the exploration ability. T-Razor utilizes a block-wise evolution method to enable training-free evaluation of Transformer architectures during the mutation and crossover processes. Benefit from the block-wise manner in calculating DSS++, DSS++ can be combined with different search strategies. In this paper, we focus on improving the evolution search based on our DSS++. This is achieved by combining the DSS++ with the mutation and crossover processes in evolution search, thereby enabling a more precise determination of the specific mutation and crossover positions of Transformers.

2) Block-Wise Evolution Search: Most of the TAS methods [1], [13], [27], [35] are based on the evolution search. However, there is a considerable amount of randomness in the operation of mutation and crossover, which brings instability into TAS. To address this issue, we propose a search strategy

Algorithm 1: T-Razor.

Input: The search space $\mathcal{O}$, datasets, inference limit $\mathcal{I}$, population size P, and search epoch N;
Output: The searched optimal architecture $\mathcal{A}_{top} = None$;

for p *in* $[0, P]$ **do**
  Randomly sample a Transformer from $\mathcal{O}$ with DSS++ scores calculated, which meets the inference limit $\mathcal{I}$, to be added to the population;
end
for *search step* i *in* $[0, N]$ **do**
  Sample two Transformers from population (A_1^i, A_2^i);
  for *mutation step* m *in* $[0, M]$ **do**
    Sample one Transformer from population, the blocks of corresponding Transformers with lowest DSS++ score are mutated;
  end
  for *crossover step* c *in* $[0, C]$ **do**
    Replace the block in A_1^i which has lower DSS++ score than the corresponding block in A_1^i;
  end
  Update the population with the mutation and crossover results;
end
The Transformer with the top DSS++ score in the population is the searched optimal architecture.

called block-wise evolution search, which is based on the proposed DSS++. It aims to improve the stability of TAS in a more granular manner by guiding the mutation and crossover processes according to the DSS++ scores of different blocks of Transformers.

Inspired by Particle Swarm Optimization (PSO), the proposed block-wise evolution search strategy is based on the DSS++ to mitigate the impact of random noise. Specifically, since our proposed DSS++ is obtained according to different blocks of Transformers, we propose to guide the structure to mutating and crossing according to the DSS++ scores obtained in different blocks. The block with the lowest DSS++ score is prioritized during the mutation and crossover processes. To start the search process, a set of Transformers is randomly selected from the search space and evaluated based on their corresponding DSS++ score. The $top - k$ Transformers with the highest DSS++ scores are then chosen to form the population for further exploration. From this population, a subset of Transformers is selected for mutation and crossover. During the crossover process, blocks from two different Transformers are exchanged in a fixed proportion to create a new Transformer that replaces blocks with lower DSS++ scores. For the mutation process, a fixed proportion of blocks with lower DSS++ scores are selected to undergo mutation. After t steps of crossover and mutation, the DSS++ scores for the blocks of the selected Transformer A_i are updated as follows:

$$A_i^{t+1} = A_i^t + v_i^{t+1}, \tag{12}$$

where v_i^{t+1} denotes the change of the DSS++ after the mutation and crossover of the block with the lowest proxy score selected by the current Transformer:

$$v_i^{t+1} = wv_i^t + \gamma_1(pbest_i - x_i(t)) + \gamma_2(gbest - x_i(t)), \quad (13)$$

where w is the inertia weight that controls the balance between exploration and exploitation, γ_1 and γ_2 are the acceleration coefficients that control the influence of personal best and global best positions, $pbest_i$ is the searched best position of particle i, and $gbest$ is the best position found by any particle in the population.

As illustrated in Algorithm 1, given a specified parameter constraint, T-Razor first randomly samples a batch of Transformers from a specified Transformer search space. Subsequently, the DSS++ scores are calculated for each Transformer, reflecting their evaluation rank. T-Razor then picks the Transformers with the higher DSS++ scores to form the population. From this population, a required number of Transformers are subjected to mutation and crossover using our block-wise evolution search strategy. The population is then updated with the results of the mutation and crossover operations. This process is repeated for a total of 160 epochs. Finally, the Transformer in the population with highest DSS++ score as the searched result by T-Razor.

IV. Experiments

A. Implementation Details

T-Razor includes a search stage and a re-train stage. In the block-wise search stage, the number of population size, mutation and crossover number are set to $50, 25, 25$, respectively. The search epoch is set to 160. Each sampled Transformer is initialized with weights and DSS++. To compute the proposed DSS++, the inputs are constructed with each pixel being 1. After the search stage, we retrain the subnet in the population with top-1 DSS++. In the retrain stage, we follow the training configuration in AutoFormer [1] to train the obtained optimal Transformer networks: AdamW optimizer [58] with weight decay 0.05, initial learning rate 1×10^{-3} and minimal learning rate 1×10^{-5} with cosine scheduler, 5 epochs warmup, batch size of 256, and the models are trained with 300 epochs, etc. For BERT, we follow the configuration in [4] to train the searched architectures. All experiments are implemented on NVIDIA Tesla V100 GPUs and the results are estimated on ImageNet [59], CIFAR-10/CIFAR-100 [60], COCO 2017 dataset [61], and ADE20 K dataset [62]. The image resolution is 224×224 by default.

B. Comparison to SOTAs on Various Search Space

In this section, we conduct experiments on the search spaces of the current open source TAS methods and the search spaces built on the existing mainstream Transformers.

Results on AutoFormer Search Space: We first conducted an assessment of T-Razor on the search space of AutoFormer, i.e., AutoFormer search space $\mathcal{S}_A$. We compare the performance of searched optimal Transformers with that of *state-of-the-art* TAS methods [1], [13], [27], [31], as well as manually designed CNNs and Transformers [8], [9], [25], [26], [55] on the ImageNet dataset.

As outlined in Table I, the searched optimal architectures T-Razor (i.e., T-Razor-Ti, T-Razor-S, and T-Razor-B) demonstrate a significant performance improvement over manually designed CNNs [6], [10], [54], [55] across all three model sizes (i.e., tiny, small, and base). Moreover, T-Razor achieves competitive results compared to other manually designed Transformers [8], [9], [24], [25], [26]. Specifically, the searched T-Razor-Ti yields a top-1 accuracy of 75.3%, surpassing DeiT-tiny by 3.1 percent. Furthermore, unlike other popular TAS methods [1], [13], [27], [31] that require more than 24 GPU days to search for optimal architectures, the proposed DSS++ enables us to achieve comparable results with much fewer GPU Days. Our DSS++ comprehensively considers effectiveness and efficiency in searching for optimal Transformer architectures. Based on the estimation results of the proposed DSS++ for each input Transformer, we reduce lots of computation budgets in performance estimation and obtain optimal Transformer architectures with comparable performance in just 0.4 GPU days.

The rank of a Transformer is refined into the ranks of the proxy scores obtained at the MSA and MLP layers. In this manner, the final ranking of variables highly correlated with Rank (MSA) and Rank (MLP) will be relatively stable, while those with low correlation will be filtered. This can also partly explain the dense head structure with high tangential correlation in the distribution results of 100 samples.

Results on PiT Search Space: To substantiate the generalization of our T-Razor, we establish a PiT search space denoted as $\mathcal{S}_P$ based on PiT [2], which adopts some depth-wise convolution as pooling to obtain deep-narrow architectures. In an effort to ensure comprehensiveness, we introduce $\mathcal{S}_P$ on PiT [2] and introduce several important dimensions of Transformer (e.g., depth, head number of MSA, MLP ratio), coupled with depth-wise convolution.

As listed in Table II, our DSS++ is still capable of searching the optimal Transformer architectures that exhibit comparable or superior Top-1 classification accuracy to PiT-Ti and PiT-S. Furthermore, the result of the searched networks outperform the randomly selected ones, PiT-Ti$_{rand}$ and PiT-S$_{rand}$, by a margin of about $2.9 \sim 5\%$. Notably, we observe that the searched architectures of T-Razor on PiT search space achieve lower performance than those on AutoFormer search space listed in Table I. The observation implies that the search space is also an important part of TAS.

Results on Shunted ViT Search Space: Shunted ViT [3] is another cutting-edge Transformer in vision tasks proposed in CVPR-2022, which proposes Shunted Self-Attention (SSA) scheme to explicitly account for multi-scale features. We build a search space based on Shunted ViT to include base dimension, patch size, layer number, head number and MLP ratio as the searchable dimensions, which is about 2.6×10^{12}. We follow the training settings described in Ren et al. [3] to retrain the searched architectures.

The results shown in Table III indicate that T-Razor outperforms Shunted ViT in three levels of architectures. We compare the searched Transformer architectures on Shunted ViT search

TABLE I
COMPARISON RESULTS ON THE AUTOFORMER SEARCH SPACE

Models	#Param (M)	FLOPS (B)	Top-1 (%)	Top-5 (%)	Model Type	Design Type	GPU Days
ResNet-18* [54]	11.7	1.8	72.5	-	CNN	Manual	-
MobileNet-V3 [55]	5.5	-	75.2	-	CNN	Manual	-
Deit-Ti [9]	5.7	1.2	72.2	91.1	Transformer	Manual	-
TNT-Ti [25]	6.1	1.4	73.9	91.9	Transformer	Manual	-
ViT-Ti [8]	5.7	-	74.5	-	Transformer	Manual	-
CPVT-Ti [24]	6.0	-	74.9	92.6	Transformer	Manual	-
PVT-Tiny [6]	13.2	1.9	75.1	-	Transformer	Manual	-
ViTAS-C [27]	5.6	1.3	74.7	91.6	Transformer	Auto	32
AutoFormer-Ti [1]	5.7	1.3	74.7	92.6	Transformer	Auto	24
GLiT-Ti [13]	7.2	1.4	76.3	-	Hybrid	Auto	N/A
T-Razor-Ti (Ours)	**5.9**	**1.4**	**75.5**	**92.9**	Transformer	Auto	**0.4**
ResNet-50* [54]	25.6	4.1	80.2	-	CNN	Manual	-
RegNetY-4GF [56]	20.6	-	79.4	-	CNN	Manual	-
DeiT-S [9]	22.1	4.7	79.9	95.0	Transformer	Manual	-
ViT-S/16 [8]	22.1	4.7	78.8	-	Transformer	Manual	-
PVT-Small [6]	24.5	3.8	79.8	-	Transformer	Manual	-
Swin-T [26]	29.0	4.5	81.3	-	Transformer	Manual	-
TNT-S [25]	23.8	5.2	81.5	95.7	Transformer	Manual	-
CPVT-S [24]	23.0	-	81.5	95.7	Transformer	Manual	-
T2T-ViT_t-14 [10]	21.5	-	81.7	-	Transformer	Manual	-
ViTAS-F [27]	27.6	6.0	80.5	95.1	Transformer	Auto	32
AutoFormer-S [1]	22.9	5.1	81.7	95.7	Transformer	Auto	24
GLiT-S [13]	24.6	4.4	80.5	-	Hybrid	Auto	N/A
T-Razor-S (Ours)	**22.3**	**5.1**	**82.2**	**95.9**	Transformer	Auto	**0.4**
ResNet-152* [54]	60.2	11.5	81.9	-	CNN	Manual	-
RegNetY-16GF [57]	83.6	15.9	80.4	-	CNN	Manual	-
ViT-B/16 [8]	86	18	79.7	-	Transformer	Manual	-
PVT-Large [6]	61.0	9.8	81.7	-	Transformer	Manual	-
DeiT-B [9]	86.0	18.0	81.8	95.6	Transformer	Manual	-
CPVT-B [24]	88.0	-	82.3	-	Transformer	Manual	-
TNT-B [25]	65.5	14.1	82.9	96.3	Transformer	Manual	-
Swin-B [26]	88.0	15.4	83.5	-	Transformer	Manual	-
T2T-ViT-24 [10]	64.1	-	82.6	-	Transformer	Manual	-
GLiT-B [13]	96.0	17.0	82.3	-	Hybrid	Auto	N/A
AutoFormer-B [1]	54.0	11.0	82.4	95.7	Transformer	Auto	24
T-Razor-B (Ours)	**53.8**	**11.6**	**82.3**	**95.6**	Transformer	Auto	**0.4**

* Denotes the results reported in [2].

TABLE II
COMPARISON RESULTS ON THE PIT SEARCH SPACE

Models	#Param (M)	FLOPs (B)	Top-1 (%)	Top-5 (%)
PiT-Ti† [2]	4.9	0.7	73.8	91.7
PiT-Ti$_{rand}$	4.9	0.7	69.7	89.1
TF-TAS-Ti [31]	4.6	0.6	73.7	91.7
T-Razor-Ti (Ours)	**4.9**	**0.7**	**74.2**	**92.0**
PiT-XS† [2]	10.6	1.4	78.2	94.0
PiT-XS$_{rand}$	10.5	1.8	74.8	92.2
TF-TAS-XS [31]	10.0	1.8	77.7	93.8
T-Razor-XS (Ours)	**10.1**	**1.8**	**78.0**	**94.0**
PiT-S† [2]	23.5	2.9	79.9	94.4
PiT-S$_{rand}$	24.2	3.3	75.1	92.4
TF-TAS-S [31]	23.8	3.2	80.5	94.9
T-Razor-S (Ours)	**22.3**	**3.1**	**80.4**	**94.9**

† Indicates the results we reproduce.

TABLE III
SEARCHED RESULTS ON THE SHUNTED VIT SEARCH SPACE

Models	#Param (M)	FLOPs (B)	Top-1 (%)
Shunted-T [3]	11.5	2.1	79.8
T-Razor-Ti (Ours)	**11.1**	**1.9**	**80.1**
Shunted-S [3]	22.4	4.9	82.9
T-Razor-S (Ours)	**22.3**	**5.0**	**83.0**
Shunted-B [3]	39.6	8.1	84.0
T-Razor-B (Ours)	**39.1**	**7.6**	**84.3**

space with that on AutoFormer and PiT search spaces which are applied in the vision task, there are several empirical findings: (1) Optimal architectures tend to maximize their depths; (2) Larger MLP ratio is generally better for shallow layers of large ViTs; (3) A higher proportion of small number of head is shown in the deep layer. Moreover, compared to the manually designed Transformers, we find that the Transformers searched by our T-Razor are more likely to transfer the amount of wasted computation on some redundant dimensions to other dimensions with insufficient computation.

Results on BERT Search Space: Given the fact that our DSS++ evaluates Transformer architecture from a basic perspective and that Transformer has been widely applied in NLP research filed [4], then it is interesting and worth-while to investigate how

TABLE IV
COMPARISON GLUE TEST RESULTS ON THE BERT SEARCH SPACE USING THE GLUE EVALUATION SERVER

System	#Params (M)	#FLOPs (B)	MNLI-m 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	**Average**
Pre-OpenAI [63]	-	-	80.6	66.1	82.3	93.2	35.0	81.0	86.0	61.7	73.2
OpenAI GPT [63]	-	-	82.1	70.3	87.4	91.3	45.4	80.0	82.3	56.0	74.4
BERT_{base} [4]	109	22.5	84.6	71.2	90.5	93.5	52.1	85.5	88.9	66.4	79.1
T-Razor (Ours)	**100 - 110**	**20 - 25**	**84.3**	**89.6**	**90.6**	**89.9**	**53.0**	**86.3**	**89.2**	**66.9**	**81.2**

TPE constraints the searched model size in 95-110 m.

our metric works to find optimal networks in NLP setting. To check the versatility of the proposed method on NLP scenarios, we experiment our DSS++ with six NLP tasks on a BERT search space based on [4]. Specifically, the BERT search space includes three dimensions: block number, intermediate size of MLP, and head number. Our T-Razor searches a stack of Transformer encoder blocks, which includes the block number sampled in {6, 8, 10, 12}. In each Transformer encoder block, the intermediate size of MLP is sampled in {512, 768, 1024, 3072} and the head number is sampled in {4, 8, 12, 16}. In this case, the number of candidates within the BERT search space is about 1.6×10^{12}, and we follow the pipeline and evaluation metric of down-stream task scenarios in [4].[1] T-Razor costs less than 0.4 GPU days to complete the search process on the BERT search space. As for evaluation, the NLP tasks include MNLI, QQP, QNLI, CoLA, MRPC and RET, which is selected from The General Language Understanding Evaluation (GLUE) benchmark [64].[2] GLUE is a collection of nine language understanding tasks, including question answering, linguistic acceptability, sentiment analysis, text similarity, paraphrase detection, and natural language inference. In our experiments, we follow the common practical [4], [64]: F1 scores are reported for QQP and MRPC, Spearman correlations are reported for STS-B, and accuracy scores are reported for the other tasks.

As demonstrate in Table IV, the architectures searched by our T-Razor achieve competitive performance among the compared models [4], [63], slightly higher accuracy on a few tasks. Moreover, T-Razor requires less than 0.4 GPU days to search architectures for each NLP tasks, which is much more efficiency than the manual design manner. These experimental results validates that our T-Razor can be generalized to other Transformer search space for NLP tasks. For the NLP task, we observe that the architectures searched by our T-Razor share some commonalities: (1) Tend to have large intermediate size of MLP in the shallow blocks; (2) Tend to have small head number in the deep blocks; (3) Deeper is not always better.

C. Further Investigation

Ablation Studies: First, we conduct additional experiments to comprehensively study the contributions of the key components in our T-Razor on the PiT search space. Noted that the Random + DSS and BE (Block-wise Evolution) + DSS++ in Table V correspond to TF-TAS [31] and our T-Razor, respectively. As listed in Table V, our DSS++ with random search strategy (Random + DSS++) outperforms TF-TAS from 73.7 % to 74.2 %, and our block-wise evolution search strategy with DSS (BE + DSS) is also better than TF-TAS and compresses the time by an additional 20%.

TABLE V
CLASSIFICATION ACCURACIES ON IMAGENET OF THE SEARCHED OPTIMAL TRANSFORMER ARCHITECTURES VIA DIFFERENT COMPONENTS ON THE PIT SEARCH SPACE

Methods	#Param (M)	Top-1 (%)	GPU Days
Random + DSS	4.6	73.7	0.5
Random + DSS++	4.8	74.1	0.5
BE + DSS	4.8	73.9	0.4
BE + DSS++	4.9	74.2	0.4

Random and BE denote the random search and block-wise evolution search strategies, respectively.

TABLE VI
EVALUATION RESULTS OF THE PROPOSED ZERO-COST PROXY ON THE STATE-OF-THE-ART TRANSFORMER ARCHITECTURES

Models	#Param (M)	Top-1 (%)	DSS++	Diversity	Saliency
PiT [2]					
PiT-Ti	4.9	73.8	2	8.6×10^3	2.1×10^4
PiT-XS	10.6	78.2	4	1.1×10^4	2.3×10^4
PiT-S	23.5	80.5	6	1.7×10^4	2.7×10^4
T2T-ViT [10]					
T2T-ViT-7	4.3	71.7	2	2.5×10^4	8.0×10^4
T2T-ViT-10	5.9	75.2	4	3.1×10^4	9.8×10^4
T2T-ViT-12	6.9	76.5	6	3.8×10^4	1.2×10^5
T2T-ViT-14	21.5	81.5	8	4.7×10^4	5.3×10^5
T2T-ViT-19	39.2	81.9	10	8.3×10^4	1.0×10^6
T2T-ViT-24	64.1	82.3	12	1.4×10^5	2.2×10^6
XCiT [65]					
XCiT-tiny	12.0	79.4	2	1.8×10^3	3.2×10^4
XCiT-small	48.0	82.6	4	3.5×10^3	8.8×10^4
XCiT-medium	84.0	82.7	6	4.5×10^4	9.0×10^4
XCiT-large	189.0	82.9	8	3.2×10^4	2.1×10^5

On Evaluating Popular Architectures: To further investigate the versatility of our DSS++, we conduct additional evaluation experiments on three *state-of-the-art* Transformer architectures: PiT [2], T2T-ViT [10], and XCiT [65].

As listed in Table VI, our DSS++ can evaluate the correct rank of architectures sampled from different Transformer search

[1]https://github.com/google-research/bert
[2]https://gluebenchmark.com/leaderboard/

TABLE VII
CLASSIFICATION RESULTS (%) ON DOWNSTREAM DATASETS

Models	#Param	ImageNet	C-10	C-100
ViT-B/16 [8]	86M	77.9	98.1	87.1
DeiT-B [9]↑ 384	86M	83.1	99.1	90.8
AutoFormer-S [1]↑ 384	23M	83.4	99.1	91.1
T-Razor-S↑ 384 (Ours)	**23M**	**83.7**	**99.1**	**91.3**

↑ 384 means 384 × 384 resolution.

TABLE VIII
COCO DETECTION RESULTS ON DEFORMABLE-DETR

Backbone	#Param (M)	Avg. Precision at IOU AP	AP_{50}	AP_{75}
ResNet-50 [54]	41.0	41.5	60.5	44.3
ViT-S [8]	34.9	36.9	57.0	38.0
PiT-S [2]	39.3	39.4	58.8	41.5
Shunted-S [3]	40.3	42.8	64.1	47.5
T-Razor-PiT (Ours)	**41.8**	**42.0**	**62.4**	**45.1**
T-Razor-SSA (Ours)	**40.9**	**43.5**	**63.7**	**47.8**

space. It is worthwhile and important to note that the values obtained by the DSS++ across different search spaces are not comparable directly, as they depend on the number of the sampled subset. Besides, the values of the diversity and saliency scores obtained across different search spaces are not comparable too. It might be caused by several factors. For example, the different ways of model initialization, and the search space itself contains different modules and makes it difficult to achieve a fair comparison.

Transfer Learning Results: To test the transferability of the searched optimal Transformer architectures, we conduct some transfer learning experiments. We follow the same settings as DeiT [9] and finetune the T-Razor-S (see Table I) in ImageNet [59], CIFAR-10 (C-10) and CIFAR-100 (C-100) [60]. The results are listed in Table VII. As we observe that the optimal Transformer architectures found by T-Razor in a training-free manner have a similar fine-tuning performance as that of the architectures searched by AutoFormer [1]. Compared with ViT-B/16 [8] and DeiT-B [9], our T-Razor-S achieves better performance with a smaller #Param in in ImageNet [59], CIFAR-10 and CIFAR-100 [60].

Object Detection: To ascertain the versatility of our T-Razor across various tasks, we conduct transfer experiments on the detection task with COCO 2017 dataset [61]. We follow the training and testing settings that adopted in [67] to train different backbones including ResNet-50 [54], ViT-S [8], PiT-S [2], Shunted-S [3] and the models searched by our T-Razor (T-Razor-PiT, T-Razor-SSA) on different Transformer search spaces. Specifically, these backbones are pretrained on ImageNet-1 K and fine-tuning on COCO 2017. To accommodate transformer-based backbones, we follow the strategy in [2] to reduce image resolution by half during both training and testing for all backbones.

The measured AP scores are presented in Table VIII. The results indicates that the architectures searched by T-Razor achieve competitive performance in the detection task. Furthermore, based on the different Transformer search spaces, the architectures searched by our T-Razor outperform the manually designed Transformers (i.e.: ViT-S [8], PiT-S [2] and Shunted-S [3]). We also find that the architectures from the Shunted ViT space seem to be more suitable for the detection task than the ones from PiT search space. This finding is consistent with the conclusion drawn by Ren et al [3] which proposes shunted self-attention (SSA) to capture multi-scale features for detection task. These results confirm that the proposed T-Razor can be generalized across various Transformer search spaces, including for detection task.

TABLE IX
COMPARISON OF THE ADE20 K SEGMENTATION RESULTS OF DIFFERENT BACKBONES WITH SEMANTIC FPN FRAMEWORK

Backbone	#Param (M)	FLOPs (G)	mIOU (%)
ResNet-50 [54]	28.5	183	36.7
Swin-T [26]	31.9	182	41.5
PVT-S [6]	28.2	116	39.8
Twin-S [66]	28.3	114	43.2
Shunted-S [3]	26.1	183	48.2
T-Razor-SSA (Ours)	26.4	186	48.4

Semantic Segmentation: We further evaluate the result of our T-Razor for semantic segmentation on ADE20K [62] benchmark with the Shunted ViT search space. We report the mIOU without multi-scale testing. And we take Semantic FPN [68] as the main framework and follow the default practices [68] and `mmsegmentation` [69] (e.g., crop size is 512×512, use AdamW with weight decay of 1×10^{-4}, set the learning rate as 1×10^{-4}, and train the model for 80 K iterations on 8 NVIDIA V100 GPUs).

As shown in Table IX, our method can also find the optimal backbone for the semantic segmentation task. Compared with the expert manual designed transformers [3], [26], [66], it takes us less than 1 GPU day to seek for the backbone network to achieve competitive, or even better, performance. Besides, we also note that T-Razor-SSA also outperforms Shunted-S when the model is trained without being pre-trained on ImageNet: T-Razor-SSA and Shunted-S achieves 44.1% mIOU and 43.0% mIOU, respectively.

Comparison of Zero-cost Proxies: For full investigation, we compare our DSS++ with alternative *state-of-the-art* zero-cost proxies [21], [22], [23], [28], [29], [30], [31], [37], [38] for CNN search spaces. For the general Transformers do not have convolutional layers, when transferring the proxies originally designed for CNNs to the Transformers, only the results obtained in the linear layer part are considered. To build a reliable testbed to evaluate these zero-cost proxies, we build a large proxy TAS benchmark based on the AutoFormer search space. We denote the search space of AutoFormer [1] as $\mathcal{S}_A$ for simplicity. Empirically, Chen et al. [1] find that, the subnet from $\mathcal{S}_A$ with its weights inherited from the pre-trained supernet can achieve the performance comparable to that of the retrained one. Building on this observation, we sample 3,000 subsets from $\mathcal{S}_A$ and obtain their accuracies by inheriting their weights from the pre-trained

TABLE X
KENDALL τ VALUES BETWEEN VARIOUS TRAINING-FREE EVALUATION METRICS AND THE FINAL CLASSIFICATION ACCURACY ON THE INHERIT NETWORKS RANDOMLY SAMPLED FROM THREE PRE-TRAINED AUTOFORMER SUPERNETS

Proxy	Venue	#Param (M)		
		5 - 7	15 - 19	23 - 25
SNIP [38]	ICLR-19	0.481	0.028	-0.282
GraSP [21]	ICLR-20	0.053	-0.022	-0.029
TE-score [22]	ICLR-21	-0.039	-0.248	-0.075
NASWOT [23]	ICML-21	0.378	0.171	0.208
NTK-trace [28]	NeurIPS-22	-0.182	-0.211	-0.106
GradSign [29]	ICLR-21	0.232	0.176	0.128
MGM [30]	ICLR-21	0.402	0.429	0.245
Grad-norm [37]	ICLR-21	0.128	0.036	0.338
DSS [31]	CVPR-22	0.697	0.615	0.306
DSS++ (Ours)	-	**0.716**	**0.664**	**0.431**

TABLE XI
CLASSIFICATION ACCURACIES ON IMAGENET OF THE SEARCHED OPTIMAL TRANSFORMER ARCHITECTURES VIA DIFFERENT ZERO-COST PROXIES FROM THE AUTOFORMER SEARCH SPACE

Proxy	#Param (M)	FLOPs (B)	Top-1 (%)	Top-5 (%)
SNIP [38]	5.8	1.4	74.8	92.7
GraSP [21]	5.5	1.3	74.3	92.2
TE-score [22]	5.6	1.3	74.6	92.6
NASWOT [23]	5.9	1.5	74.8	92.8
NTK-trace [28]	5.8	1.4	74.9	92.5
GradSign [29]	6.0	1.5	74.9	92.6
MGM [30]	6.0	1.5	75.0	92.6
Grad-norm [37]	5.9	1.4	74.8	92.8
DSS [31]	5.9	1.4	75.3	92.8
DSS++ (Ours)	**5.9**	**1.4**	**75.5**	**92.9**

supernet. Without loss of generality, we sample the subnets with the amount of parameter in three common ranges: 5 M $\sim$ 7M, 15 M $\sim$ 19 M, and 23 M $\sim$ 25 M, to form a large proxy TAS benchmark. With this benchmark, we compare our DSS++ with several cutting-edge zero-cost proxy methods, like SNIP [38], GraSP [21], NASWOT [23], TE-score [22], NTK-trace, [28] GradSign [29], MGM [30], Grad-norm [37], and DSS [31].

The results of Kendall τ [70] are illustrated in Table X . Overall, the relative ranking of the proxies is: Ours $>$ DSS $>$ MGM $>$ SNIP $>$ NASWOT $>$ GradSign $>$ Grad-norm $>$ NTK-trace $>$ TE-score $>$ GraSP. Our DSS++ outperforms the others in ranking various Transformer architectures. The results also provide practical insights to design an effective zero-cost proxy for TAS: 1) Both MSA and MLP should be taken into consideration to rank Transformer effectively, which is verified by DSS and our DSS++. 2) DSS++ adopts the ranks of diversity and saliency to eliminate the dimensional differences between diversity and saliency, which improves the Kendall τ results. 3) DSS++ achieves a better Kendall τ result between 23 M $\sim$ 25M, which coincides with the conclusion drawn in Section III-D1. 4) Based on the results of SNIP [38], MGM [30], DSS [31] and our DSS++, it is clear that the gradient matrix from the initialized Transformer network contains rich information to evaluate the corresponding model. 5) Based on the performance of GraSP [21] and TE-score [22], we find that: despite its practical value in CNNs [21], the Hessian matrix of Transformer is not easy to use and requires further efforts.

TABLE XII
AFFECTION OF DIFFERENT INITIALIZATION SEEDS ON THE EVALUATION RESULTS OF VARIOUS ZERO-COST PROXIES

Proxy	Random Seed				AVG	STD
	0	1	2	3		
SNIP [38]	0.481	0.530	0.486	0.507	0.501	0.019
GraSP [21]	0.053	0.126	0.138	0.152	0.117	0.038
TE-score [22]	-0.039	-0.003	-0.04	0.013	-0.017	0.023
NASWOT [23]	0.378	0.332	0.394	0.421	0.381	0.032
DSS [31]	0.697	0.697	0.697	0.697	0.697	0
DSS++ (Ours)	**0.716**	**0.716**	**0.716**	**0.716**	**0.716**	**0**

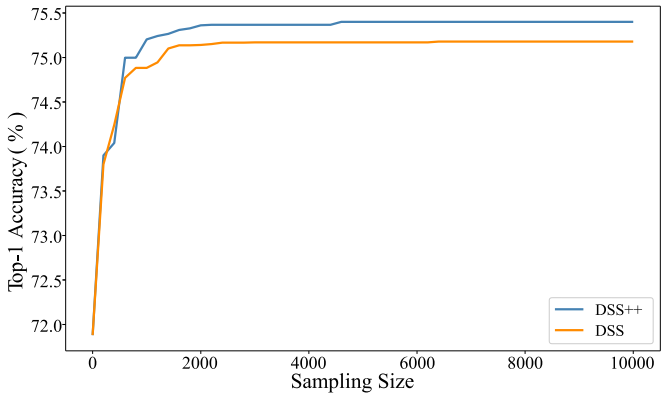


Fig. 5. Comparison of the impacts of sampling size on DSS++ and DSS. The results of DSS++ and DSS are obtained under the same settings.

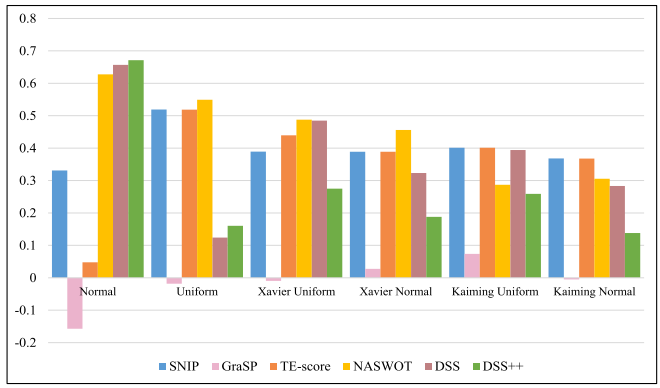
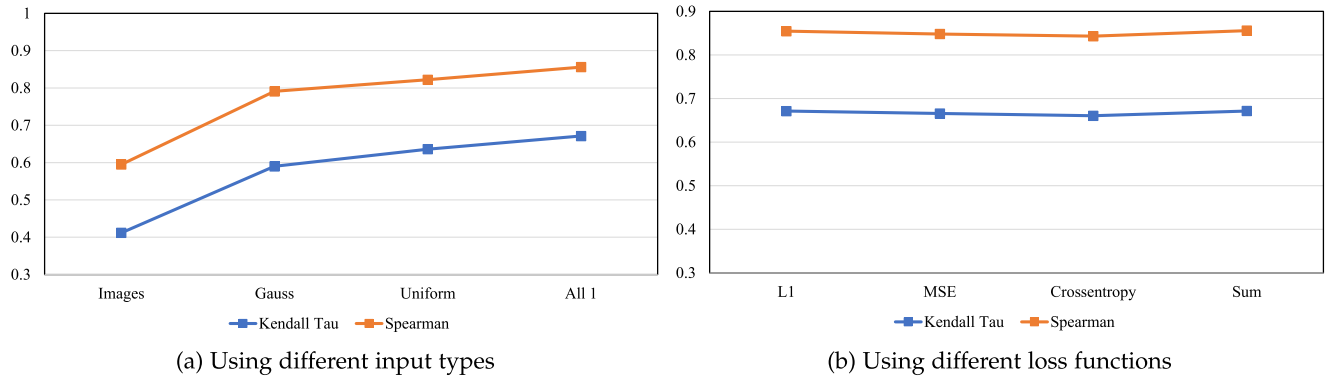


Fig. 6. Ablation study of various training-free indicators when different initialization methods are applied.

In Table XI, we list the classification result of the searched optimal networks with the help of several zero-cost proxy methods. As indicated in Table XI, the Transformer architecture searched by our DSS++ is better than that of the cutting-edge counterparts [21], [22], [23], [28], [29], [30], [31], [37], [38], which have achieved competitive performance in popular CNN search spaces. The results also verify the necessity to design a Transformer-oriented performance indicator.



(a) Using different input types (b) Using different loss functions

Fig. 7. Ranking correlation of DSS++ when we apply: (a) different input types (a) and (b) different loss functions.

Consistency across Different Random Seeds: To check the stability of various proxies, we generate the results with four random seeds and analyze the corresponding values. For simplicity, the experiments are conducted on $5 \sim 7$M subnets of the proposed large proxy TAS benchmark.

As listed in Table XII, there are certain fluctuations on several proxies under different seeds. Our DSS++ inherits the invariance to different seeds from DSS (as mentioned in Section III-D). As the most of proxies do not use all the input data, the different seeds influence the sampling of input data, making the calculation of these proxies unstable. DSS++ uses the model inputs with each pixel being 1, which do not affected by the different seeds. Therefore, DSS++ has a certain stability with respect to random seeds.

Impacts of The Sampling Size: To determine the impacts of the sampling size during the search process, we have conducted an experiment on DSS++ and DSS. In this experiment, we start with a sampling size of 1 and incrementally obtain the searched optimal sub-networks at intervals of 200 samples on the AutoFormer search space. It is worth noting that the cost of retraining each searched sub-network is too high. Therefore, to ensure fairness, we adopted the method of inheriting the super-network weights to obtain the top-1 accuracy of the searched sub-networks. The experimental results are shown in Fig. 5, which indicate that overall, the larger the sampling number, the better the results obtained by DSS++ and DSS. Compared to DSS, DSS++ demonstrates a faster convergence speed, meaning that DSS++ can identify superior structures more quickly. However, as the sampling number increases, the potential for improving the searched results become smaller and smaller, and the search efficiency decline. Therefore, considering the balance between search performance and search efficiency, we followed our first version [31] to set the sampling size to 8000.

Impacts of The Initialization Manners: Since our DSS++ is designed to evaluate Transformers in the initialized state, then the model initialization methods play an important role. We compare different training-free indicators [21], [22], [23], [31], [38] with six common initialization approaches. Without loss of generality, we conducted the experiment on the collected tiny TAS benchmark.

As shown in Fig. 6, the initialization methods have a significant influence on the tested indicators. One of the main reasons is that these indicators rely on weight and gradient information. Among the six initialization methods, the Normal initialization achieves better results for all indicators, and our DSS++ outperforms other counterparts with the Normal initialization. Meanwhile, it also shows that our DSS++ is inferior to the some training-free indicators in other initialization methods. We find some possible explanations based on the conclusions presented in [71]. As indicated in [71], the expressivity of a model depends heavily on its linear regions and the Normal initialization prevents collapse of regions at initialization, which occurs when all biases are uniquely zero. Therefore, the Normal initialization can better reflect the expressivity of the model that is related to the saliency and diversity, and accordingly is more suitable for our DSS++.

Effects of Input Types: As we demonstrate in Section III-D1, the model inputs with each pixel being 1 ('All-1') could make the model outputs to be the model weights that used to calculate DSS++. We conducted further tests on real images, Gaussian noise, and uniform noise as input types for the proposed tiny TAS benchmark. Fig. 7(a) shows that 'All-1' achieves the best correlation results while the other three input types have a negative effect on the weights leading to decreased correlation. Among these input types, the uniform noise yields results closer to those of 'All-1'. We argue that the uniform noise affects the model weights relatively uniformly, which changes the model weights relatively little. Overall, since DSS++ relies on the model weights to obtain the ranks of the diversity of MSA and the saliency of MLP, it is more appropriate to use inputs with each pixel being 1.

The Loss Functions in Calculating DSS++: In our implementation the DSS++ of one Transformer is obtained with a single forward-back propagation. As DSS++ adopts the inputs constructed with each pixel being 1, the model outputs are summed to back-propagating without the labels. To test DSS++ with other loss functions on the proposed tiny TAS benchmark, we construct the labels of these constructed inputs being 1 without loss of generality.

As demonstrates in Fig. 7(b), different loss functions have little impact on DSS++. We argue that this may be due to the

fact that inputs with each pixel being 1 make the calculation of the loss only involves the model gradient without the interference of real images. Therefore, after only one forward-back propagation, the choice of loss functions has limited impact on the gap between different DSS++ scores of different models. The experimental results suggest that our DSS++ focuses more on the impact of differences in model architectures, while the loss function may not have a significant difference in gain among different models.

V. Conclusion

In this paper, we propose T-Razor, a novel training-free Transformer architecture search with zero-cost proxy guided evolution that accelerates the efficiency of TAS. We introduce a Transformer-oriented performance indicator, the ranks of synaptic diversity and saliency denoted as DSS++, which is based on MSAs and MLPs of Transformer architectures. By leveraging the ranks of synaptic diversity of MSAs and the synaptic saliency of MLPs, we effectively estimate the performances of different Transformer architectures. We also propose a block-wise evolution search method that combines with the DSS++ to further enhance the search efficiency and improve the searched results. Compared to other cutting-edge TAS methods, our block-wise evolution search guided by the DSS++ achieves competitive performance among popular Transformer search spaces involving vision tasks as well as NLP tasks. Furthermore, we significantly improve the efficiency of TAS, requiring only 0.4 GPU days to find relatively optimal Transformer architecture, compared to 24 GPU days with existing counterparts.

As for future work, we will investigate how to accelerate the architecture design for large models (e.g., the amount of parameters is over 1 billion) and for multi-modality models.

References

[1] M. Chen, H. Peng, J. Fu, and H. Ling, "AutoFormer: Searching transformers for visual recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 12250–12260.

[2] B. Heo, S. Yun, D. Han, S. Chun, J. Choe, and S. J. Oh, "Rethinking spatial dimensions of vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 11916–11925.

[3] S. Ren, D. Zhou, S. He, J. Feng, and X. Wang, "Shunted self-attention via multi-scale token aggregation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 10843–10852.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Int. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2019, pp. 4171–4186.

[5] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[6] W. Wang et al., "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 548–558.

[7] J. Hu et al., "ISTR: End-to-end instance segmentation with transformers," 2021, *arXiv:2105.00637*.

[8] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Representations*, 2021, pp. 1–12.

[9] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 10347–10357.

[10] L. Yuan et al., "Tokens-to-token ViT: Training vision transformers from scratch on imagenet," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 538–547.

[11] D. So, Q. Le, and C. Liang, "The evolved transformer," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 5877–5886.

[12] H. Wang et al., "HAT: Hardware-aware transformers for efficient natural language processing," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 7675–7688.

[13] B. Chen et al., "GLiT: Neural architecture search for global and local image transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 12–21.

[14] A. Chavan, Z. Shen, Z. Liu, Z. Liu, K.-T. Cheng, and E. P. Xing, "Vision transformer slimming: Multi-dimension searching in continuous optimization space," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 4921–4931.

[15] Z. Guo et al., "Single path one-shot neural architecture search with uniform sampling," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 544–560.

[16] X. Dong and Y. Yang, "One-shot neural architecture search via self-evaluated template network," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 3680–3689.

[17] X. Zheng et al., "MIGO-NAS: Towards fast and generalizable neural architecture search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 9, pp. 2936–2952, 2021.

[18] X. Zheng et al., "Evolving fully automated machine learning via life-long knowledge anchors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 9, pp. 2936–2952, Sep. 2021.

[19] M. Chen et al., "Searching the search space of vision transformer," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 8714–8726.

[20] H. Liu, K. Simonyan, and Y. Yang, "Darts: Differentiable architecture search," in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–13.

[21] C. Wang, G. Zhang, and R. Grosse, "Picking winning tickets before training by preserving gradient flow," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 1–11.

[22] W. Chen, X. Gong, and Z. Wang, "Neural architecture search on imagenet in four GPU hours: A theoretically inspired perspective," in *Proc. Int. Conf. Learn. Representations*, 2021, pp. 1–15.

[23] J. Mellor, J. Turner, A. Storkey, and E. J. Crowley, "Neural architecture search without training," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 7588–7598.

[24] X. Chu et al., "Conditional positional encodings for vision transformers," 2021, *arXiv:2102.10882*.

[25] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, and Y. Wang, "Transformer in transformer," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 15908–15919.

[26] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9992–10002.

[27] X. Su et al., "Vision transformer architecture search," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 39–157.

[28] Y. Shu, S. Cai, Z. Dai, B. C. Ooi, and B. K. H. Low, "NASI: Label- and data-agnostic neural architecture search at initialization," in *Proc. Int. Conf. Learn. Representations*, 2022, pp. 1–11.

[29] Z. Zhang and Z. Jia, "Gradsign: Model performance inference with theoretical insights," in *Proc. Int. Conf. Learn. Representations*, 2022, pp. 1–13.

[30] J. Xu, L. Zhao, J. Lin, R. Gao, X. Sun, and H. Yang, "Knas: Green neural architecture search," *Int. Conf. Mach. Learn.*, pp. 1613–11625, 2021.

[31] Q. Zhou et al., "Training-free transformer architecture search," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 10894–10903.

[32] Y. Xu et al., "Evo-ViT: Slow-fast token evolution for dynamic vision transformer," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 2964–2972.

[33] Z. Wei, H. Pan, X. Niu, and D. Li, "OVO: One-shot vision transformer search with online distillation," 2022, *arXiv:2212.13766*.

[34] Y.-L. Liao, S. Karaman, and V. Sze, "Searching for efficient multi-stage vision transformers," 2021, *arXiv:2109.00642*.

[35] C. Gong et al., "Nasvit: Neural architecture search for efficient vision transformers with gradient conflict aware supernet training," in *Proc. Int. Conf. Learn. Representations*, 2022, pp. 1–13.

[36] M. Javaheripi et al., "LiteTransformerSearch: Training-free neural architecture search for efficient language models," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, pp. 24254–24267.

[37] M. S. Abdelfattah, A. Mehrotra, Ł. Dudziak, and N. D. Lane, "Zero-cost proxies for lightweight NAS," in *Proc. Int. Conf. Learn. Representations*, 2021, pp. 1–11.

[38] N. Lee, T. Ajanthan, and P. H. Torr, "SNIP: Single-shot network pruning based on connection sensitivity," in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–12.

[39] M. Lin et al., "Zen-NAS: A zero-shot NAS for high-performance deep image recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 337–346.

[40] Y. Shu, Z. Dai, Z. Wu, and B. K. H. Low, "Unifying and boosting gradient-based training-free neural architecture search," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 33001–33015.
[41] R. Mohan et al., "Neural architecture search for dense prediction tasks in computer vision," *Int. J. Comput. Vis.*, vol. 131, no. 7, pp. 1784–1807, 2023.
[42] X. Dong and Y. Yang, "NAS-Bench-201: Extending the scope of reproducible neural architecture search," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 1–13.
[43] D. Zhou et al., "DeepViT: Towards deeper vision transformer," 2021, *arXiv:2103.11886*.
[44] Y. Dong, J.-B. Cordonnier, and A. Loukas, "Attention is not all you need: Pure attention loses rank doubly exponentially with depth," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 2793–2803.
[45] M. Fazel, "Matrix rank minimization with applications," Ph.D. dissertation, PhD Thesis, Dept. Elect. Eng., Stanford Univ., 2002.
[46] H. Tanaka, D. Kunin, D. L. Yamins, and S. Ganguli, "Pruning neural networks without any data by iteratively conserving synaptic flow," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 6377–6389.
[47] R. Reed, "Pruning algorithms-a survey," *IEEE Trans. Neural Netw.*, vol. 4, no. 5, pp. 740–747, Sep. 1993.
[48] F. Meng et al., "Pruning filter in filter," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 17629–17640.
[49] W. Wang and Z. Tu, "Rethinking the value of transformer components," in *Proc. 28th Int. Conf. Comput. Linguistics*, 2020, pp. 6019–6029.
[50] P. Michel, O. Levy, and G. Neubig, "Are sixteen heads really better than one?," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 14014–14024.
[51] E. Voita, D. Talbot, F. Moiseev, R. Sennrich, and I. Titov, "Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 5797–5808.
[52] H. You et al., "Drawing early-bird tickets: Towards more efficient training of deep networks," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 1–12.
[53] N. Lee, T. Ajanthan, S. Gould, and P. H. Torr, "A signal propagation perspective for pruning neural networks at initialization," in *Proc. Int. Conf. Learn. Representations*, 2020, pp. 1–12.
[54] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[55] A. Howard et al., "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 1314–1324.
[56] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of tricks for image classification with convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 558–567.
[57] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollár, "Designing network design spaces," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10425–10433.
[58] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–10.
[59] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.
[60] A. Krizhevsky, G. Hinton et al., "Learning multiple layers of features from tiny images," Univ. Toronto, Tech. Rep., 2009.
[61] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.
[62] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ADE20K dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5122–5130.
[63] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding with unsupervised learning," OpenAI, Tech. Rep., 2018.
[64] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "Glue: A multi-task benchmark and analysis platform for natural language understanding," in *Proc. Int. Conf. Learn. Representations*, 2019, pp. 1–13.
[65] A. Ali et al., "XCiT: Cross-covariance image transformers," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 20014–20027.
[66] X. Chu et al., "Twins: Revisiting the design of spatial attention in vision transformers," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 9355–9366.
[67] Z. Xizhou, S. Weijie, L. Lewei, L. Bin, W. Xiaogang, and D. Jifeng, "Deformable DETR: Deformable transformers for end-to-end object detection," in *Proc. Int. Conf. Learn. Representations*, 2021, pp. 1–11.
[68] A. Kirillov, R. Girshick, K. He, and P. Dollár, "Panoptic feature pyramid networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6399–6408.
[69] M. Contributors, "MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark," 2020. [Online]. Available: https://github.com/open-mmlab/mmsegmentation
[70] M. G. Kendall, "A new measure of rank correlation," *Biometrika*, vol 30, no. 1/2, pp. 81–93, 1938.
[71] B. Hanin and D. Rolnick, "Complexity of linear regions in deep networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 2596–2604.

Qinqin Zhou received the BS degrees in mathematics and applied mathematics from the Qingdao Technological University, Shandong, China, in 2016, and the MS degree from the School of Computer Science and Technology, Huaqiao University, Xiamen, China, in 2019. He is currently working toward the PhD degree with the School of Information Science and Engineering, Xiamen University, Xiamen, China. His current research interests include pattern recognition, machine learning, and computer vision

Kekai Sheng received the BEng degree in telecommunication engineering from the University of Science and Technology Beijing, in 2014, and the PhD degree from the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences, in 2019. He is currently a researcher engineer with Momenta. His research interests include image quality evaluation, domain adaptation, and AutoML.

Xiawu Zheng received the MS degree in computer science from the School of Information Science and Engineering, Xiamen University, Xiamen, China, in 2018. He is currently working toward the PhD degree from the School of Information Science and Engineering, Xiamen University, China. His research interests include computer vision and machine learning. He was involved in automatic machine learning.

Ke Li received the master's degree from the Department of Artificial Intelligence, School of Informatics, Xiamen University, China. His research interests fall in the area of 3D vision and unsupervised learning. He is now a researcher with the Tencent Youtu Lab.

Yonghong Tian (Fellow, IEEE) is currently the Boya distinguished professor with the School of Electronics Engineering and Computer Science, Peking University, China, and also the deputy director with the Peng Cheng Laboratory, Artificial Intelligence Research Center, Shenzhen, China. His research interests include computer vision, multimedia Big Data, and brain-inspired computation. He is the author or co-author of more than 180 technical articles in refereed journals, such as *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *IEEE Transactions on Neural Networks and Learning Systems*, *IEEE Transactions on Image Processing*, *IEEE Transactions on Multimedia*, *IEEE Transactions on Circuits and Systems for Video Technology*, *IEEE Transactions on Knowledge and Data Engineering*, *IEEE Transactions on Parallel and Distributed Systems* and the *ACM Computing Surveys*, the *ACM Transactions on Information Systems*, the *ACM Transactions on Multimedia Computing, Communications, and Applications* and conferences, such as the NeurIPS/CVPR/ICCV/AAAI/ACMMM/WWW. He is a senior member of the CIE and CCF and a member of the ACM. He has been the steering member of the IEEE ICME since 2018 and the IEEE International Conference on Multimedia Big Data (BigMM) since 2015, and a TPC member of more than ten conferences, such as CVPR, ICCV, ACM KDD, AAAI, ACM MM, and ECCV. He was a recipient of the Chinese National Science Foundation for Distinguished Young Scholars, in 2018, two National Science and Technology Awards, three Ministerial-Level Awards in China, the 2015 EURASIP Best Paper Award for the Journal on Image and Video Processing, and the Best Paper Award of the IEEE BigMM 2018. He was/is an associate editor of *IEEE Transactions on Circuits and Systems for Video Technology* since 2018, *IEEE Transactions on Multimedia* from 2014 to 2018, the *IEEE Multimedia Magazine* since 2018, and the *IEEE Access* since 2017. He also Co-Initiated the BigMM. He has served as the TPC co-chair for BigMM 2015 and the Technical Program co-chair for the IEEE ICME 2015, the IEEE ISM 2015, and the IEEE MIPR 2018/2019, and the general co-chair for the IEEE MIPR 2020.

Jie Chen (Member, IEEE) received the MSc and PhD degrees from the Harbin Institute of Technology, China, in 2002 and 2007, respectively. He joined as the faculty member of the Shenzhen Graduate School, Peking University, in 2019. He is currently an associate professor with the School of Electronic and Computer Engineering, Peking University. Since 2018, he has been working with the Peng Cheng Laboratory, China. From 2007 to 2018, he worked as a senior researcher with the Center for Machine Vision and Signal Analysis, University of Oulu, Finland. In 2012 and 2015, he visited the Computer Vision Laboratory, University of Maryland, and School of Electrical and Computer Engineering, Duke University, respectively. He was a co-chair of International Workshops at ACCV, CVPR, ICCV, and ECCV. He was a guest editor of special issues for *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IJCV, and *Neurocomputing*. His research interests include deep learning, computer vision, and medical image analysis. He is an associate editor of the *Visual Computer*.

Rongrong Ji (Senior Member, IEEE) is currently a Nanqiang Distinguished professor with Xiamen University, also the deputy director with the Office of Science and Technology, Xiamen University, and also the director with Media Analytics and Computing Lab. He has authored or coauthored more than 50 papers in ACM/IEEE Transactions, including *IEEE Transactions on Pattern Analysis and Machine Intelligence* and IJCV, and more than 100 full papers on toptier conferences, such as CVPR and NeurIPS. His publications have got more than 20K citations in Google Scholar. His research interests include the field of computer vision, multimedia analysis, and machine learning. He was the recipient of the Best Paper Award of ACM Multimedia 2011, National Science Foundation for Excellent Young Scholars in 2014, National Ten Thousand Plan for Young Top Talents in 2017, and the National Science Fundation for Distinguished Young Scholars in 2020. He was a area chair in top-tier conferences such as CVPR and ACM Multimedia. He is also an Advisory Member for Artificial Intelligence Construction in the Electronic Information Education Commitee of the National Ministry of Education.