# Adaptive Search for Broad Attention based Vision Transformers

Nannan Li, *Graduate Student Member, IEEE*, Yaran Chen, *Member, IEEE*, Dongbin Zhao, *Fellow, IEEE*

***Abstract*—In recent years, Vision Transformer (ViT) has prevailed among computer vision tasks for its powerful capability of image representation. Frustratingly, the manual design of efficient architectures for ViTs can be time-consuming, often requiring repetitive trial and error. Moreover, existing light-weight ViTs have not been thoroughly explored, leading to weaker performance compared to convolutional neural networks. To address these challenges, we propose Adaptive Search for Broad attention based Vision Transformers, called ASB, which incorporates broad attention and adaptive neural architecture evolution to strengthen light-weight ViTs. The inclusion of broad attention within the search space allows us to explore novel architectures that can significantly enhance the performance of light-weight ViTs by providing more comprehensive attention information. We also design an efficient adaptive evolutionary algorithm to explore effective architectures by dynamically adjusting the probability distribution of candidate mutation operators. Our experimental results show that the adaptive evolution in ASB can efficiently learn excellent light-weight models, achieving a 55% improvement in convergence speed over traditional evolutionary algorithms. Moreover, the effectiveness of ASB is demonstrated in several visual tasks, including image classification, mobile COCO panoptic segmentation, and mobile ADE20K semantic segmentation. For instance, on ImageNet, searched model achieves 77.8% performance with 6.5M parameters, resulting in a 0.7% accuracy improvement over the state-of-the-art EfficientNet-B0. On mobile COCO panoptic segmentation, our approach outperforms prevalent MobileNetV2 by 7.4% PQ. On mobile ADE20K semantic segmentation, our method attains 40.9% mIoU, which exceeds MobileNetV2 with 6.9% mIoU.**

***Index Terms*—Vision transformer, light-weight, adaptive evolution, broad attention, image classification.**

## I. Introduction

Transformer has led the way in Natural Language Processing tasks for years [5], [6]. The latest trend of Vision Transformer (ViT) [7] also demonstrates the competence of Transformer in image representation. Specifically, ViT achieves remarkable performance on image classification (i.e., ImageNet and other downstream datasets) [7]. Correspondingly, enormous brilliant manually-designed models, such as DeiT [3], BViT [8], and CVT [9], are presented in succession. These models develop ViT with novel architecture and efficient performance.

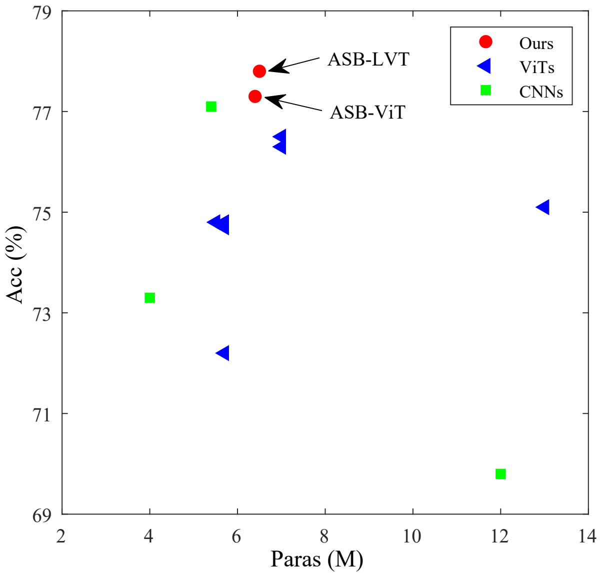


Fig. 1. Comparison of accuracy on Imagenet with respect to parameters among our models, ViTs, and CNNs. Specific models include ASB-ViT, ASB-LVT, MobileNetV2 [1], EfficientNet [2], DeiT [3], LVT [4], and so on.

Existing excellent ViT models incorporate the traits of Convolution Neural Networks (CNNs) [10]–[12], such as pyramid architecture and efficient pathway connection. Pyramid architecture divides the architecture into multiple stages with different input resolution sizes and dimensions. Pathway connection promotes information flow via a designed skip connection. As a typical pyramid ViT, Swin Transformer [13] proposes novel shifted window-based self-attention, which not only reinforces the features but also diminishes the computation complexity of self-attention. CVT [9] introduces the CNN block to reduce the sequence length to derive pyramid architecture, while PiT [14] applies the pooling operator. BViT [8] designs broad attention for comprehensive attention information via efficient pathway connection. All of these models make impressive achievements. However, the above pyramid ViTs show poor performance when with small parameters, especially compared to CNN. For example, with 5M parameters, the accuracy of EfficientNet-B0 [2] exceeds PiT [14] by 2.5%, while EfficientNet-B5 [2] surpasses Swin [13] by 2.3% with about 30M parameters. BViT delivers significant improvement on light-weight ViT. However, the enhancement brought by broad attention is limited by the capacity of the backbone network. Besides, the connection paradigm of broad attention leaves some room for further improvement. Thus it is reasonable to

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62006226 and 62173324.

N. Li, Y. Chen, and D. Zhao are with the State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China (email : linannan2017@ia.ac.cn, chenyaran2013@ia.ac.cn, dongbin.zhao@ia.ac.cn).

achieve the enhanced performance of light-weight ViTs by exploring the combination of superior ViT architecture and the broad attention paradigm.

Furthermore, although the aforementioned handcrafted ViTs have made some milestones, the procedure of architecture design is challenging and requires expert experience and plenty of trial and error. For instance, the determination of model depth, embedding dimension, number of attention heads, and connection of above broad attention. Researchers have introduced Neural Architecture Search (NAS) to automatically design ViTs in consideration of the difficulty of architecture design [15], [16]. AutoFormer [15] first presents a practical search framework for ViT. The search procedure mainly includes super-network training and evolutionary search. AutoFormer is dedicated to enhancing the training of super-network. However, the search process is time-consuming for the sake of the vast search space that contains possible candidate operations. In well-established NAS research for CNN, the efficiency is usually improved by the design of the search algorithm. Therefore, it is reasonable to explore innovative search algorithm for ViTs to alleviate the above issues that have been neglected in existing works [15], [16]. In particular, the search algorithm is expected to moderate the convergence difficulty caused by the giant search space.

This paper proposes Adaptive Search for Broad attention based Vision Transformers, called ASB. ASB prompts the competency of ViT in two ways: 1) Superior search space with broad attention. 2) Efficient search algorithm that adaptively adjusts the probability distribution of the candidate mutation operator. First, we design the search space involving broad attention with outstanding ViTs (i.e., non-pyramid architecture DeiT [3] and pyramid architecture LVT [4]) as the backbone network. The search space is exquisite for ViTs. Subsequently, we build the super-network according to the above search space and search the optimal architecture via an adaptive evolutionary algorithm on pre-trained super-network. Notably, we learn the probability distribution of candidate mutation operators in the search process. Finally, we retrain the searched non-pyramid architecture ASB-ViT and pyramid architecture ASB-LVT on ImageNet classification [17] to validate the performance of ASB. Additionally, given the effectiveness of pyramid architecture in dense prediction tasks, we apply ASB-LVT to several computer vision applications, including COCO panoptic segmentation [18] and ADE20K semantic segmentation [19]. On the above tasks, ASB-ViT and ASB-LVT deliver superior results with few parameters. The comparison of searched models and other classification models on ImageNet is shown in Fig. 1.

Our contributions are outlined below:

- We propose a broad search space that facilitates the search for light-weight architecture with excellent performance. Broad attention is introduced into the search space of ASB to enhance attention information effectively. The searched models ASB-ViT and ASB-LVT exhibit 5.0% and 3.0% rise respectively compared to vanilla backbone models (i.e., DeiT and LVT).
- We design the adaptive evolutionary algorithm to learn the optimal architecture efficiently. By adaptively learning the probability distribution of candidate mutation operators during the search process, ASB effectively accelerates the convergence of the search algorithm. In contrast to the traditional evolutionary algorithm that is adopted in AutoFormer, the search cost for pyramid ViT is shortened by 55%.
- The experiments on various computer vision tasks demonstrate that the proposed ASB is robust and powerful. Compared to the backbone model for search, ASB-LVT performs better among three tasks by a rise of +3% top-1 accuracy on ImageNet, +1.6% mIoU on ADE20K semantic segmentation, and +0.9% PQ on COCO panoptic segmentation.

## II. Related Work

### A. Vision Transformer

Transformer [6] has dominated natural language processing since it was proposed. In recent years, researchers have devoted themselves to exploring the performance of Transformer in other fields. Surprisingly, Transformer has shown remarkable performance in the field of image processing, such as ViT [7], Swin Transformer [13], LVT [4]. ViT [7] demonstrated the impressive performance of Transformer encoder on large-scale image classification (i.e., ImageNet-21K [20], JFT-300M [21]) for the first time. In order to understand the image, ViT reshapes the image into a sequence, which can be directly fed to the Transformer encoder by adding trainable position embedding and classification token.

The brilliant performance of ViT [7] brought a series of innovative ViTs that mainly contributed to the improvement of the architecture design and enhancement in performance. DeiT [3] achieved promising results without pre-trained on large-scale datasets via data-efficient training. Besides, DeiT delivered effective knowledge distillation by a distillation token. Based on DeiT, BViT [8] developed broad attention for comprehensive attention information and delivered significant performance gain. Another type of architecture is the pyramid ViT, which requires downsampling of feature size. Swin Transformer [13] downsampled the features by an elaborate reshape operation and proposed shifted window-based self-attention, which not only reinforces the features but also decreases the computation complexity. CVT [9] introduced the strided-convolution to reduce the sequence length, while PiT [14] used the pooling operator. Although these models are excellent, manual design for ViTs is time-consuming, and most of them are inferior to CNN in terms of models with small parameters.

### B. Neural Architecture Search

NAS was initially employed for the automatic design of CNN, which has been developed to a sophisticated state with many excellent algorithms [2], [22], [23]. NAS consists of three main components: search space, search strategy, and performance estimation strategy. The primary difference among the popular NAS algorithms now lies in the search strategy, including reinforcement learning algorithms [22], evolutionary algorithms [24], and gradient-based algorithms [23].

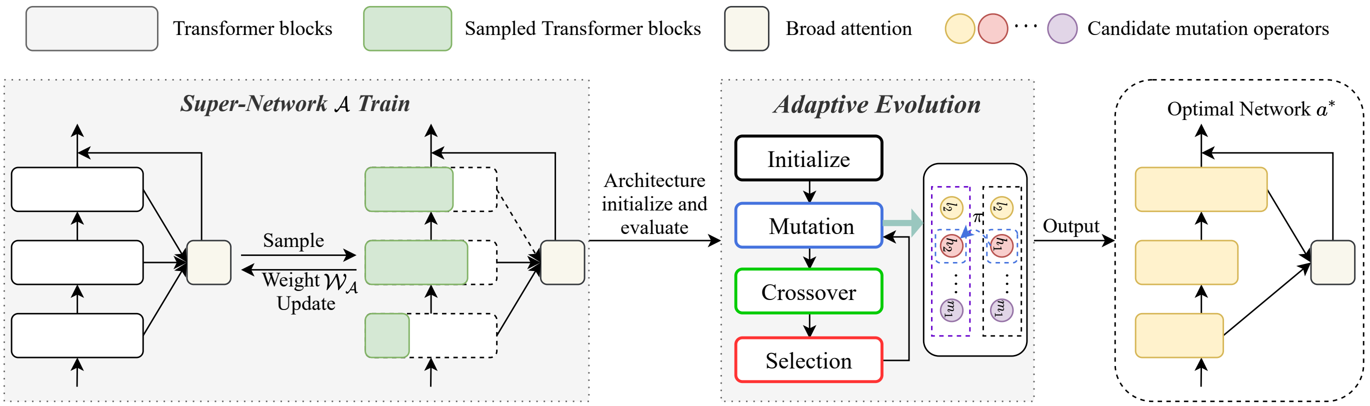


Fig. 2. The framework of ASB, which mainly consists of two phases: 1) super-network train to deliver pre-trained weights. 2) adaptive evolution with the adjustable probability distribution of the mutation operation to evolve the optimal network.

The above NAS methods accelerate the search process via weight-sharing. In particular, they construct and train a super-network that includes all candidate operations to ensure that all networks in the search space can share the weights of the super-network. While NAS has achieved great success in CNN architecture design, but there is a margin in automated ViT architecture design.

Indeed, several researchers have attempted to apply NAS to the design of ViT [15], [16], [25], [26]. Autoformer [15] first searched the high-performed ViT model through one-shot architecture search. It proposed a practical framework to train super-network involving embedding dimension, MLP ratio, depth, etc. BossNAS [25] introduced CNN block into search space to combine advantages of both CNN and ViT. It also presented the ensemble bootstrapping training technique and the unsupervised evaluation metric. ViT-Res [16] designed residual spatial reduction for pyramid ViT and proposed multi-architecture sampling for optimizing the train of the super-network. S3 [26] searched the search space of ViT and concluded the guideline for the design of pyramid ViT (i.e., Swin Transformer [13]). Unfortunately, the above works focus on the design of the search space and the training of the super-network, neglecting the development of the search algorithm.

## III. Method

The key points of ASB framework are the broad attention based search space and adaptive evolutionary algorithm. This section first gives an overview of the adaptive ViT architecture search framework. Then we introduce the details of the search space with broad attention, which is the basis for building the super-network. Finally, we elaborate on the adaptive evolution, with a focus on the learning mechanism of the probability distribution for candidate mutation operators.

### A. Overview

Similar to most one-shot NAS algorithms, our search algorithm is comprised of two main phases, as shown in Fig. 2.

- **Super-Network Train**: Based on the designed search space with broad attention, we build the super-network $\mathcal{A}$. As shown in Alg. 1, in the training process of the super-network $\mathcal{A}$, we sample sub-network $\{a|a \in \mathcal{A}\}$ uniformly to train at each iteration that improves the efficiency via partially update weights of super-network. All sub-networks share the weights $\mathcal{W}_{\mathcal{A}}$ inherited from the super-network, i.e. $\{w_a \in \mathcal{W}_{\mathcal{A}}\}$. Thus we can directly resume training the sub-network $a$ via applying the $\mathcal{W}_{\mathcal{A}}$ of the super-network instead of training from scratch.
- **Adaptive Evolution**: With pre-trained super-network, we perform adaptive evolution to learn outstanding sub-network. The goal of the evolution is maximal classification accuracy. The flows of the adaptive evolution are listed below: i) randomly sampling sub-networks from super-network $\mathcal{A}$ to *initialize* the population, ii) generating new individuals via *crossover* and *mutation* operations. iii) evaluating the performance of individuals by inheriting the weights $\mathcal{W}^*_{\mathcal{A}}$ of the pre-trained super-network $\mathcal{A}(\mathcal{W}^*_{\mathcal{A}})$ to *select* sub-networks as new generation. iiii) Repeating ii) to iii) until the termination condition is met.

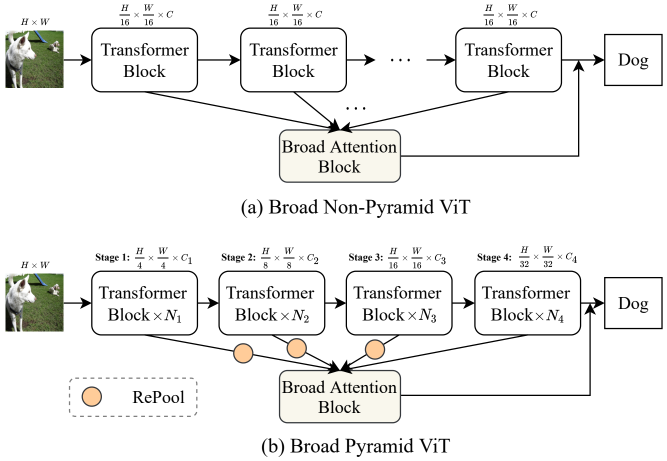
In the search algorithm framework, the super-network train mainly follows previous NAS works, as shown in Alg. 1. Our innovation lies in the design of the broad search space. Therefore we concentrate on *broad search space* and *adaptive evolution*, the former strengthens the architecture, and the latter promotes the efficiency of the search procedure.

**Algorithm 1** Super-Network $\mathcal{A}$ Train

```
Input: Datasets $\mathcal{D}$, Super-Network $\mathcal{A}$,
       epochs of super-network train $\mathcal{E}$.
  Initialize the weights $\mathcal{W}_{\mathcal{A}}$ of super-network $\mathcal{A}$
  for $n = 1, 2, ..., \mathcal{E}$ do
      Sample and train the sub-network $a$ with inherited
      weights $w_a \in \mathcal{W}_{\mathcal{A}}$
      Update the weights $\mathcal{W}_{\mathcal{A}}$ partially
  end for
Output: Pre-trained super-network $\mathcal{A}(\mathcal{W}^*_{\mathcal{A}})$
```

### B. Broad Search Space

In order to clearly illustrate the broad search space, we introduce preliminary knowledge and then present the specific


$H \times W$
$\frac{H}{16} \times \frac{W}{16} \times C$
Transformer Block
Transformer Block
Transformer Block
Dog
Broad Attention Block
Stage 1: $\frac{H}{4} \times \frac{W}{4} \times C_1$
Stage 2: $\frac{H}{8} \times \frac{W}{8} \times C_2$
Stage 3: $\frac{H}{16} \times \frac{W}{16} \times C_3$
Stage 4: $\frac{H}{32} \times \frac{W}{32} \times C_4$
Transformer Block $\times N_1$
Transformer Block $\times N_2$
Transformer Block $\times N_3$
Transformer Block $\times N_4$
RePool

(a) Broad Non-Pyramid ViT

(b) Broad Pyramid ViT

Fig. 3. The architectural details of broad non-pyramid ViT and broad pyramid ViT.

broad search space design.

*1) **Preliminaries**:* To validate the performance of the proposed adaptive evolution algorithm, we conduct searches on both pyramid ViT and non-pyramid ViT. To be specific, we adopt both two architectures as the backbone network respectively for the search space of ASB. Further, we integrate broad attention into the backbone to search for superior lightweight models. The following presents the architectural details of the broad ViTs.

We build the super-network based on backbone ViTs. In detail, given an image input, ViTs uniformly split it into non-overlapping and fixed-length patches. Then the patches are projected to sequences via a linear layer, i.e., patch embedding. The sequences are directly processed by the transformer encoder, which is the key part of feature extraction. Finally, applying a fully-connected layer yields the output of the classification category.

As shown in Fig. 3, the distinction between non-pyramid ViT and pyramid ViT lies in their architectural composition. Specifically, non-pyramid ViT consists of multiple transformer blocks with consistent resolution. In contrast, pyramid ViT comprises multiple stages of varying resolutions, each of which contains multiple transformer blocks. The transformer block is composed of self-attention layer and Multi-Layer Perceptron (MLP) layer. The architectural hyper-parameters of the transformer encoder include embedding dimension $d$, the number of transformer blocks $l$, the number of heads $h$, and MLP ratio $m$.

Furthermore, we involve broad attention to ViTs for boosting the search results and extending our previous work BViT [8]. The broad attention block in BViT consists of broad connection and parameter-free self-attention. As a pathway connection paradigm, the broad connection promotes the transmission and integration of information from different layers. Parameter-free self-attention focuses on helpful attention information hierarchically without any extra trainable parameters.

As shown in Fig. 3, in broad ViT, the broad connection enhances the path connection of attention information, which respectively concatenate query $q$, key $k$, and value $v$ at each transformer block for non-pyramid ViT or each stage for pyramid ViT. Furthermore, given the varying dimensions across different stages in the pyramid ViT, the broad pyramid ViT incorporates a modified broad attention mechanism by introducing an additional RePool operation that involves reshaping and pooling. We align $q-k-v$ dimension to obtain $q^b-k^b-v^b$ with the same dimension (i.e. the dimension $d_s$ of the last stage $s$). For example, for non-pyramid ViT, query $q^b=q$, key $k^b=k$, and value $v^b=v$. In contrast, for pyramid ViT, in $i$-th stage, $q_i^b$ can be stated as follows:

$$q_i^b = \begin{cases} \text{RePool}(q_i), & i = 1, 2, \ldots, s-1 \\ q_i, & i = s \end{cases} \tag{1}$$

where $s$ is the number of stages. $k_i^b$ and $v_i^b$ are calculated in same way as Eq (1). Then we concatenate aligned query $q^b$, key $k^b$, and value $v^b$ of $l$ transformer blocks for non-pyramid ViT or $s$ stages for pyramid ViT as below:

$$\begin{aligned} Q &= [q_1^b, q_2^b, \ldots, q_{s/l}^b] * w_b \\ K &= [k_1^b, k_2^b, \ldots, k_{s/l}^b] * w_b \\ V &= [v_1^b, v_2^b, \ldots, v_{s/l}^b] * w_b \end{aligned} \tag{2}$$

where $w_b$ is the architectural hyper-parameter of broad connection to determine which layers are connected. In BViT, $w_b = [1, 1, \ldots, 1]$. $Q$, $K$, and $V$ are concatenated queries, keys, and values accordingly. Then we perform parameter-free self-attention on $Q$, $K$, and $V$ to deliver output $Out_{broad}$ of broad attention, as below:

$$Out_{broad} = \text{BPool}(\text{Atten}_{\text{pf}}(Q, K, V), \{d_p\}), \tag{3}$$

where $\text{BPool}$ is a 1D adaptive average pooling, $\text{Atten}_{\text{pf}}$ is standard self-attention except for linear projection, $d_p$ is the output dimension of backbone network.

*2) **Broad Search Space Design**:* Based on the aforementioned broad pyramid ViT, we design the search space that consists of five architectural hyper-parameters: embedding dimension $d$ and number of transformer blocks $l$, number of heads $h$, MLP ratio $m$, and broad connection $w_b$. The presented search space is detailed in Table I. As a result of the inconsistent resolution in pyramid ViT, the candidate operations for each stage are different. It is noteworthy that the selection of these hyper-parameters exerts a substantial influence on the performance of ViTs.

TABLE I
THE SEARCH SPACE OF ASB, INCLUDING NON-PYRAMID VIT AND PYRAMID VIT.

| **Stage**<br>$i$ | **Embed**<br>**Dim** $d$ | **Trans**<br>**Num** $l$ | **Head**<br>**Num** $h$ | **MLP**<br>**Ratio** $m$ | **Broad**<br>$w_b$ |
|---|---|---|---|---|---|
| | | Non-Pyramid ViT | | | |
| - | (192,216,240) | (12,13,14) | (3,4) | (3.5,4) | (0,1) |
| | | Pyramid ViT | | | |
| 1 | (56,64,72) | (1,2,3) | (2,3) | (3.5,4) | (0,1) |
| 2 | (56,64,72) | (1,2,3) | (2,3) | (6,8) | (0,1) |
| 3 | (148,160,172) | (2,3,4) | (3,5) | (4,5) | (0,1) |
| 4 | (240,256,272) | (2,3,4) | (5,8) | (4,6) | (0,1) |

Remarkably, we extend and improve our previous work BViT [8] about the hyper-parameter $w_b$, which is set to

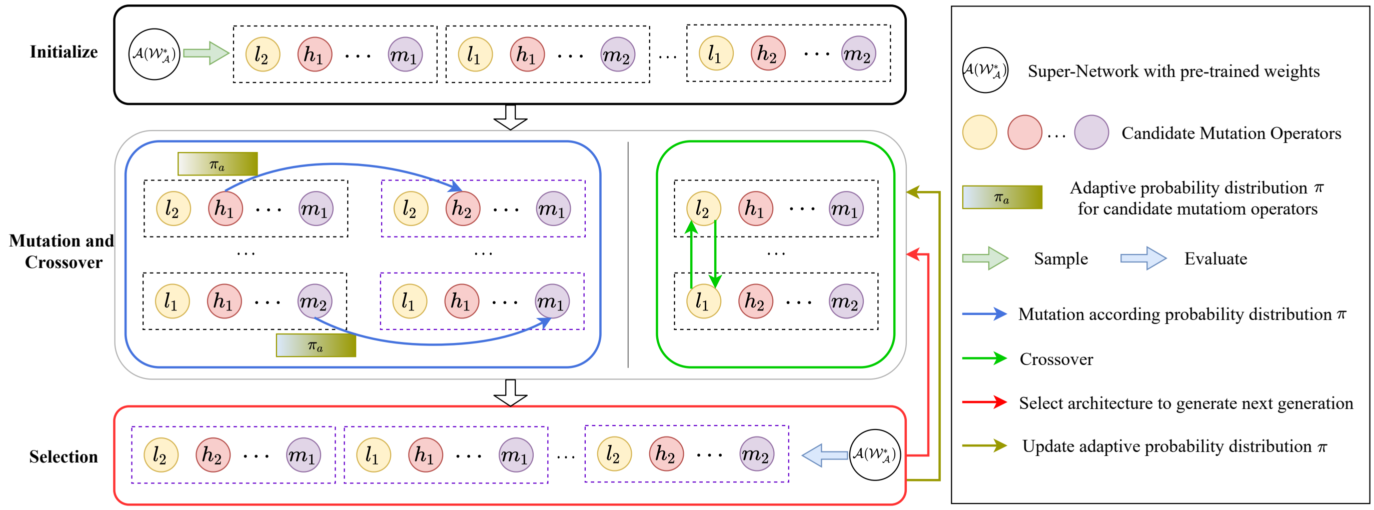

Initialize
Mutation and Crossover
Selection
$\mathcal{A}(\mathcal{W}_\mathcal{A}^*)$
$l_2$ $h_1$ ··· $m_1$
$l_1$ $h_1$ ··· $m_2$
$l_1$ $h_2$ ··· $m_2$
$\pi_a$
Super-Network with pre-trained weights
Candidate Mutation Operators
Adaptive probability distribution $\pi$ for candidate mutatiom operators
Sample
Evaluate
Mutation according probability distribution $\pi$
Crossover
Select architecture to generate next generation
Update adaptive probability distribution $\pi$

Fig. 4. The flows of adaptive evolution, which includes 1) initialization of the population, 2) mutation on candidate operations with a corresponding probability distribution and crossover between two sub-network individuals to generate new populations, 3) evaluation and selection of the top $k$ sub-network individuals for deriving next generation.

$[1, 1, \ldots, 1]$. However, such a hyper-parameter may be redundant and cannot deliver optimal performance gains. To seek the optimal broad connection, we include $w_b$ into the search space as shown in Table I. To be specific, 0 denotes no broad connection and 1 denotes with the broad connection. The experimental results in Section IV shows that the introduction of broad attention indeed improves the quality of the search space.

### C. Adaptive Evolution

As shown in Fig. 4, our architecture evolution process develops the mutation operations of the traditional evolutionary algorithm flow. Due to the excessive search space, the evolutionary algorithm is difficult to converge and tends to fall into the local optimum. Therefore, we learn the direction of mutation adaptively during the evolution process, which makes the mutation operation more inclined to choose the excellent candidate operation instead of blindly exploring. To be specific, ASB proposes an adaptive learning mechanism for the probability distribution of candidate mutation operators. The adaptive learning mechanism provides guidance for automated model design and makes the search more efficient.

The ASB approach places a significant emphasis on the learning of the probability distribution, wherein the accuracy rank during the search process plays a crucial role. We design two types of the probability distribution for mutation, including uniform probability distribution $\pi_u$ to promote exploration of effective architecture and rank probability distribution $\pi_r$ to guide the evolutionary direction of the mutation operators. Then we employ them in each generation based on the probability $\mathrm{f}(g)$ associated with the number of generations $g$ for performing adaptive evolution.

**Design of Probability Distribution**: During the early period of the evolutionary process, the uniform probability distribution $\pi_u$ is implemented to foster the exploration of the network architecture, ultimately leading to the credible rank of candidate mutation operators. The uniform probability distribution $\pi_u$ is expressed as follows

$$\pi_u = [\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n}], \tag{4}$$

where $n$ denotes to number of candidate mutation operators. For example, the uniform probability distribution $\pi_u = [1/2, 1/2]$ for MLP ratio that has two candidate mutation operators.

In the search process, an adaptive learning-based rank probability distribution $\pi_r$ is employed to guide the mutation direction. The computation of the rank probability distribution entails a frequency analysis of the occurrence of candidate mutation operators in the top-$k$ models, culminating in an estimation of the probability distribution of the mutation operators. The equation of the rank probability distribution $\pi_r$ is as follows

$$\pi_r = [\frac{r_1}{k}, \frac{r_2}{k}, \ldots, \frac{r_n}{k}], \tag{5}$$

where $r_j$ is the numbers of $j$-th candidate mutation operator appears in the top-$k$ models. For example, with $k = 5$, if the first candidate mutation operator appears two times and the second candidate mutation operator appears three times for MLP ratio which has two candidate mutation operators, the rank probability distribution $\pi_r = [2/5, 3/5]$.

**Implementation of Adaptive Probability Distribution**: Considering that the rank probability distribution $\pi_r$ is more reliable with iterations of the search process, we adopt the probability distribution based on the probability $\mathrm{f}(g)$ associated with the number of iterations $g$. The

$$\begin{aligned} \pi_a &= \mathrm{f}(g)\pi_u + (1 - \mathrm{f}(g))\pi_r, \\ \mathrm{f}(g) &= \begin{cases} \frac{\mathcal{G}-g}{\mathcal{G}}, & if\ g > g_w \\ 1, & else \end{cases} \end{aligned} \tag{6}$$

where $\mathcal{G}$ is the generations of evolution, $g$ is current generation for search, $g_w$ is warmup generation for derive dependable adaptive probability distribution $\pi_a$.

**Algorithm 2** Adaptive Evolution

```
Input: Datasets $\mathcal{D}$, Pre-trained Super-Network $\mathcal{A}(\mathcal{W}_{\mathcal{A}}^{*})$,
        generations of evolutionary search $\mathcal{G}$,
        uniform probability distribution $\pi_u$.
  Initialize the population
  for $g = 1, 2, ..., \mathcal{G}$ do
      Do mutation according to adaptive probability
      distribution $\pi_a$ in Eq. (6)
      Do crossover
      Evaluate the performance $acc$ of population
      Select top-$k$ sub-networks according to $acc$
      Calculate rank probability distribution $\pi_r$ according
      to Eq. (5)
      Determine the population of the next generation
  end for
Output: Optimal sub-network $a^*$
```

As shown in Alg. 2, in the adaptive evolution process, the adaptive probability distribution $\pi_a$ is performed according to Eq (6) that employs uniform probability distribution $\pi_u$ and rank probability distribution $\pi_r$ via corresponding probability $\mathrm{f}(g)$ to prevent the search from being caught in the local optimum due to a single choice of the rank probability distribution $\pi_r$. Specifically, in the first $g_w$ generations, ASB employs uniform probability distribution $\pi_u$ to learn a relatively sound rank probability distribution $\pi_r$. The rank probability distribution $\pi_r$ is drawn by statistically counting the distribution of candidate operators in the top-$k$ architectures for each generation. It contributes to accelerating the convergence of the evolutionary algorithm and thereby improving search efficiency. As the iterative process advances, the probability of utilizing rank probability distribution $\pi_r$ progressively increases, eventually fulfilling the objective of directing the mutation orientation. In general, the adaptive probability distribution $\pi_a$ for candidate mutation operators facilitates the exploration and exploitation of novel architectures.

## IV. Experiments

In this section, we conduct the following experiments to illustrate the effectiveness of ASB. First, to verify the quality of broad search space and the efficiency of adaptive evolution, we conduct the ablation study, including whether to introduce broad attention and whether to employ adaptive probability distribution. Next, we present the probability variation of candidate mutation operators and discuss the rules for light-weight architecture design. Then, we compare searched models ASB-ViT and ASB-LVT with other excellent classification models on ImageNet [17] to demonstrate the superiority of ASB. Finally, we validate the generalization of pyramid architecture ASB-LVT on ADE20K [19] semantic segmentation and COCO [18] panoptic segmentation. The experiments are implemented on Nvidia Tesla V100 GPUs.

### A. Ablation Study

The two critical elements of ASB are broad search space and adaptive evolution. Therefore, we prove the effectiveness of ASB from these two aspects. For the broad search space, we discuss the impact of broad attention on the performance of the searched model. For adaptive evolution, we analyze the effect of adaptive probability distribution on the speed of convergence of the search algorithm.

The search experimental setup is consistent for all experiments in the ablation study. The population size is set to 50. The number of evolutionary generations is set to 20. The number of mutations is set to 25 and the mutation probability is set to 0.5. The number of crossovers is set to 25. In addition, to obtain light-weight model, we limit the parameters of the models in the search process to a lower limit of 3.0M and an upper limit of 6.5M. In particular, for ASB, the warmup generation is set to 3, and the rank probability distribution is calculated in the top-5 models at each generation.

*1) **Broad Search Space:*** In order to discuss the effect of broad attention on the searched model, we carry out two experiments, 1) search in the search space without broad attention that derives AS-ViT and AS-LVT, and 2) search in the search space with broad attention that yields ASB-ViT and ASB-LVT. Then we train the searched models from scratch and compare their performance on ImageNet, as shown in Table II.

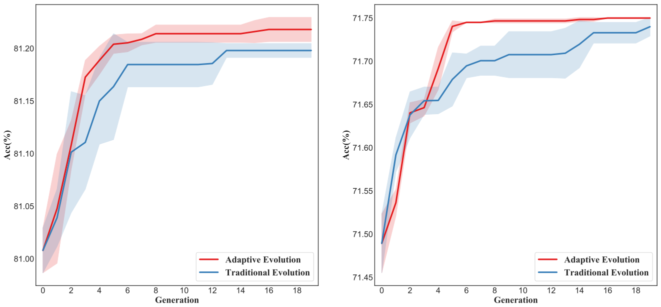
TABLE II
Ablation on broad attention of search space. Search space with broad attention has significant advantage compared to search space without broad attention.

| **Method** | **Broad Attention** | **Top1 Acc** |
|---|---|---|
| | Non-pyramid ViT | |
| DeiT [3] | × | 72.2 |
| AS-ViT | × | 76.5 |
| **ASB-ViT(ours)** | ✓ | **77.3** |
| | Pyramid ViT | |
| LVT [4] | × | 74.8 |
| AS-LVT | × | 77.4 |
| **ASB-LVT(ours)** | ✓ | **77.8** |

The experimental results show that the performance of models that searched in the broad search space is superior with a rise of 0.8% in non-pyramid architecture ASB-ViT and 0.4% in pyramid architecture ASB-LVT, which suggests that broad attention improves the quality of the search space. In brief, the broad search space is more capable of finding high-performance models with fewer parameters.

*2) **Adaptive Evolution:*** To analyze the impact of adaptive probability distribution on the search algorithm, we perform two types of search for non-pyramid ViT and pyramid ViT respectively, i.e., adaptive evolution and traditional evolution. Then we compare the search curves of the two types of evolution, as shown in Fig 5. Notably, to demonstrate the robustness of adaptive evolution, we randomly select three seeds (i.e., seed 0, seed 901, and seed 3407) to search three times for each search type. All comparisons are averaged over the three search results.

As seen from Fig 5, compared to traditional evolution, adaptive evolution exhibits a significant improvement of optimal search accuracy on non-pyramid ViT and faster convergence on pyramid ViT. According to the experimental results, it can



(a) Non-pyramid ViT (b) Pyramid ViT

Fig. 5. Search curves of the two types of search algorithms, including adaptive evolution and traditional evolution.

be observed that the traditional evolutionary algorithm is constrained by a local optimum in the case of non-pyramid ViT, resulting in faster convergence. On the other hand, for pyramid ViT, the proposed ASB demonstrates a 55% enhancement in search efficiency with superior search results. Specifically, on a single V100, ASB for pyramid ViT can save about 16 hours of search cost. Evidently, the design of adaptive evolution can alleviate the difficulty in the convergence of the search in the case of a giant search space. It is noteworthy that the higher search accuracy of non-pyramid ViT can be attributed to its search space having smaller complexity. Consequently, the non-pyramid super-network encompasses fewer candidate operations and is trained more fully.

## B. Analysis of Probability Distribution

Given that the majority of current ViTs adopt the pyramid architecture, we conduct an observation and analysis of the probability distribution of candidate mutation operators during the search process of pyramid ASB-LVT. As shown in Fig 6, we present the variation pattern of the rank probability distribution for candidate mutation operators in the search space, which consist of MLP ratio $m$, number of heads $h$, number of transformer blocks $l$, and embedding dimension $s$. Since we limit the model size, the following analysis is for light-weight models. Without the restriction on the number of parameters, the probability distribution may evolve quite differently. The concrete observations and analysis are as follows:

- *MLP Ratio*: The selection of MLP ratio has a significant tendency in the last three stages. Specifically, the second stage prefers to select a large MLP ratio, while the latter two stages prefer a small MLP ratio. Thus when computational resources are constrained, increasing the MLP ratio in the second stage instead of the last stage should be considered.
- *Number of Heads*: The latter two stages demonstrate a stark difference in tendency to head number. The third stage tends to have fewer heads, while the last stage requires more heads. It can be seen that multi-view attention to the features of the last stage is more conducive to performance improvement.
- *Number of Transformer blocks*: The first three stages all favor two transformer blocks, and the last stage favors three transformer blocks. For the poor performance of the pyramid ViT, increasing the number of transformer blocks in the last stage should be taken into account. Notably, the observed phenomenon is consistent with the analysis conducted by the research [27].
- *Embedding Dimension*: Basically, the embedding dimension increases with the stage. For light-weight architectures, priority is placed on adding the dimension of the first two stages.

We sincerely expect that the above observations and analysis will inspire the manual design of light-weight pyramid architectures. Indeed, given the differences among backbone architectures, it is most effective to employ ASB to automate the design of architectures specifically.

## C. ImageNet Classification

*1) **Dataset**:* We conduct image classification experiments on ImageNet [17]. Imagenet is a popular benchmark dataset in computer vision, which includes training sets with 1.3M images and validation sets with 50K images, respectively. In total, the dataset has various 1000 object classes.

*2) **Settings**:* The training settings follow the settings commonly used by previous methods [3]. For ASB-ViT and ASB-LVT, the input image resolution is set to 224 × 224. And we train the model for 300 epochs, employing Adamw [28] optimizer and using cosine decay learning rate scheduler. The learning rate varies based on the batch size, following the formula $lr = \frac{batch\_size}{1024} \times lr\_base$. $lr\_base$ is set to $1 \times 10^{-3}$ for ASB-ViT and $1.6 \times 10^{-3}$ for ASB-LVT respectively. The weight decay is set to 0.05. We employ stochastic depth with drop path rate of 0.1 [29]. Moreover, similar to vanilla backbone models DeiT [3] and LVT [4], ASB-ViT and ASB-LVT employ a majority of the augmentation and regularization strategies in training.

*3) **Architecture**:* The searched non-pyramid architecture ASB-ViT is shown in Table III. From the observation, it can be inferred that the deep transformer block exhibits a tendency toward selecting a larger MLP ratio. The empirical analysis indicates a tendency for the number of heads to select relatively large values. Furthermore, the broad connection exhibits a more uniform distribution, potentially due to the redundancy of neighboring transformer blocks with similar features.

TABLE III
THE SEARCHED NON-PYRAMID ARCHITECTURE ASB-VIT.

| | |
|---|---|
| Embed Dim | 192 |
| Trans Num | 13 |
| Head Num | (3,4,3,3,3,3,3,3,4,4,4,4,3) |
| MLP Ratio | (3.5,4,4,4,3.5,4,4,4,4,4,4,4,4) |
| Broad | (0,1,0,1,0,0,1,1,0,0,1,0,0) |

The searched pyramid architecture ASB-LVT is shown in Table IV. The rest of the architectural details are consistent with backbone pyramid architecture LVT [4]. The architecture analysis is exhibited in IV-B. Given the performance advantage in dense prediction exhibited by the pyramid architecture,

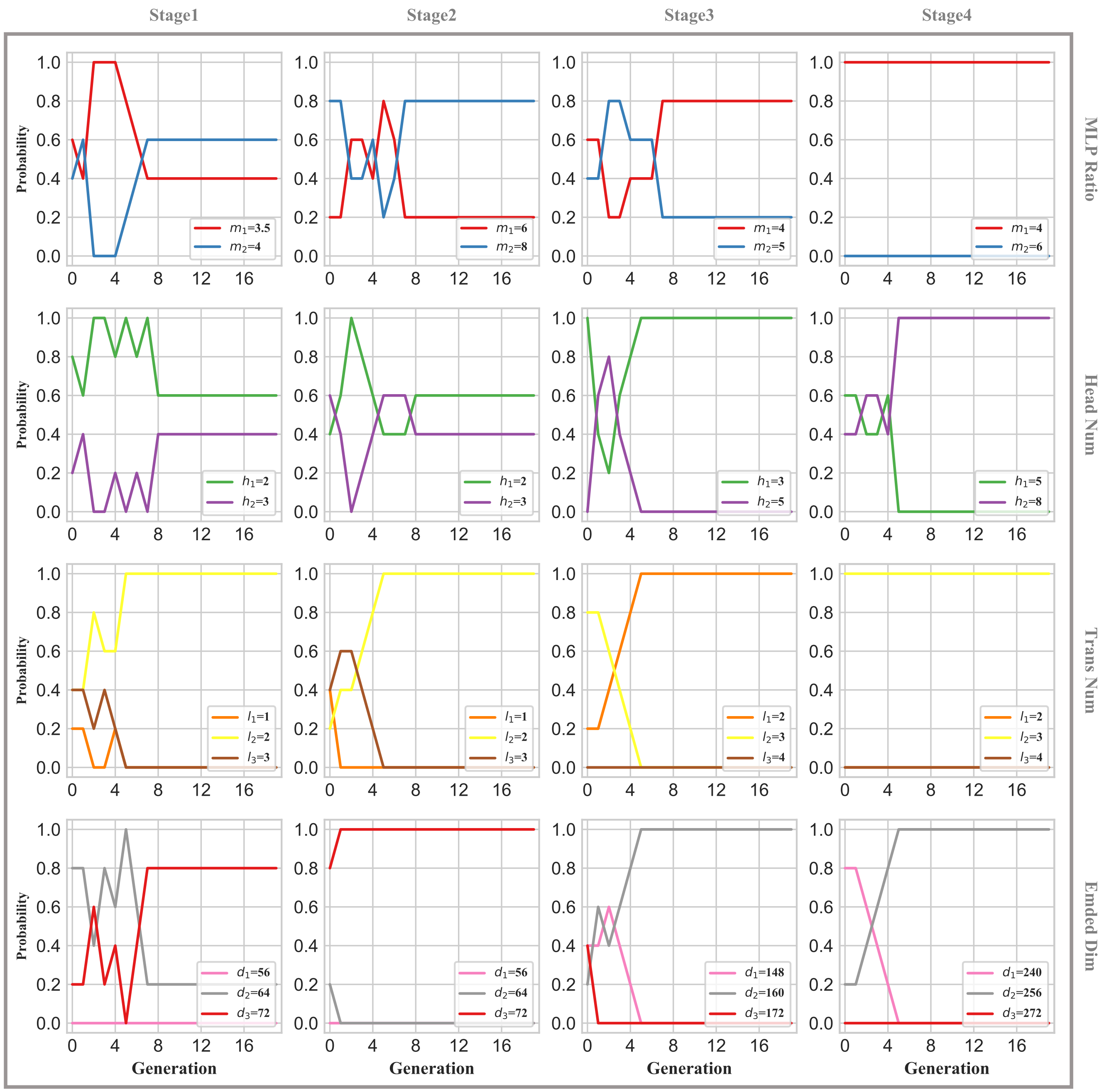


Fig. 6. Visualization of the probability distribution evolution. Rows represent operations in the search space, including MLP ratio $m$, number of heads $h$, number of transformer blocks $l$, and embedding dimension $d$. Columns indicate the different stages of the pyramid architecture. The horizontal axis of the subplot indicates the evolutionary generation and the vertical axis indicates the probability value of the candidate mutation operators. The lines of different colors denote different candidate mutation operators.

we translated the searched pyramid architecture ASB-LVT to other visual tasks. The architecture used in the ADE20K semantic segmentation and COCO panoptic segmentation is consistent and not described subsequently.

*4) **Results**:* Table V presents the performance comparison to excellent vision models with few parameters on ImageNet [17], including CNNs, MLPs, and ViTs. Among lightweight models, pyramid ASB-LVT searched by ASB achieves state-of-the-art performance, outperforming MobileNetV3 [1], LVT [4] with approximately 3% improvement, and even EfficientNet [13], which leads in vision tasks. We also exceed gMLP [30] by about 5%, which delivers top results in MLP models. Despite its slight inferiority to the pyramid architecture ASB-LVT, ASB-ViT exceeds all other existing models.

In summary, the impressive performance of ASB-ViT and ASB-LVT serves as evidence of the effectiveness of ASB. Through the use of a high-quality broad search space and efficient adaptive search algorithms, ASB is capable of producing high-performance architecture achieved within 10M parameters. The experimental results demonstrate the great potential

TABLE IV
THE SEARCHED PYRAMID ARCHITECTURE ASB-LVT.

| Stage | Embed Dim | Trans Num | Head Num | MLP Ratio | Is Broad |
|---|---|---|---|---|---|
| 1 | 72 | 2 | 3 | 3.5 | 1 |
| 2 | 72 | 2 | 2 | 8 | 0 |
| 3 | 160 | 2 | 3 | 4 | 0 |
| 4 | 256 | 3 | 8 | 4 | 1 |

of ASB as a powerful tool for exploring and optimizing neural network architectures in computer vision tasks.

### *D. Generalization Study*

We affirm the generalization of ASB-LVT on ADE20K semantic segmentation [19] and COCO panoptic segmentation [18].

*1) **ADE20K Semantic Segmentation**:* We apply ASB-LVT to semantic segmentation task, i.e., challenging ADE20K dataset [19]. The dataset has 150 categories that consist of 35 stuff classes and 115 discrete objects. ADE20K dataset includes training sets with 20,210 images and validation sets with 2,000 images, respectively.

**Settings:** The training settings follow the settings commonly used by the previous method. We adopt the Segformer framework [34] with an MLP decoder and ASB-LVT encoder, where the ASB-LVT encoder is pre-trained on ImageNet, while the MLP decoder is trained from scratch. Our implementation is based on the mmsegmentation [35] codebase. We train the model for 160K training iterations using a batch size of 16 with the Adamw [28] optimizer and a poly learning rate schedule with a power of 1. The initial learning rate is set to $6 \times 10^{-5}$, while the weight decay is set to 0.01. To augment the data, we randomly resize the image with a ratio of $0.5 - 2.0$ and perform random cropping of size $512 \times 512$. We also apply horizontal flipping with a probability of 0.5. For evaluation, we perform single-scale testing.

**Results**: With Segformer [34] as the framework, we show the performance of ASB-LVT on ADE20K semantic segmentation in Table VI. ASB aims to search light-weight models, thus comparison models are mobile methods for semantic segmentation. ASB-LVT yields the best result, surpassing LVT [4] with 1.7% mIoU, which is the backbone ViT used in our search procedure.

The high adaptability of the searched ASB-LVT model to other tasks is evident, thereby demonstrating its transferability, which can be attributed to its robust architecture and effective search algorithm.

*2) **COCO Panoptic Segmentation**:* We carry out the panoptic segmentation on COCO dataset [18]. We choose COCO 2017 split which includes training sets with 118K images and validation sets with 5K images. Specifically, each image contains 3.5 categories and 7.7 instances. To make a better evaluation of our approach, we choose the panoptic segmentation task, which integrates object recognition, detection, localization, and segmentation.

**Settings:** The training settings follow the settings commonly used by LVT [4]. We adopt the panoptic FPN framework [36] and compare it with models based on this framework. The mmdetection codebase [37] is employed for implementation. We train the model for 36 epochs using the Adamw optimizer [28] and 3× schedule with learning rate decays by 10 times after 24 and 33 epochs. The initial learning rate is set to $3 \times 10^{-4}$, and the weight decay is set to $1 \times 10^{-4}$. We employ multi-scale training, where the images are randomly resized with the maximum length limit of 1333, and the maximum allowable length of the short side is randomly sampled within the range of $640 - 800$. We also apply horizontal flipping with a probability of 0.5 for data augmentation. In testing, single-scale test is performed for evaluation.

**Results:** The comparison between ASB-LVT and other mobile models on COCO panoptic segmentation is provided in Table VII. With the top result, ASB-LVT brings 0.9%, 0.5% and 1.7% improvement respectively in Panoptic Quality all (PQ), Panoptic Quality things ($\text{PQ}^{\text{th}}$) and Panoptic Quality stuff ($\text{PQ}^{\text{st}}$), compared to LVT [4]. In contrast to traditional CNN model [1], ASB-LVT shows even greater superiority with 3.4%, 7.1%, and 7.9% respectively in three PQ metrics.

As a benchmark task that integrates various computer vision tasks, including object recognition, detection, localization,

TABLE V
PERFORMANCE COMPARISON WITH STATE-OF-THE-ART MODELS ON IMAGENET, INCLUDING VITS, CNNS, AND MLPS. WITH ABOUT 6M PARAMETERS, THE SEARCHED MODELS ASB-VIT AND ASB-LVT OUTPERFORM ALL THE METHODS.

| **Network** | **Resolution** | **Params** | **FLOPs** | **Top1 Acc** | **Method Type** | **Design Type** |
|---|---|---|---|---|---|---|
| MobileNetV3$_{Large0.75}$ [1] | $224^2$ | 4.0M | 0.16G | 73.3 | CNN | Auto |
| ResNet-18 [10] | $224^2$ | 12.0M | 1.8G | 69.8 | CNN | Manual |
| EfficientNet-B0 [2] | $224^2$ | 5.4M | 0.39G | 77.1 | CNN | Auto |
| gMLP-Ti [30] | $224^2$ | 6.0M | 1.4G | 72.3 | MLP | Manual |
| DeiT-Ti [3] | $224^2$ | 5.7M | 1.2G | 72.2 | Transformer | Manual |
| T2T-ViT-12 [31] | $224^2$ | 7M | 2.2G | 76.5 | Transformer | Manual |
| PVT-Tiny [32] | $224^2$ | 13M | 1.9G | 75.1 | Transformer | Manual |
| ViL-Tiny [33] | $224^2$ | 7M | 1.3G | 76.3 | Transformer | Manual |
| LVT [4] | $224^2$ | 5.5M | 0.9G | 74.8 | Transformer | Manual |
| AutoFormer-Ti [15] | $224^2$ | 5.7M | 1.3G | 74.7 | Transformer | Auto |
| BViT-5M [8] | $224^2$ | 5.7M | 1.2G | 74.8 | Transformer | Manual |
| **ASB-ViT (Ours)** | $224^2$ | 6.4M | 1.4G | 77.3 | Transformer | Auto |
| **ASB-LVT (Ours)** | $224^2$ | 6.5M | 1.0G | **77.8** | Transformer | Auto |

TABLE VI

PERFORMANCE COMPARISON WITH STATE-OF-THE-ART MODELS ON ADE20K SEMANTIC SEGMENTATION. WE REPORT THE RESULTS FOR THE SINGLE-SCALE INPUT. ASB-LVT ACHIEVES THE BEST PERFORMANCE AMONG THE EXCELLENT LIGHT-WEIGHT SEMANTIC SEGMENTATION MODELS.

| **Method** | **Encoder** | **mIoU** | **Params(M)** | **FLOPS(G)** |
|---|---|---|---|---|
| FCN [38] | MobileNetV2 [39] | 19.7 | 9.8 | 39.6 |
| PSPNet [40] | MobileNetV2 [39] | 29.6 | 13.7 | 52.9 |
| DeepLabV3+ [41] | MobileNetV2 [39] | 34.0 | 15.4 | 69.4 |
| SegFormer [34] | MiT-B0 [34] | 37.4 | 3.8 | 8.4 |
| SegFormer [34] | LVT [4] | 39.3 | 3.9 | 10.6 |
| SegFormer [34] | **ASB-LVT(ours)** | **40.9** | 4.4 | 10.5 |

TABLE VII

PERFORMANCE COMPARISON WITH STATE-OF-THE-ART MODELS ON COCO PANOPTIC SEGMENTATION. ASB-LVT ACHIEVES THE BEST PERFORMANCE AMONG THE EXCELLENT LIGHT-WEIGHT MODELS FOR PANOPTIC SEGMENTATION.

| **Method** | **Backbone** | **COCO$_{\mathrm{val}}$** | | | **Params** | **FLOPS** |
|---|---|---|---|---|---|---|
| | | PQ | PQ$^{\mathrm{th}}$ | PQ$^{\mathrm{st}}$ | (M) | (G) |
| Panoptic FPN [36] | MobileNetV2 [39] | 36.3 | 42.9 | 26.4 | 4.1 | 32.9 |
| Panoptic FPN [36] | PVTv2-B0 [42] | 41.3 | 47.5 | 31.9 | 5.3 | 49.7 |
| Panoptic FPN [36] | LVT [4] | 42.8 | 49.5 | 32.6 | 5.4 | 56.4 |
| Panoptic FPN [36] | **ASB-LVT(ours)** | **43.7** | **50.0** | **34.3** | 6.0 | 56.4 |

and segmentation, COCO panoptic segmentation serves as a reliable indicator of the generalization ability of ASB-LVT. The demonstrated success of ASB-LVT in this task highlights its potential for convenient application to other tasks with performance improvements. This further underscores the versatility and robustness of the ASB-LVT architecture.

## V. CONCLUSION

This paper proposes the Adaptive Search for Broad attention based Vision Transformers, called ASB. There are two critical elements of ASB, i.e., broad search space and adaptive evolution. The former explores high-performance models by introducing broad attention to improve the quality of the search space. The latter adaptively learn probability distribution to guide search and improve search efficiency. Consequently, the searched models achieve leading performance on vision tasks benefiting from the superior search space and search strategy of ASB. Specifically, on ImageNet, searched model ASB-LVT arrives at state-of-the-art performance among models with about 10M parameters. Then we transfer pyramid ASB-LVT to ADE20K semantic segmentation and COCO panoptic segmentation that affirms the robust transferability of the model. Moreover, the ablation study confirms the effectiveness of broad attention and adaptive probability distribution, which can derive outstanding architecture with less cost.

ASB shines in improving search efficiency. However, ASB is for searching specific backbone architecture and does not cover a variety of novel architectures. In future research, we expect to build a search space containing various architectures and explore more efficient structures.

## REFERENCES

[1] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan *et al.*, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1314–1324.

[2] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 6105–6114.

[3] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *International Conference on Machine Learning*. PMLR, 2021, pp. 10 347–10 357.

[4] C. Yang, Y. Wang, J. Zhang, H. Zhang, Z. Wei, Z. Lin, and A. Yuille, "Lite vision transformer with enhanced self-attention," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11 998–12 008.

[5] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *3rd International Conference on Learning Representations, ICLR 2015,*, 2015.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[7] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021.

[8] N. Li, Y. Chen, W. Li, Z. Ding, and D. Zhao, "Bvit: Broad attention based vision transformer," *arXiv preprint arXiv:2202.06268*, 2022.

[9] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, "Cvt: Introducing convolutions to vision transformers," *arXiv preprint arXiv:2103.15808*, 2021.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[11] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[12] Z. Ding, Y. Chen, N. Li, D. Zhao, Z. Sun, and C. P. Chen, "Bnas: Efficient neural architecture search using broad scalable architecture," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[13] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," *arXiv preprint arXiv:2103.14030*, 2021.

[14] B. Heo, S. Yun, D. Han, S. Chun, J. Choe, and S. J. Oh, "Rethinking spatial dimensions of vision transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 11 936–11 945.

[15] M. Chen, H. Peng, J. Fu, and H. Ling, "Autoformer: Searching transformers for visual recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12 270–12 280.

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

[17] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[18] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[19] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ade20k dataset," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 633–641.

[20] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[21] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, "Revisiting unreasonable effectiveness of data in deep learning era," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 843–852.

[22] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," *arXiv preprint arXiv:1611.01578*, 2016.

[23] H. Liu, K. Simonyan, and Y. Yang, "Darts: Differentiable architecture search," *arXiv preprint arXiv:1806.09055*, 2018.

[24] E. Real, S. Moore, A. Selle, S. Saxena, Y. L. Suematsu, J. Tan, Q. V. Le, and A. Kurakin, "Large-scale evolution of image classifiers," in *International Conference on Machine Learning*. PMLR, 2017, pp. 2902–2911.

[25] C. Li, T. Tang, G. Wang, J. Peng, B. Wang, X. Liang, and X. Chang, "Bossnas: Exploring hybrid cnn-transformers with block-wisely self-supervised neural architecture search," *arXiv preprint arXiv:2103.12424*, 2021.

[26] M. Chen, K. Wu, B. Ni, H. Peng, B. Liu, J. Fu, H. Chao, and H. Ling, "Searching the search space of vision transformer," *Advances in Neural Information Processing Systems*, vol. 34, pp. 8714–8726, 2021.

[27] N. Park and S. Kim, "How do vision transformers work?" *arXiv preprint arXiv:2202.06709*, 2022.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[29] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, "Deep networks with stochastic depth," in *European conference on computer vision*. Springer, 2016, pp. 646–661.

[30] H. Liu, Z. Dai, D. R. So, and Q. V. Le, "Pay attention to mlps," *arXiv preprint arXiv:2105.08050*, 2021.

[31] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, Z. Jiang, F. E. Tay, J. Feng, and S. Yan, "Tokens-to-token vit: Training vision transformers from scratch on imagenet," *arXiv preprint arXiv:2101.11986*, 2021.

[32] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pyramid vision transformer: A versatile backbone for dense prediction without convolutions," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 568–578.

[33] P. Zhang, X. Dai, J. Yang, B. Xiao, L. Yuan, L. Zhang, and J. Gao, "Multi-scale vision longformer: A new vision transformer for high-resolution image encoding," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 2998–3008.

[34] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," *Advances in Neural Information Processing Systems*, vol. 34, pp. 12 077–12 090, 2021.

[35] M. Contributors, "Mmsegmentation: Openmmlab semantic segmentation toolbox and benchmark," *Availabe online: https://github. com/open-mmlab/mmsegmentation (accessed on 18 May 2022)*, 2020.

[36] A. Kirillov, R. Girshick, K. He, and P. Dollár, "Panoptic feature pyramid networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 6399–6408.

[37] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu *et al.*, "Mmdetection: Open mmlab detection toolbox and benchmark," *arXiv preprint arXiv:1906.07155*, 2019.

[38] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[39] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[40] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2881–2890.

[41] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 801–818.

[42] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, "Pvt v2: Improved baselines with pyramid vision transformer," *Computational Visual Media*, vol. 8, no. 3, pp. 415–424, 2022.