Hybrid Architecture-Based Evolutionary Robust Neural Architecture Search

Shangshang Yang, *Member, IEEE*, Xiangkun Sun, Ke Xu, Yuanchao Liu, Ye Tian, *Member, IEEE*, and Xingyi Zhang, *Senior Member, IEEE*

***Abstract*—The robustness of neural networks in image classification is important to resist adversarial attacks. Although many researchers proposed to enhance the network robustness by inventing network training paradigms or designing network architectures, existing approaches are mainly based on a single type of networks, e.g., convolution neural networks (CNNs) or vision Transformer (ViT). Considering a recently revealed fact that CNNs and ViT can effectively defend against adversarial attacks transferred from each other, this paper aims to enhance network robustness by designing robust hybrid architecture networks containing different types of networks. To this end, we propose a hybrid architecture-based evolutionary neural architecture search approach for robust architecture design, termed HA-ENAS. Specifically, to combine or aggregate different types of networks in the same network framework, a multi-stage block-wise hybrid architecture network is first devised as the supernet, where three types of blocks (called convolution blocks, Transformer blocks, multi-layer perception blocks) are further designed as each block's candidate, and thus a hybrid architecture-based search space is established for HA-ENAS; then, the robust hybrid architecture search is formulated as an optimization problem maximizing both clean and adversarial accuracy of architectures, and an efficient multi-objective evolutionary algorithm is employed to solve the problem, where a supernet-based retraining evaluation and a surrogate model are used to mitigate coupled weight influence and reduce the whole search cost. Experimental results show that the hybrid architectures found by the proposed HA-ENAS outperform state-of-the-art single-type architectures in terms of clean accuracy and adversarial accuracy under a variety of common attacks.**

***Index Terms*—Networks robustness, hybrid architectures, evolutionary neural architecture search, surrogate-assisted.**

Manuscript received 18 January 2024; accepted 7 May 2024. Date of publication 29 May 2024; date of current version 24 July 2024. This work was supported in part by the National Key R&D Program of China under Grant 2018AAA0100100, in part by the National Natural Science Foundation of China under Grant 62302010, Grant 62136008, Grant 62276001, Grant 62206003, and Grant U21A20512, in part by Anhui Provincial Natural Science Foundation under Grant 2308085J03, and in part by China Postdoctoral Science Foundation under Grant 2023M740015. *(Corresponding authors: Ye Tian; Xingyi Zhang.)*

Shangshang Yang, Xiangkun Sun, and Ke Xu are with the Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, School of Artificial Intelligence, Anhui University, Hefei 230601, China (e-mail: yangshang0308@gmail.com; ahusxk@163.com; xuke@ahu.edu.cn).

Yuanchao Liu is with the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang 110819, China, and also with the College of Information Science and Engineering, Northeastern University, Shenyang 110819, China (e-mail: yuanchaoliu@126.com).

Ye Tian and Xingyi Zhang are with the Information Materials and Intelligent Sensing Laboratory of Anhui Province, School of Computer Science and Technology, Anhui University, Hefei 230601, China (e-mail: field910921@gmail.com; xyzhanghust@gmail.com).

Recommended for acceptance by L. Feng.

Digital Object Identifier 10.1109/TETCI.2024.3400867

I. Introduction

Deep neural networks (DNNs) have achieved significant success in various computer vision tasks, such as image classification [1], [2], object detection [3], and semantic segmentation [4]. However, it has been observed that DNNs are susceptible to adversarial attacks [5], [6]. These attacks involve making small perturbations to an input image, which is usually imperceptible to human eyes but can cause misclassification by DNNs. Over the past decade, researchers have made considerable efforts to improve DNNs' robustness against adversarial samples (i.e., perturbed images).

Existing approaches to enhancing DNNs' robustness can generally be divided into two types. The first type of approaches focuses on designing network architecture-agnostic training paradigms or strategies by leveraging different techniques. The representative techniques for training paradigms include defensive distillation using the teacher network's knowledge [7], gradient regularization/masking incorporating gradient penalty into loss functions [8], adversarial training (AT) by using continuously generated adversarial samples [5], [9], where AT has been the most popular and effective technique. In addition, researchers have also developed other network architecture-agnostic strategies, including feature squeezing [10], image preprocessing [11], model ensemble [12], and adding unlabeled data [13]. The second type of approaches explores the robustness of neural networks (NNs) from the perspective of network architecture design based on two different genres of ideas. The first genre of ideas is to manually design or adjust the architectures of NNs to enhance their robustness, such as increasing the network capacity [14], [15], adjusting the normalization position [16], and replacing harmful components with handcrafted modules [17]. Considering the great success made by neural architecture search (NAS) [18], [19], [20], [21], [22], the second genre of ideas is based on NAS [23] to design robust neural network architectures. After Cubuk et al. [24] first attempted to search robust convolution neural networks (CNNs) by directly using NAS-Net [25], robust NAS for CNNs has attracted a lot of researchers' attention and thus a series of approaches have been reported in recent years [26], [27], [28], [29], [30]. The representatives contain RobNet [27], RAS [26], AdvRush [28], and MORAS [30], which are based on cell-based search space for searching robust CNNs' architectures but utilize different metrics under different adversarial attacks to measure the network architecture robustness.

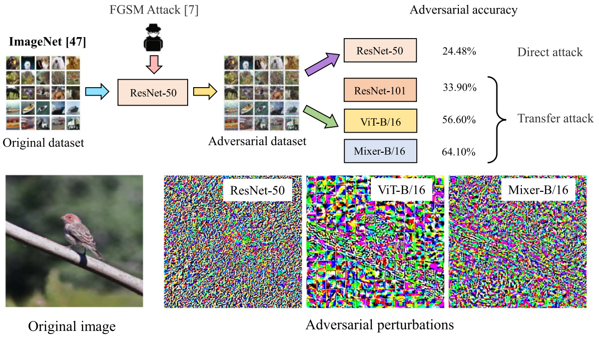


Fig. 1. The results of attacks transferred from ResNet-50 [31] to ResNet-101, ViT, and MLP-Mixer, and the visualization of adversarial perturbations to an image obtained by attacking three networks.

Although the architectures found by these robust NAS approaches hold promising robustness, these approaches merely focus on designing the metric to measure the architecture robustness, and they put less focus on the search space design [32] (still under the cell-based search space used for CNNs), which may limit the emergence of more robust architectures. More importantly, the above approaches are developed mainly based on a single type of networks, i.e., CNNs or vision Transformer (ViT) [33], and enhancing the robustness by designing hybrid architecture networks containing multiple types of networks has been hardly explored. However, recent research [34], [35] has revealed an intriguing fact that CNNs, ViT, and MLP-Mixer [36] can effectively defend against adversarial attacks transferred from each other, which is because the three types of networks process their input images very differently as shown in Fig. 1.

The above phenomenon indicates that, for the same image classification task, the target network can effectively defend against adversarial attacks transferred from the other types of networks except for the same type of networks [37], i.e, the attacks transferred from other types of networks are more difficult to be successful for target networks. That further inspires us with an intuitive idea: enhancing the network's robustness by adding different types of architectures to the network. Based on this, this paper proposes a new and different research perspective for network robustness: enhancing network robustness by designing hybrid architecture-based neural networks (containing multiple types of networks in the same network) manually or automatically. However, how much each type of architecture should remain in the network needs to be addressed. To achieve this paradigm, our natural idea is to employ the NAS technique to design robust hybrid architecture networks, resisting the attacks transferred from existing types of networks (i.e., CNNs, and variants of ViT and MLP-Mixer). But there still exist two challenges to the above idea: firstly, due to the big difference among existing network types in processing input features, it is difficult to design a search space to contain different types of networks in the same network framework; secondly, due to the higher complexity of ViT and MLP-Mixer than common CNNs, it is challenging to explore the complex search space efficiently.

To this end, we propose a hybrid architecture-based evolutionary neural architecture search approach, termed HA-ENAS, where a hybrid architecture-based search space is devised and an efficient multi-objective evolutionary algorithm (MOEA) is proposed to explore the search space. Our main contributions are as follows:

- To the best of our knowledge, we are the first to explore the design of hybrid network architectures for network robustness, and we propose a feasible paradigm for applying the NAS technique to search robust hybrid architecture networks. To combine or aggregate the modules of CNNs, Transformer, and multi-layer perception (MLP) in a unified network framework, we propose a multi-stage block-wise hybrid architecture network (HA-Net) for image classification, where each stage contains a certain number of blocks and the image resolution will be halved after each stage. Here, three types of blocks (i.e., convolution blocks, Transformer blocks, and MLP blocks) are devised as the candidates for each block. As a result, the hybrid architecture-based search space can be established based on the HA-Net and three types of blocks.
- To efficiently explore the search space, we first formulate the robust architecture search task as a bi-objective problem to maximize architectures' clean accuracy and adversarial accuracy, and then devise an efficient MOEA to solve the problem. In the MOEA design, considering the high complexity of hybrid architecture networks, a supernet-based retraining evaluation is first adopted to reduce both evaluation burden and coupled weight influence, and then a surrogate model is created to predict some architectures' accuracy instead of retraining them to reduce the cost further.
- Experimental results demonstrate that the hybrid architecture found by the proposed HA-ENAS holds better robustness against the adversarial attacks transferred from CNNs, ViT, and MLP-Mixer, and the found architecture also holds competitive robustness against the white-box attacks compared to comparison networks. Besides, the ablation study is also executed to demonstrate the effectiveness of the search space, the objectives, and the surrogate mode. Moreover, we further present a discussion about how to design robust hybrid architecture networks after observing the found architectures.

The rest of this paper is as follows: Section II introduces the related work, Section III presents the proposed approach, followed by the experiments, and Section V gives conclusions.

II. Related Work

A. Adversarial Attacks and Defense in Computer Vision

Generally, adversarial examples refer to the samples that are imperceptible to the human eyes but can make a DNN predict a no-true label. For a given input image $\mathbf{x}$ with label i and a target DNN $\mathcal{F}$, one of its adversarial samples $\mathbf{x}'$ can be generated by solving the following optimization problem:

$$\min_{\mathbf{x}'} ||\mathbf{x}-\mathbf{x}'||^{L_0,L_1,L_2,L_\infty}, \text{ s. t. } \mathcal{F}(\mathbf{x}') \neq i, \tag{1}$$

where $\mathcal{F}(\mathbf{x}')$ refers to the predicted label of $\mathbf{x}'$ by the model $\mathcal{F}$, and the first item is to minimize the distance between $\mathbf{x}$ and $\mathbf{x}'$ in terms of L_0 norm or L_1 norm or L_2 norm or L_∞ norm or multiple norms. Since there commonly exist a constraint ϵ to the distance, such as $||\mathbf{x}-\mathbf{x}'||^{L_0,L_1,L_2,L_\infty} \leq \epsilon$, the obtained adversarial sample $\mathbf{x}'$ cannot always fool the target DNN.

Existing adversarial attacks can be categorized into white-box and black-box attacks [38]. In white-box attacks, the adversary generally can access the detailed knowledge about the target network to be attacked, including the architecture, parameters, gradients, and so on. The representatives include FGSM [5], JSMA [39], and PGD [14]. In black-box attacks, the adversary does not know the model details but can only resort to querying access to generate adversarial samples. The representative approaches include SQUARE [40], OPA [41], and SA-ES [42] Compared to directly applying the attack approaches to target networks (called direct attacks), there is an important concept, i.e., transfer attacks or transfer-based attacks [30], referring to applying the adversarial samples generated for other networks to the target network.

Actually, in the real world transfer attacks and black-box attacks are broader and more common than white-box attacks, thus it is more necessary for DNNs to resist the former two, especially for transfer attacks. To assist DNNs to resist adversarial samples that are generated or transferred, researchers proposed to enhance the network robustness by leveraging different strategies from the perspectives of network training paradigm, preprocessing, and so on [43]. Specific strategies include defense distillation [7], gradient regularization [8], AT [5], feature squeezing [10], model ensemble [12], and so on [38], where AT is more time-consuming yet has been validated to be the most effective strategy.

B. *Robust Network Architecture Design*

In addition to above perspectives, some researchers also tried to explore the role of network architectures in resisting adversarial attacks, and thus developed some work on robust architecture design for image classification [32], where these approaches generally lie in two categories in terms of the architecture design manner.

The first type of approaches can be called manual design-based approaches, which manually adjust or design network architectures to enhance network robustness. As the early pioneers of manual design-based approaches, Madry et al. [14] manually increased the network capacity (network size) to enhance the CNNs' robustness; Xie and Yulle [16] manually adjusted the position of batch normalization and the network capacity to boost the CNNs' robustness; while Su et al. [44] tried to manually select the most robust network from existing CNNs. Due to the better performance of ViT than CNNs in computer vision, some recent research focuses on manually enhancing the robustness of ViT and its variants. For example, Mao et al. [17] first analyzed the effect of ViT's components on its robustness and then replaced some harmful components with manually designed modules to enhance ViT's robustness; Zhou et al. [45] manually added a channel attention branch in each ViT block to enhance ViT's robustness, where the added branch takes the same input as the MLP module and is combined with the MLP's output.

The second type of approaches is to automatically search robust architectures [32] in a given search space by leveraging the NAS technique, where their found architectures indeed show promising robustness due to the advantages of NAS [46], [47], [48]. We will briefly review some representative approaches in the following: the earliest attempt made by Cubuk et al. [24] directly adopts a reinforcement learning-based NAS approach NAS-Net to search robust CNNs in cell-based search space, where the architecture adversarial accuracy under the FGSM attack is taken as the reward; based on a multiple population-based evolutionary algorithm, RAS [26] also explores the cell-based search space to search robust CNNs and computes the architecture adversarial accuracy under adversarial samples, which are pre-generated by two black-box attacks on ResNet [31] and CapsNet [49]; similarly, AdvRush [28], CRoZe [29], RobNet [27], and MORAS [30] are still proposed for searching robust CNNs in the cell-based search space, and their difference lies in the adopted search strategies and the manners to measure the architecture robustness (i.e., the adversarial accuracy); specifically, ABanditNAS adopts the architecture accuracy as the architecture robustness after the adversarial training with FGSM is used for the network architecture, RobNet adopts the architecture accuracy under the PGD attack, and MORAS adopts the architecture accuracy under four white-box attacks and one black-box attack.

C. *Motivation of This Work*

The above work has demonstrated that network architectures play an important role in enhancing network robustness, and the NAS technique is also helpful to search more robust network architectures. But these robust NAS approaches only focus on the design of search strategies and architecture robustness metrics with less design in the search space, and they still adopt the most common cell-based search space to search pure CNN architectures, which may hinder the emergence of more robust network architectures.

More importantly, recently some researchers have found some inspiring facts [34], [35], [50]: for the same image classification task, ViT, MLP-Mixer, and their variants are more robust than CNNs when holding the same level of parameters, and the adversarial attacks transferred among ViT, MLP-Mixer, and CNNs are not promising. These phenomena imply that combining multiple different types of networks in the same network may enhance the network's robustness against transfer attacks from existing types of networks. To validate this, we execute a simple experiment on ImageNet [51]: first, generate adversarial samples for ResNet-50 and MLP-Mixer by FGSM, and then apply the generated samples to attack ResNet-101, ResMLP [52], and BoTNet [53], where ResMLP is the same type as MLP-Mixer and BoTNet is a handcrafted hybrid architecture network. We can see from Fig. 2 that the hybrid architecture network BoTNet is more robust to both transfer attacks while ResNet-101 and ResMLP are more vulnerable to transfer attacks from the same types of networks. Therefore, it is feasible and promising to design robust hybrid architecture networks to resist transfer attacks, where transfer attacks are more extensive than direct attacks in the real world.

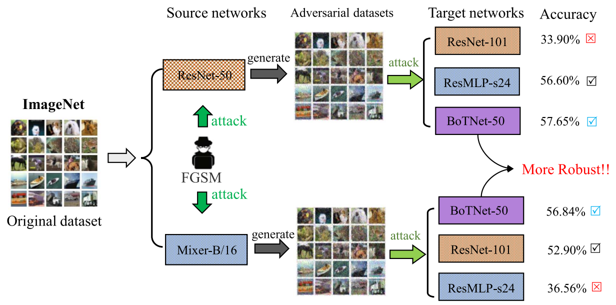


Fig. 2. The results of attacks transferred from ResNet-50 and MLP-Mixer to ResNet-101, ResMLP, and BoTNet on the ImageNet dataset.

For the above considerations, this paper aims to design hybrid architecture networks containing different types of networks by NAS, where the found architectures hold good robustness against transfer attacks from CNNs, ViT, and MLP-Mixer. To this end, we propose a hybrid architecture-based evolutionary neural architecture search approach, called HA-ENAS, where a hybrid architecture-based search space is devised to contain three types of networks and an efficient MOEA is employed to efficiently explore the search space. To the best of our knowledge, there is no publicly published work for searching robust ViT or MLP-Mixer architectures or designing hybrid architecture networks to enhance network robustness, especially by NAS.

III. The Proposed HA-ENAS

A. Overall Framework of HA-ENAS

Since the proposed HA-ENAS is to search robust hybrid neural architectures in the devised hybrid architecture-based search space, and the hybrid architectures (containing the modules of Transformer and MLP) are commonly more complex than CNNs, it is more time-consuming to train and evaluate each architecture. To effectively and efficiently explore the search space, the proposed HA-ENAS takes NSGA-II [54] as the basic framework to solve the formulated problem for the robust architecture search task and adopts a supernet-based retaining evaluation to reduce both evaluation cost and coupled weight influence, where the architecture robustness is measured under transfer attacks for fast evaluation. To further reduce the search cost, a surrogate model is used to assist the optimization of HA-ENAS by predicting some architectures' performance.

Fig. 3 summarizes the overall framework of the proposed HA-ENAS, which is mainly composed of four phases. To start with, a multi-stage block-wise hybrid architecture network is taken as the supernet and trained by a modified sandwich training rule. Then, three sets of adversarial samples are generated for CNNs, ViT, and MLP-Mixer by white-box attacks, which are used for computing architectures' robustness. Next, a surrogate model is built based on a certain number of randomly sampled architectures and their evaluation results under retraining. Finally, search robust hybrid architectures by leveraging a surrogate-assisted MOEA to solve a bi-objective optimization problem. For better understanding, Algorithm 1 also summarizes the procedure of the proposed HA-ENAS and presents more details about its fourth phase, i.e., the surrogate-assisted MOEA for architecture search, which contains multiple key steps: initialization (Lines 9–11), surrogate-assisted optimization (Lines 14–19), main optimization (Lines 20–22), and surrogate update (Lines 23–24). The main loop will be repeated until the evolution termination criterion is satisfied (Lines 12–25), after which the non-dominated individuals $\mathbf{P}_{non}$ will be output (Lines 26–27).

Algorithm 1: Procedure of HA-ENAS.

```
Input: $Gen$: maximum number of generations; $Pop$: population size;
  $Num_{Ad}$: number of adversarial samples to be generated; $Num_{Sa}$:
  number of architectures to be sampled; $Num_{So}$: number of generations
  for the surrogate-assisted optimization; $Num_E$: number of epochs for
  training the supernet; $Num_R$: number of epoch for retraining;
Output: $\mathbf{P_{non}}$: Non-dominated individuals;
  ———Phase 1: Train a supernet———
1:  $SN$ ←train a supernet under the devised search space for $Num_E$
    epochs by a modified sandwich rule; % elaborated in
    Section III-E1
  ———Phase 2: Generate adversarial samples———
2: $Ad \leftarrow \emptyset$; % adversarial sample sets
3: $Ad \leftarrow Ad \cup Num_{Ad}$ adversarial samples generated for a CNN;
4: $Ad \leftarrow Ad \cup Num_{Ad}$ adversarial samples generated for ViT;
5: $Ad \leftarrow Ad \cup Num_{Ad}$ adversarial samples generated for MLP-Mixer;
  ———Phase 3: Build a surrogate model———
6: $A \leftarrow Num_{Sa}$ randomly sampled architectures;
7: $A_{fit} \leftarrow$ retrain architectures in $A$ for $Num_R$ epochs based on the
   weights from $SN$ and obtain the objective values in (7) based on $Ad$;
   % supernet-based retraining evaluation
8: $Surro \leftarrow$ a regression model trained on $A$ and $A_{fit}$;
  ——Phase 4: Searching by a surrogate-assisted MOEA——
  // Initialization
9: $\mathbf{P} \leftarrow Pop$ randomly generated architectures, $g \leftarrow 1$;
10: $\mathbf{P}_{fit} \leftarrow$ retrain $\mathbf{P}$'s architectures for $Num_R$ epochs based on $SN$ and
    obtain the objective values in (7) based on $Ad$; % supernet-based
    retraining evaluation
11: $\mathbf{A}, \mathbf{A}_{fit} \leftarrow \mathbf{A} \cup \mathbf{P}, \mathbf{A}_{fit} \cup \mathbf{P}_{fit}$; % update archive
  - - - - - - - - - - - - - - Main loop starts- - - - - - - - - - - - - -
12: while $g \leq Gen$ do
13:   $\mathbf{R}, \mathbf{R}_{fit} \leftarrow \mathbf{P}, \mathbf{P}_{fit}$;
  - - - - - - - - Inner loop starts - - - - - - - -
 // surrogate-assisted optimization
14:   for $g\_inner = 1$ to $Num_{So}$
15:     $\mathbf{R}' \leftarrow$ select $Pop$ parents from $\mathbf{R}$; % Mating pool
16:     $\mathbf{T} \leftarrow$ Genetic Operator$(\mathbf{R}')$; % Generation
17:     $\mathbf{T}_{fit} \leftarrow$ predict $\mathbf{R}$'s fitness with $Surro$;
18:     $\mathbf{R}, \mathbf{R}_{fit} \leftarrow$ Environment Selection$(\mathbf{R} \cup \mathbf{T}, \mathbf{R}_{fit} \cup \mathbf{T}_{fit})$;
19:   end for
  - - - - - - - - Inner loop ends - - - - - - - -
  // Main optimization
20:   $\mathbf{Q} \leftarrow \mathbf{R}$; % potential solutions from surrogates
21:   $\mathbf{Q}_{fit} \leftarrow$ retrain $\mathbf{Q}$'s architectures for $Num_R$ epochs based on $SN$
      and obtain the objective values in (7) based on $Ad$; %
      supernet-based retraining evaluation
22:   $\mathbf{P}, \mathbf{P}_{fit} \leftarrow$ Environment Selection$(\mathbf{P} \cup \mathbf{Q}, \mathbf{P}_{fit} \cup \mathbf{Q}_{fit})$;
  // surrogate model update
23:   $\mathbf{A}, \mathbf{A}_{fit}, g \leftarrow \mathbf{A} \cup \mathbf{Q}, \mathbf{A}_{fit} \cup \mathbf{Q}_{fit}, g + 1$;%update archive
24:   $Surro \leftarrow$ update the model $Surro$ on $A$ and $A_{fit}$;
25: end while
  - - - - - - - - - - - - - - Main loop ends- - - - - - - - - - - - - -
26: Select non-dominated individuals $\mathbf{P_{non}}$ from $\mathbf{P}$ as the output
27: return $\mathbf{P_{non}}$;
```

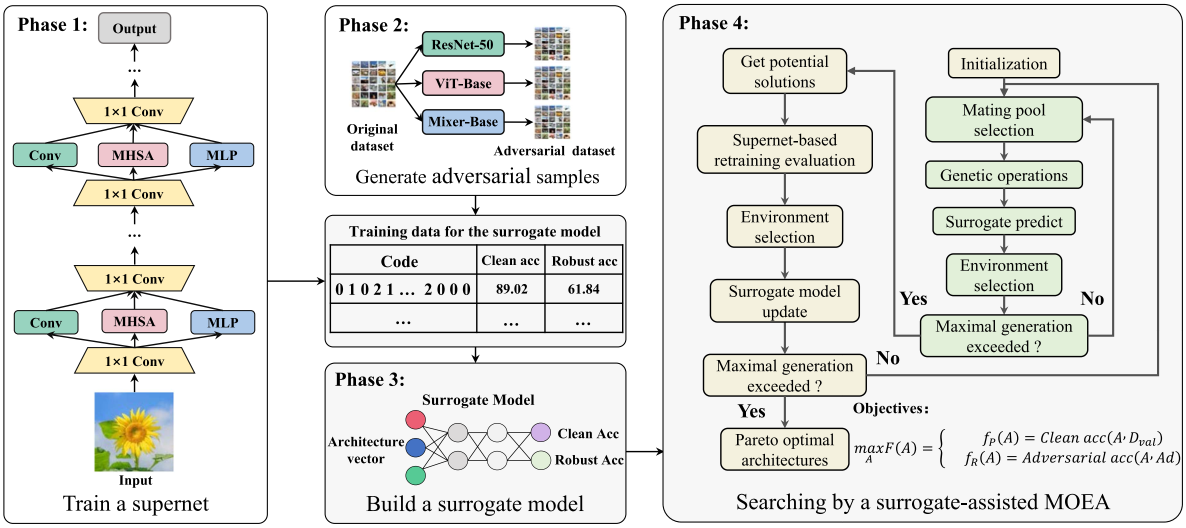


Fig. 3. The overall framework of the proposed HA-ENAS, consisting of four phases: training a supetnet for fast evaluation, generating adversarial examples for measuring networks' robustness, building a surrogate model for fitness prediction, and searching by a surrogate-assisted MOEA.

B. Hybrid Architecture-Based Search Space and Encoding

1) Hybrid Architecture-Based Search Space Design: To design robust hybrid architecture networks for image classification, we consider making the proposed search space contain both the convolution operations in CNNs and the basic modules of ViT and MLP-Mixer due to the great success of ViT and MLP-Mixer in vision tasks.

To maintain convolution (Conv) operations, multi-head self-attention (MSHA) modules, and MLP modules in the same network, we propose a multi-stage block-wise hybrid-architecture network (termed HA-Net) inspired by ResNet-50, and HA-Net has five stages with stage c1 fixed with a 7 × 7 Conv operation and stages c2 to c5 composed of different numbers of blocks. where the number of blocks at each stage is consistent with ResNet-50 and its overall architecture has been presented in Table I. Note that before each stage the input feature map resolution will be halved and the channel number will double by pooling operations, and it is obvious that the architecture of HA-Net is determined by the block types at each stage.

To make the designed blocks hold the modules of Conv, MHSA, and MLP, we arrange each block $BLOCK(C_1, C_2, C_3)$ as follows:

$$\begin{bmatrix} 1 \times 1,\ C_1 \\ \text{Conv}(\cdot) \text{ or MSHA}(\cdot) \text{ or MLP}(\cdot),\ C_2 \\ 1 \times 1,\ C_3 \end{bmatrix}, \tag{2}$$

where the first item and the third item are the 1×1 convolution operation, C_1 to C_3 represent the number of output channels, and the second item determines the type of the block: the block is named the **Conv block** or the **Transformer block** or the **MLP block** when $\text{Conv}(\cdot)$ or $\text{MHSA}(\cdot)$ or $\text{MLP}(\cdot)$ is used, respectively.

TABLE I
THE ARCHITECTURE OF HA-NET

stage	output	ResNet-50	HA-Net
c1	112x112	7x7,64,stride2	7x7,64,stride2
		3x3 max pool,stride 2	3x3 max pool,stride 2
c2	56x56	$\begin{bmatrix} 1\times1,64 \\ 3\times3,64 \\ 1\times1,256 \end{bmatrix} \times 3$	$[BLOCK(64, 64, 256)] \times 3$
c3	28x28	$\begin{bmatrix} 1\times1,128 \\ 3\times3,128 \\ 1\times1,512 \end{bmatrix} \times 4$	$[BLOCK(128, 128, 512)] \times 4$
c4	14x14	$\begin{bmatrix} 1\times1,256 \\ 3\times3,256 \\ 1\times1,1024 \end{bmatrix} \times 6$	$[BLOCK(256, 256, 1024)] \times 6$
c5	7x7	$\begin{bmatrix} 1\times1,512 \\ 3\times3,512 \\ 1\times1,2048 \end{bmatrix} \times 3$	$[BLOCK(512, 512, 2048)] \times 3$

In **Conv blocks**, the process of $\text{Conv}(\cdot)$ to handle the input feature X^{h*w*C} can be denoted by

$$Y = X \otimes W \oplus \mathbf{b},\ W \in R^{k*k*C*C_{out}}, \mathbf{b} \in R^{C_{out}}, \tag{3}$$

where $Y \in R^{h*w*C_{out}}$ is the output feature with C_{out} channels, k denotes the kernel size of Conv (k=3 in this paper), W is learnable parameter matrix, $\mathbf{b}$ is the bias, and $\otimes$ and $\oplus$ are the tensor multiplication and addition.

In **Transformer blocks**, $\text{MHSA}(X)$ can be denoted by

$$\text{MHSA}(X) = [SA_1(X); \cdots ; SA_h(X)]$$
$$* U_{mhsa}, U_{mhsa} \in R^{C \times C}$$

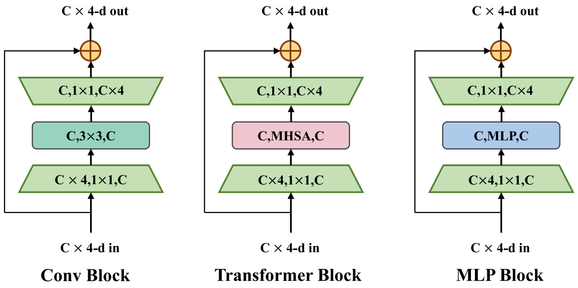


Fig. 4. Three types of devised blocks for HA-Net, whose difference is the middle module: use convolution, MHSA and MLP, respectively. C is the number of channels at each stage beginning.

$$SA_i(X) = \begin{cases} [Q, K, V] = X * U_{QKV},\ U_{QKV} \in R^{C\times\frac{3C}{head}} \\ A = softmax(Q * K'/\sqrt{\frac{C}{h}}) \\ SA_i(X) = A * V \end{cases}, \tag{4}$$

where $[\cdot]$ denotes the concatenation operation, $head$ is the head number (set to 1, 2, 4, 8 for stages c2 to c5), K' represents the transpose of K, $softmax(\cdot)$ denotes the softmax operation for normalization, and both U_{mhsa} and U_{QKV} are learnable parameters.

In **MLP blocks**, MLP$(\cdot)$ contains two types of MLP: token-mixing MLP mlp_t and channel-mixing MLP mlp_c, and its process of handing the input $X \in R^{hw*C}$ can be denoted as

$$MLP(X) = \begin{cases} X = mlp_t(X) = X + [\sigma(LN(X') \\ \qquad *W_{t1}) * W_{t2}]', W_{t1}, W_{t2} \in R^{hw\times hw}, \\ MLP(X) = mlp_c(X) = X + \\ \theta(LN(X) * W_{c1}) * W_{c2},\ W_{c1},\ W_{c2} \in R^{C\times C} \end{cases}, \tag{5}$$

where $LN(\cdot)$ is the layer normalization, σ denotes the activation function, and W_{t1}, W_{t2}, W_{c1}, and W_{c2} are learnable parameters.

Equipped with the above three blocks, the hybrid architecture-based search space $\mathcal{S}$ can be established under the framework of HA-Net. For a better understanding, Fig. 4 plots the architectures of three types of blocks. It can be seen that the bottleneck design [31] is used for each block to reduce the computational burden, i.e., $C_1 = C_2$ and $C_3 > C_2$ ($C_3 = 4 * C_2$ in this paper).

We further show the internal structures of the modules in Fig. 5. Our framework adopts the structure of ResNet, where the feature maps processed in the main process are all three-dimensional. However, both the Transformer and MLP blocks can only process two-dimensional inputs, it is necessary to perform an internal projection $proj(\cdot)$ to fuse the last two dimensions of the feature maps. By doing so, the input and output dimensions through each module can remain unchanged, allowing for versatile assembly of different blocks.

2) Encoding Strategy: Under the proposed search space, there theoretically exist 3^{16} architectures in total, and each candidate architecture $\mathcal{A}$ can be represented by an integer vector as

$$\mathcal{A} = (a_1, a_2, \ldots, a_i, \ldots, a_L) \in \{0, 1, 2\}^L, 1 \leq i \leq L, \tag{6}$$

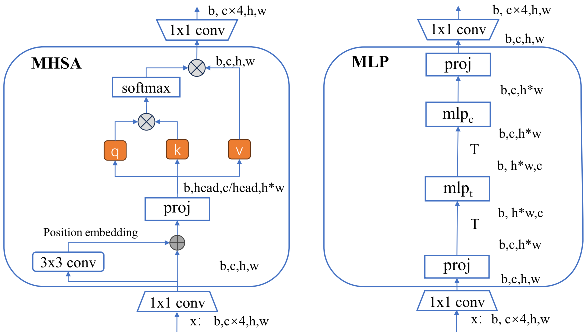


Fig. 5. The internal structure of the Transformer and MLP blocks, where T denotes the transpose of a matrix, b is the batch size. c, h, and w are the channel size, height, and width of feature map X. $proj(\cdot)$ is used to fuse the last two dimensions of the feature maps.

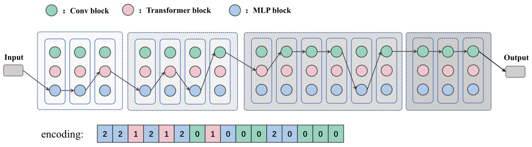


Fig. 6. An illustrative example of the encoding strategy.

where L is the encoding length determined by the number of blocks in the HA-Net (L=16 in this paper) and a_i indicates which type of blocks is used as the i-th block, where 0, 1, and 2 represent the Conv block, the Transformer block, and the MLP block, respectively. For a better understanding, Fig. 6 gives an illustrative example of the adopted encoding strategy.

C. Architecture Evaluation and Objectives

1) Supernet-Based Retraining Evaluation: Considering the huge computational burden of training each architecture from scratch for evaluation, the proposed HA-ENAS adopts common one-shot NAS approaches' idea: training a supernet in advance and evaluating each architecture with the weights in the supernet, which has been validated to be effective in significantly reducing the whole search cost [18], [55]. However, the supernet's weights are coupled after training due to the influence of different architectures on the same module, and thus there may cause inaccurate evaluations of some architectures [56]. This problem will be more severe when including more types of architectures in a supernet.

To reduce the influence of coupled weights for accurate evaluation, the proposed HA-ENAS further retrains each architecture for a certain number of epochs (Num_R) based on the trained supernet SN, which is called *supernet-based retraining evaluation* in this paper.

2) Problem Formulation and Objectives: To make the searched architectures hold high robustness and performance, given a dataset $Data = \{D_{train}, D_{val}, D_{test}\}$, we formulate the robust architecture search task as the following multi-objective optimization problem (MOP):

$$\max_{\mathcal{A}\in\mathcal{S}} F(\mathcal{A}) = \begin{cases} f_P(\mathcal{A}) = Clean\ accuracy(\mathcal{A}, D_{val}) \\ f_R(\mathcal{A}) = Adversarial\ accuracy(\mathcal{A}, Ad) \end{cases}, \tag{7}$$

where $\mathcal{A}$ is a candidate architecture well-trained on the training dataset D_{train} by the supernet-based retraining evaluation. Here, $f_P(\mathcal{A})$ represents the performance of architecture $\mathcal{A}$ (i.e., its clean accuracy on the validation dataset D_{val}), while $f_R(\mathcal{A})$ denotes the robustness of architecture $\mathcal{A}$, which is measured by the adversarial accuracy of $\mathcal{A}$ on adversarial samples in Ad.

It is worth noting that there are commonly three methods [26], [30], [57] to measure the robustness of architecture $\mathcal{A}$ against the adversarial samples in Ad. The first method [27], [57] is to compute the adversarial accuracy of $\mathcal{A}$ on Ad after architecture $\mathcal{A}$ is enhanced by AT, where the adversarial samples in Ad are directly generated for $\mathcal{A}$ by white-box or black-box attacks (i.e, direct attacks); the second one [24] is to compute the adversarial accuracy of $\mathcal{A}$ on Ad after $\mathcal{A}$ is well-trained without AT, where the adversarial samples are also generated for $\mathcal{A}$ by direct attacks; the third one [26] is nearly the same as the second one, but the adversarial accuracy of $\mathcal{A}$ is based on the transfer attacks, i.e., the adversarial samples in Ad are pre-generated for other networks by direct attacks.

Since the time cost of AT for each architecture is highly larger than the standard training, and it is also time-consuming to generate adversarial samples for each architecture by direct attacks [32], this paper adopts the idea of the third method to obtain the architecture robustness for fast evaluation. To make the searched architecture effectively resist transfer attacks from existing types of networks, we first obtain the adversarial sample set $Ad = \{Ad_{cnn}, Ad_{vit}, Ad_{mlp}\} = \{sample_1, sample_2, \ldots, sample_{Num_{3*Ad}}\}$, where Ad_{cnn}, Ad_{vit}, and Ad_{mlp} represent the sample sets generated by applying a white-box attack to ResNet-50, ViT, and MLP-Mixer, respectively. Then we directly compute the adversarial accuracy of architecture $\mathcal{A}$ as $f_R(\mathcal{A})$, where the typical FGSM [5] is used to generate the adversarial sample set Ad.

D. Surrogate Model Design and Management

Although the proposed supernet-based retraining evaluation can save an amount of computation sources compared to training architectures from scratch, it is still expensive to retrain each architecture for Num_R epochs. To further decrease the search cost, a surrogate model is created and continuously updated during the search to predict some architectures' objective values.

To start with, a simple M-layer fully connected (FC) neural network is used as $Surro$ to predict the objective values $\{f_P\hat{(}\mathcal{A}), f_R\hat{(}\mathcal{A})\}$ of architecture $\mathcal{A}$ by

$$\left.\begin{matrix} f_P\hat{(}\mathcal{A}) \\ f_R\hat{(}\mathcal{A}) \end{matrix}\right\} = Surro(A) = \begin{cases} h_1 = FC_1(\mathcal{A}) \\ h_i = FC_i(h_{i-1}), i \leq M-1 \\ f_P\hat{(}\mathcal{A}) = FC_M^1(h_{M-1}) \\ f_R\hat{(}\mathcal{A}) = FC_M^2(h_{M-1}) \end{cases}. \tag{8}$$

Here $FC_i(\cdot)$ refers to the i-th FC layer, $FC_1(\mathcal{A})$ maps the integer encoding vector of $\mathcal{A}$ to a D-dimension hidden vector $h_1 \in R^{1\times D}$, $h_i \in R^{1\times D}$ denotes i-th layer's hidden vector. $Surro$ has a total of M FC layers, and its M-th layer is composed of two parts $FC_M^1(\cdot)$ and $FC_M^2(\cdot)$, which are used to predict $f_P\hat{(}\mathcal{A})$ and $f_R\hat{(}\mathcal{A})$ for architecture $\mathcal{A}$, respectively.

Next, a certain number of architectures are randomly sampled and stored in an archive A, and their objective values are also obtained by supernet-based retraining evaluation and stored in another archive A_{fit}. With the collected data $A = \{a_i | 1 \leq i \leq |A|\}$ and $A_{fit} = \{(f_P(a_i), f_R(a_i)) | 1 \leq i \leq |A|\}$, the surrogate model $Surro$ is trained by minimizing the basic cross entropy loss.

Afterward, the surrogate model is used to assist the optimization process of NSGA-II as shown in the phase 4 of Fig. 3. Specifically, an inner loop same as the procedure of NSGA-II (lines 14–19 in Algorithm 1) is created to continuously generate solutions whose fitness are predicted by the surrogate model. By doing so, the potential solutions generated from the inner loop can accelerate and improve the convergence of the MOEA. Utilizing surrogate model offers the advantage of low computational cost, enabling us to explore a broader range of solutions through cross-mutation over multiple generations. Considering that the surrogate model's predictions may be imprecise, we have to evaluate all architectures' performance by the supernet-based retraining evaluation. Since we have re-evaluated all architectures generated from the inner loop by the supernet-based retraining evaluation, it is important to update the surrogate model [58], [59] based on newly generated architecture data, by which the updated surrogate model can make more accurate predictions. In brief, newly generated architecture $\mathbf{O}$ and their exactly evaluated objective values $\mathbf{O}_{fit}$ will be added to A and A_{fit}, and then retrain the surrogate model $Surro$ from scratch based on the updated data by optimizing the loss.

E. Related Details

In the proposed HA-ENAS, the mating pool selection and the environment selection are the same as that of NSGA-II [54]. The single-point crossover and bit-wise mutation [60] are used as the genetic operator in the inner loop to generate offspring. In addition, there also contain other two techniques, including the modified sandwich training rule and population initialization.

1) Modified Sandwich Training Rule: To make the supernet be trained well, we consider adopting the sandwich training rule, which has been the most effective way and widely adopted by one-shot NAS approaches [61], [62]. The original sandwich training rule commonly employs three subnets for one batch of training, including the largest subnet, the smallest subnet, and a randomly sampled subnet, where three forward passes and one backward pass are executed for one batch update.

To make the training rule adapt to the proposed HA-Net, we will utilize $N+3$ $(1 \leq N)$ subnets for one batch of training, including the full convolution subnet $\mathcal{A}_{con}$, the full Transformer subnet $\mathcal{A}_{trans}$, the full MLP subnet $\mathcal{A}_{mlp}$, and N randomly sampled subnets $\{\mathcal{A}_{rand}^1, \ldots, \mathcal{A}_{rand}^N\}$, whose encoding is represented by

$$\begin{cases} \mathcal{A}_{con} = (0, 0, \ldots, 0, \ldots, 0) \\ \mathcal{A}_{trans} = (1, 1, \ldots, 1, \ldots, 1) \\ \mathcal{A}_{mlp} = (2, 2, \ldots, 2, \ldots, 2) \\ \mathcal{A}_{rand}^j = (\cdots, a_i, \cdots) \in \{0, 1, 2\}^L, 1 \leq i \leq L,\ 1 \leq j \leq N \end{cases}. \tag{9}$$

By doing so, there are $N+3$ forward passes and one backward pass executed in one batch for updating the supernet. Note that the full convolution subnet $\mathcal{A}_{con}$ is exactly the classical ResNet-50 network.

2) Population Initialization: For better diversity and convergence of the initial population, an intuitive yet effective strategy is suggested to initialize the population instead of random initialization.

To be specific, half of the individuals in the population $\mathbf{P}$ are randomly generated, while another half of the individuals in $\mathbf{P}$ are generated as follows:

1) Firstly, store the three architectures $\mathcal{A}_{con}$, $\mathcal{A}_{trans}$, and $\mathcal{A}_{mlp}$ (anchor individuals) in $\mathbf{P}_{half} = \emptyset$;
2) secondly, mate individuals in $\mathbf{P}_{half}$, apply the crossover with the cross point fixed at $\frac{L}{2}$ for generating new individuals, and store them in $\mathbf{P}_{half}$;
3) thirdly, mate individuals in $\mathbf{P}_{half}$, apply the crossover with a cross point in $\{\frac{L}{4}, \frac{2L}{4}, \frac{3L}{4}\}$ for generating new individuals, and store them in $\mathbf{P}_{half}$;
4) repeatedly execute a similar operation to the above until the size of $\mathbf{P}_{half}$ meets the need.

IV. Experiments

In this section, we conduct the experiments to answer the following research questions:

- *RQ1:* How about the robustness of the architecture found by HA-ENAS against transfer attacks from existing networks compared to state-of-the-art neural architectures?
- *RQ2:* How about the robustness of the best-found architecture against direct white-box attacks compared to existing neural architectures?
- *RQ3:* How about the effectiveness of the proposed hybrid architecture-based search space, the adopted objectives, and the surrogate model on HA-ENAS?
- *RQ4:* What experience can the found hybrid architectures bring to researchers for designing more robust networks?

A. Experimental Settings

1) Datasets: Most experiments are conducted on the CIFAR-10 [63] dataset, and the CIFAR-100 [63] dataset is also used. CIFAR-10 is a 10-class natural image dataset consisting of 50,000 training images and 10,000 testing images. CIFAR-100 is a 100-class natural image dataset consisting of 50,000 training images and 10,000 testing images. The original size of each color image in CIFAR-10 and CIFAR-100 is 32 × 32, but we resize each color image to 224 × 224 since existing typical networks, including ResNet [31] series, ViT [33], MLP-Mixer [36], and so on, are predefined to handle images with sizes larger than 224 × 224, which commonly provides better performance than handling smaller-size images.

2) Peer Competitors: To demonstrate the robustness of the best architecture found by HA-ENAS, various state-of-the-art networks are selected as peer competitors for comparison. The selected competitors can be roughly divided into two types. The first type of competitors is some classical CNN architectures, including VGG-16, VGG-19 [64], ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-152 [31], Wide ResNet [65], ShuffleNetV2 [66], MobileNetV2 [67], RobNet-large-v1 [27], AdvRush [28] and CRoZe [29], where RobNet-large-v1, AdvRush and CRoZe are pure CNN architectures searched by robust NAS approaches. The second type of competitors comprises non-CNN architectures, including ViT-S/16, ViT-B/16 [33], ResMLP-S24 [52], MLP-Mixer-B/16 [36] and BotNet-50 [53], where BotNet-50 is a hybrid architecture network consisting of CNNs and Transformer.

3) Search Settings:

Dataset Details: According to previous NAS approaches' experiences [30], the original training set is divided into two parts: a new training set D_{train} and the validation set D_{val} (by the ratio of 90%-10%).

Supernet (HA-Net) Settings: The number of blocks L in HA-Net is fixed at 16, and the channels of its different stages are the same as that of ResNet-50. To train the supernet, N is set to 2, and the standard SGD optimizer with momentum is employed, where the initial learning rate, the momentum rate, the L2 weight decay, batch size, and the number of training epochs Num_E are set to 0.1, 0.9, 5e-4, 128, and 300, respectively. In addition, a single-period cosine decay is used to adjust the learning rate.

Surrogate Model Settings: For the surrogate model $Surro$, the number of FC layers M is set to 6, the hidden dimension D is set to 200, the number of sampled architectures Num_{Sa} in the third phase is set to 100, and the Adam optimizer is used to minimize the loss to train and update the surrogate model, where the learning rate, L2 weight decay, batch size, and epoch number are set to 2e-4, 5e-4, 32, and 100, respectively.

MOEA Settings of HA-ENAS: Population size $Pop = 50$, retraining epoch number $Num_R = 5$, the maximum number of generations $Gen = 30$, the generations of surrogate-assisted optimization $Num_{So} = 20$, and the number of adversarial samples $Num_{Ad} = 10,000$.

4) Training Details: After obtaining Pareto optimal individuals, some promising individuals in the knee area will be selected for retraining. The architecture setting is consistent with HA-Net, and the SGD optimizer with momentum is also utilized, whose hyperparameters are the same as that of supernet training but the OneCycleLR [68] strategy is employed to adjust the learning rate.

5) Comparison Methodology: To compare the performance of different network architectures, the test accuracy on clean samples (*clean accuracy*) and the test accuracy on adversarial samples (*adversarial accuracy*) obtained by them will be reported for comparison, where the latter commonly measures the robustness of network architectures, both clean samples and adversarial samples are generated on test datasets.

Note that there are two manners to obtain adversarial samples for a target network: the first is to directly generate adversarial samples for the target network by white-box and black-box attacks (direct attacks), and the second is to first generate adversarial samples for other networks (i.e., source networks) by white-box or black-box attacks and then apply the samples to the target network (transfer attacks).

Transfer Attack Settings: Three types of representative networks are used as source networks, including ResNet-50, ViT-B/16, and Mixer-B/16, which are variants of ResNet, ViT, and MLP-Mixer. Five attack approaches are utilized to generate

TABLE II
THE PERFORMANCE COMPARISON BETWEEN THE ARCHITECTURE HA-NET-A1 FOUND BY THE PROPOSED HA-ENAS AND COMPARED NETWORKS IN TERMS OF CLEAN ACCURACY AND ADVERSARIAL ACCURACY UNDER **TRANSFER ATTACKS** (AVERAGED ON 30 RUNS) ON CIFAR-10 DATASET

Model	**Clean**	**FFGSM**			**MI-FGSM**			**PGD**		
	Accuracy	ViT-B/16	ResNet-50	Mixer-B/16	ViT-B/16	ResNet-50	Mixer-B/16	ViT-B/16	ResNet-50	Mixer-B/16
VGG-16	94.55%	55.73%	52.11%	55.23%	55.94%	53.75%	55.14%	57.07%	53.51%	56.22%
VGG-19	95.03%	57.91%	53.38%	57.32%	58.07%	54.93%	56.95%	58.98%	55.86%	58.26%
WideResNet-50-2	97.32%	46.06%	41.94%	45.70%	46.19%	41.37%	45.35%	47.81%	42.07%	46.70%
ShuffleNetV2-0.5	92.26%	35.47%	32.43%	34.85%	35.85%	34.06%	35.04%	36.59%	34.34%	35.62%
MobileNetV2	96.07%	44.84%	41.35%	44.23%	44.83%	41.87%	43.73%	46.81%	44.83%	45.38%
ResNet-18	96.24%	51.36%	47.63%	50.37%	52.97%	48.43%	49.98%	52.97%	49.43%	51.43%
ResNet-34	97.06%	46.64%	43.89%	46.15%	48.45%	44.84%	45.80%	48.45%	45.84%	47.24%
ResNet-50	97.14%	44.20%	6.87%	44.35%	44.18%	2.55%	44.23%	45.80%	1.45%	45.29%
ResNet-101	97.50%	53.34%	49.08%	52.91%	55.37%	50.25%	52.39%	55.37%	50.52%	53.78%
ResNet-152	97.76%	53.27%	47.27%	53.20%	54.94%	49.61%	52.95%	54.94%	49.61%	54.12%
RobNet-large-v1 †	94.59%	57.98%	58.74%	55.34%	58.03%	58.32%	54.88%	59.29%	59.54%	56.61%
AdvRush	93.34%	52.67%	53.25%	51.06%	52.58%	52.93%	50.45%	53.86%	54.49%	51.87%
CRoZe	94.40%	53.99%	54.40%	52.18%	51.43%	54.24%	51.75%	54.92%	54.99%	52.92%
ViT-S/16	97.18%	41.29%	59.25%	48.86%	41.63%	59.01%	48.08%	44.66%	60.08%	50.73%
ViT-B/16	**98.71%**	8.32%	59.46%	49.54%	5.48%	59.26%	48.48%	2.93%	60.52%	51.31%
ResMLP-S24	96.32%	55.94%	61.79%	45.56%	56.28%	61.39%	47.39%	58.87%	62.26%	45.07%
Mixer-B/16	96.20 %	62.02%	**68.14%**	11.27%	61.94%	**67.54%**	8.96%	64.92%	**68.38%**	6.96%
BoTNet-50 †	95.22%	57.99%	57.51%	55.84%	57.61%	56.33%	55.48%	59.18%	58.65%	57.08%
HA-Net-A1 †	94.79%	**64.99%**	64.91%	**63.38%**	**64.78%**	64.50%	**62.82%**	**66.10%**	65.97%	**64.09%**

The best result in each column is bold, and the second-best result in each column is underlined;
'†' refers that the networks are retrained from scratch since there are no publicly available weights pre-trained on ImageNet [51].
Three **white-box attack** approaches are applied to three networks for generating adversarial samples as transfer attacks.

adversarial samples for the above networks, including FFGSM [69], MI-FGSM, PGD, SQUARE, and signSGD [70], their attack settings are as follows: the perturbation constraint ϵ is set to 8/255, PGD and MI-FGSM have 10 attack steps with each step size of 1/255, the number of queries for SQUARE is set to 200, and signSGD is set to hold 50 attack steps.

Direct attack Settings: Four white-box attack approaches are used as direct attacks, which are FGSM, FFGSM, PGD, and MI-FGSM, and their attack settings are as follows: the perturbation constraint ϵ is set to 8/255, the iterative approaches are set to hold 10 attack steps with each step size of 1/255.

All experiments are implemented with PyTorch and conducted on one NVIDIA RTX 3090 GPU. We use the torchattacks[1] library to implement our attack algorithm. For a fair comparison, all compared networks are fine-tuned[2]or retrained[3] (determined by whether the pre-trained weights of networks are provided) on the CIFAR-10 dataset by leveraging the same scripts. The source code of the proposed HA-ENAS can be available at https://github.com/BIMK/HA-ENAS.

B. RQ1: Effectiveness Against Transfer Attacks

Since this paper aims to design robust network architectures against transfer attacks from existing types of networks (i.e, CNNs, and variants of ViT and MLP-Mixer), it is necessary to validate the effectiveness of the best architecture found by HA-ENAS by comparing it with state-or-the-art network architectures. Table II summarizes the performance comparison on the CIFAR-10 between the best architecture HA-Net-A1 found by HA-ENAS and compared network architectures, in terms of clean accuracy and adversarial accuracy averaged on 30 runs. Here the adversarial accuracy of each network architecture is computed under the transferred adversarial samples, where the samples are generated on ViT-B/16 or ResNet-50 or Mixer-B/16 by a white-box attack. For more convincing, three white-box attacks: FFGSM, MI-FGSM, and PGD are used, and thus there report nine types of adversarial accuracy in Table II for each network architecture to show its robustness.

As can be observed from Table II, the adversarial accuracy of HA-Net-A1 is significantly better than that of pure CNN architectures, including handcrafted VGG series and ResNet series. These pure CNN architectures hold relatively good defensive abilities against transfer attacks from ViT-B/16 and Mixer-B/16, but their defensive abilities become worse when the transfer attacks are from the same type of network ResNet-50. Even compared to robust NAS-based architectures: RobNet-large-v1, AdvRush, and CRoZe, the robustness difference between HA-Net-A1 and them is still significant. But RobNet-large-v1, AdvRush, and CRoZe hold good robustness against the attacks from ResNet-50, which is because they are based on NAS and tailored to defend against attacks from CNNs. A similar observation can be obtained from the results of the ViT series and Mixer series (ResMLP-S24 is similar to Mixer-B/16), which can effectively defend against transfer attacks from ResNet-50 but are relatively vulnerable to transfer attacks from ViT-B/16 and Mixer-B/16, especially for the same type of networks as themself. Although the clean accuracy of HA-Net-A1 is no better than most compared networks since HA-Net-A1 is directly trained on CIFAR-10 instead of fine-tuning on ImageNet, HA-Net-A1 exhibits significantly good robustness against transfer

[1] [Online]. Available: https://github.com/Harry24k/adversarial-attacks-pytorch

[2] [Online]. Available: https://github.com/asyml/vision-transformer-pytorch/blob/main/src/train.py

[3] [Online]. Available: https://github.com/blackcow/pytorch-cifar-master/blob/main/main.py

TABLE III
THE PERFORMANCE COMPARISON BETWEEN THE ARCHITECTURE HA-NET-A1 FOUND BY THE PROPOSED HA-ENAS AND COMPARED NETWORKS IN TERMS OF CLEAN ACCURACY AND ADVERSARIAL ACCURACY UNDER **TRANSFER ATTACKS** (AVERAGED ON 30 RUNS) ON CIFAR-100 DATASET

Model	**Clean**	**FFGSM**			**MI-FGSM**			**PGD**		
	Accuracy	ViT-B/16	ResNet-50	Mixer-B/16	ViT-B/16	ResNet-50	Mixer-B/16	ViT-B/16	ResNet-50	Mixer-B/16
VGG-16	51.76%	24.63%	24.60%	24.32%	24.72%	24.13%	24.26%	24.91%	25.12%	24.78%
VGG-19	50.80%	23.55%	23.79%	23.42%	22.99%	23.48%	22.83%	23.64%	24.07%	23.67%
WideResNet-50-2	76.84%	27.38%	17.36%	26.58%	26.86%	15.19%	26.52%	27.51%	18.07%	27.57%
ShuffleNet-V2-0.5	67.59%	21.01%	18.72%	20.79%	20.95%	17.82%	20.30%	21.44%	20.11%	21.63%
MobileNetV2	68.92%	17.53%	15.36%	17.02%	17.28%	14.28%	16.86%	17.89%	16.33%	17.75%
ResNet-18	73.79%	24.63%	18.87%	23.36%	24.38%	17.14%	23.79%	25.09%	20.64%	24.50%
ResNet-34	75.12%	27.07%	21.72%	26.08%	26.61%	19.09%	26.18%	27.85%	22.87%	27.10%
ResNet-50	77.05%	27.51%	5.26%	27.07%	27.10%	2.60%	26.58%	28.28%	1.70%	28.03%
ResNet-101	77.35%	25.99%	16.77%	25.06%	25.37%	14.76%	24.81%	26.27%	18.04%	26.05%
ResNet-152	77.34%	28.16%	18.69%	27.66%	27.88%	15.78%	27.69%	28.74%	20.08%	28.81%
RobNet-large-v1	73.07%	27.21%	24.78%	26.65%	26.87%	24.00%	26.41%	27.88%	26.77%	27.55%
AdvRush	75.70%	27.51%	25.43%	26.73%	27.17%	24.91%	26.67%	28.34%	27.75%	27.57%
CRoZe	77.39%	27.88%	26.73%	27.01%	27.48%	26.08%	26.64%	28.09%	28.43%	27.85%
ViT-S/16	**84.65**%	24.35%	32.24%	24.41%	23.98%	31.90%	23.86%	25.68%	33.26%	26.49%
ViT-B/16	80.28%	6.96%	30.94%	24.72%	5.04%	30.72%	24.69%	3.62%	32.18%	26.76%
ResMLP-S24	84.33%	35.04%	**39.96%**	27.24%	33.98%	**39.36%**	26.16%	35.87%	**41.10%**	29.64%
Mixer-B/16	80.75%	27.94%	34.19%	4.12%	26.79%	34.47%	3.68%	28.25%	35.92%	3.09%
BoTNet-50	74.19%	25.96%	22.12%	25.80%	25.87%	21.01%	25.43%	26.42%	24.60%	26.52%
HA-Net-A1-C100	75.75%	**37.88%**	35.63%	**37.42%**	**37.52%**	34.94%	**37.48%**	**38.46%**	37.65%	**38.46%**

HA-Net-A1-C100 is the best architecture found by HA-ENAS on the CIFAR-100 dataset.
Three **white-box attack** approaches are applied to three networks for generating adversarial samples as transfer attacks.

attacks from three existing network architectures (i.e., CNNs, ViT, and MLP-Mixer). The robustness of HA-Net-A1 is only slightly worse than Mixer-B/16 in terms of against transfer attacks from ResNet-50 due to the smaller network size, but HA-Net-A1 holds more balanced robustness than Mixer-B/16. For the comparison results on CIFAR-100 in Table III, the same observation as that on CIFAR-10 can be obtained: the architecture HA-Net-A1-C100 found by HA-ENAS on CIFAR-100 also holds the nearly best robustness. Therefore, we can conclude that the architecture HA-Net-A1 (HA-Net-A1-C100) found by the proposed HA-ENAS is more robust than compared networks and can effectively resist transfer attacks from existing networks.

TABLE IV
MODEL SIZE AND COMPLEXITY

Model	Parameters	FLOPs	Model	Parameters	FLOPs
VGG-16	134.30M	15.48G	RobNet-large-v1	11.99M	1.43G
VGG-19	139.61M	19.64G	AdvRush	12.74M	1.59G
WideResNet-50-2	66.85M	11.42G	CRoZe	16.73M	2.00G
ShuffleNet-V2-0.5	352.04K	41.51M	ViT-S/16	21.57M	4.24G
mobilenet_v2	2.23M	312.86M	ViT-B/16	57.26M	11.27G
ResNet-18	11.18M	1.81G	ResMLP-S24	29.58M	5.96G
ResNet-34	21.29M	3.67G	Mixer-B/16	59.08M	12.60G
ResNet-50	23.52M	4.10G	BoTNet-50	18.81M	3.99G
ResNet-101	42.52M	7.83G	HA-Net-A1	34.50M	4.90G
ResNet-152	58.16M	11.55G			

The high robustness of HA-Net-A1 against transfer attacks from existing networks can be attributed to two aspects: the hybrid architectures contained in HA-Net-A1 and the assistance of the proposed NAS approaches. To validate this, we can compare the results of BoTNet-50 with that of other compared networks, where BoTNet-50 is also a hybrid architecture network. It can be seen from Table II that BoTNet-50 is more robust than most networks against transfer attacks, and its robustness is also more balanced as same as that of HA-Net-A1, which can validate that hybrid architectures are effective in assisting networks to resist transfer attacks to some extent. In addition, the comparison between BoTNet-50 and HA-Net-A1 demonstrates the effectiveness of the proposed HA-ENAS.

Besides, Table IV lists the model parameters sizes and the model computational complexity. Although larger networks usually represent better performance, this is not a determining factor when it comes to the robustness of the network. For example RobNet-large-v1 has a smaller number of parameters and computational complexity compared to ResNet-151, but it has better robustness. Similarly, our found architecture HA-Net-A1 has a smaller number of parameters and computational complexity compared to ViT-B/16, but it performs better against transfer attacks.

To further validate above conclusions, two black-box attacks SQUARE and signSGD are used to generate adversarial samples as transfer attacks, and the overall comparison results have been summarized in Table V. As can be seen, the robustness of the found architecture HA-Net-A1 is still superior over nearly all compared networks (including robust NAS-based networks), except for Mixer-B/16, and its robustness against transfer attacks from different networks is still balanced, which indicates that the robustness of HA-Net-A1 is strong enough to defend against most types of transfer attacks from existing networks. In summary, the above comparisons not only validate the effectiveness

TABLE V
THE ADVERSARIAL ACCURACY COMPARISON BETWEEN THE ARCHITECTURE HA-NET-A1 FOUND BY THE PROPOSED HA-ENAS AND COMPARED NETWORKS UNDER **TRANSFER ATTACKS** (AVERAGED ON 30 RUNS)

Model	**SQUARE**			**signSGD**		
	ViT-B/16	ResNet-50	Mixer-B/16	ViT-B/16	ResNet-50	Mixer-B/16
VGG-16	64.73%	61.18%	68.82%	59.47%	58.33%	59.30%
VGG-19	66.55%	64.09%	70.00%	61.41%	60.43%	61.45%
WideResNet-50-2	58.09%	54.64%	66.00%	50.91%	50.05%	50.78%
ShuffleNet-V2-0.5	53.82%	44.91%	61.27%	38.08%	37.33%	38.43%
MobileNetV2	60.45%	54.64%	65.00%	50.52%	48.34%	50.40%
ResNet-18	62.82%	57.55%	66.91%	56.00%	55.02%	56.00%
ResNet-34	60.09%	55.73%	66.82%	51.72%	50.64%	51.69%
ResNet-50	60.09%	28.09%	67.73%	50.14%	46.77%	50.26%
ResNet-101	64.64%	61.82%	69.00%	58.49%	58.14%	58.42%
ResNet-152	63.45%	59.45%	69.36%	58.22%	57.94%	58.16%
RobNet-large-v1	66.27%	66.09%	69.18%	61.27%	61.33%	61.21%
AdvRush	62.73%	59.36%	66.64%	55.79%	55.80%	55.78%
CRoZe	63.36%	61.36%	67.82%	56.61%	56.58%	56.43%
ViT-S/16	53.45%	63.00%	67.91%	60.48%	61.72%	61.44%
ViT-B/16	42.36%	67.27%	73.00%	59.01%	63.72%	63.26%
ResMLP-S24	66.77%	66.78%	55.43%	64.35%	64.51%	64.10%
Mixer-B/16	72.91%	**72.73%**	50.27%	**69.81%**	**69.89%**	67.31%
BoTNet-50	67.55%	64.64%	70.45%	61.55%	61.36%	61.48%
HA-Net-A1	**73.00%**	72.00%	**74.27%**	67.73%	67.89%	**67.75%**

The best result in each column is bold, and the second-best result in each column is underlined.
Two **black-box attack** approaches are applied to three networks for generating adversarial samples as transfer attacks.

TABLE VI
THE ADVERSARIAL ACCURACY COMPARISON BETWEEN THE ARCHITECTURE HA-NET-A1 FOUND BY THE PROPOSED HA-ENAS AND COMPARED NETWORKS UNDER TRANSFER ATTACKS FROM BOTNET-50 (AVERAGED ON 30 RUNS)

Model	**Clean**	BoTNet-50		
	Accuracy	FFGSM	MI-FGSM	PGD
VGG-16	94.55%	55.78%	54.82%	56.38%
VGG-19	95.03%	57.40%	56.59%	58.21%
WideResNet-50-2	97.32%	44.73%	43.34%	45.27%
ShuffleNet-V2-0.5	92.26%	35.43%	35.09%	36.35%
mobilenet_v2	96.07%	44.54%	43.56%	46.05%
ResNet-18	96.24%	49.78%	47.89%	50.42%
ResNet-34	97.06%	54.65%	50.80%	54.61%
ResNet-50	97.14%	45.18%	44.15%	45.29%
ResNet-101	97.50%	50.18%	44.80%	48.57%
ResNet-152	97.76%	53.05%	51.43%	53.43%
RobNet-large-v1	94.59%	54.59%	52.77%	55.09%
ViT-S/16	97.18%	59.17%	58.99%	59.69%
ViT-B/16	**98.71%**	60.65%	59.83%	60.85%
ResMLP-S24	96.32%	62.33%	61.81%	62.48%
Mixer-B/16	96.20%	**67.47%**	**66.62%**	**67.74%**
BoTNet-50	95.22%	8.60%	4.19%	2.36%
HA-Net-A1	94.79%	61.34%	59.95%	60.86%

The third-best result in each column is underlined with a tilde.

of hybrid architecture networks against transfer attacks but also show the effectiveness of the proposed NAS approaches.

Further, to verify the robustness of HA-Net-A1 against attacks transferred from hybrid-structure-based networks, the BoTNet-50 is used to generate adversarial samples, and the experimental results are presented in Table VI. It can be seen that, in terms of defending against the attacks transferred from BoTNet-50, HA-Net-A1 is more robust than the pure CNNs, holding similar robustness to ViT series networks but slightly underperforming MLP-based networks: ResMLP-S24 and Mixer-B/16. The above results are reasonable because both architecture HA-Net-A1 and BoTNet-50 are based on hybrid structures, where transfer attacks between the same types of networks are more aggressive. Despite that, the defense ability of HA-Net-A1 does not decrease too much, still holding highly competitive robustness, which further demonstrates the robustness of HA-Net-A1 and the effectiveness of the proposed approach.

C. *RQ2*: *Effectiveness Against Direct White-Box Attacks*

In addition to the network robustness in defending against transfer attacks, the network robustness in defending against direct attacks, especially white-box attacks, is also an important part, which is often investigated in some work related to network robustness. To this end, Table VII compares the adversarial accuracy of the best architecture HA-Net-A1 and partial comparison networks under direct attacks, where four white-box attacks FGSM, FFGSM, MIFGSM, and PGD are used. It can be found that ViT-B/16 holds the best adversarial accuracy under FGSM and the best clean accuracy due to the benefits of larger model parameters, and HA-Net-A1 holds competitive adversarial accuracy under FGSM and FFGSM but its robustness is slightly worse when two more strong attack approaches MI-FGSM and PGM are used. The reason behind this may be that there does not consider the objective of the architecture robustness against direct attacks in the proposed HA-ENAS.

To validate this, Table VII further presents the results of an architecture HA-Net-O1 found by a variant of HA-ENAS (termed *HA-ENAS(three)*), where *HA-ENAS(three)* considers an extra objective of measuring the architecture's robustness against direct attacks. As can be observed, the robustness of HA-Net-O1 is better than nearly all compared networks except for the result of ViT-B/16 under FGSM, and HA-Net-O1 is more robust than HA-Net-A1 at only a little expense of clean accuracy. As a result, we can conclude that HA-Net-A1 is effective in resisting

TABLE VII
THE PERFORMANCE COMPARISON BETWEEN THE ARCHITECTURES HA-NET-A1, HA-NET-O1, AND COMPARED NETWORKS IN TERMS OF CLEAN ACCURACY AND ADVERSARIAL ACCURACY UNDER **DIRECT ATTACKS** (AVERAGED ON 30 RUNS), WHERE FOUR **WHITE-BOX ATTACK** APPROACHES ARE USED

Model	Parameters (M)	Clean Accuracy	FGSM	FFGSM	MI-FGSM	PGD
ResNet-18	11.18M	96.24%	43.93%	10.70%	7.44%	5.34%
ResNet-34	21.29M	97.06%	41.25%	10.34%	6.74%	4.43%
ResNet-50	23.52M	97.14%	42.55%	6.93%	2.50%	1.43%
BotNet-50	18.81M	95.22%	40.69%	8.46%	4.20%	2.40%
ViT-B/16	57.26M	**98.71%**	**51.90**%	8.39%	5.49%	3.00%
HA-Net-A1	34.50M	94.79%	44.62%	**12.98**%	7.33%	4.54%
HA-Net-O1	29.89M	94.28%	44.72%	12.54%	**7.88**%	**6.19**%

"M" is short for "million" in "Parameters (M)".
The best result in each column is bold, and the second-best result in each column is underlined.

TABLE VIII
THE PERFORMANCE COMPARISON AMONG HA-NET-A1, HA-NET-O1, AND HA-NET-S1 IN TERMS OF CLEAN ACCURACY AND ADVERSARIAL ACCURACY UNDER **TRANSFER ATTACKS** (AVERAGED ON 30 RUNS), WHERE THREE **WHITE-BOX ATTACK** APPROACHES ARE USED

Model	Clean	FFGSM			MI-FGSM			PGD		
	Accuracy	ViT-B/16	ResNet-50	Mixer-B/16	ViT-B/16	ResNet-50	Mixer-B/16	ViT-B/16	ResNet-50	Mixer-B/16
HA-Net-O1	94.30%	63.53%	63.43%	61.42%	63.17%	62.96%	60.93%	64.27%	64.32%	62.49%
HA-Net-S1	94.29%	63.09%	62.83%	61.08%	62.84%	62.04%	60.45%	64.60%	64.03%	62.43%
HA-Net-S2	94.87%	63.90%	64.20%	62.24%	63.59%	63.53%	61.67%	64.71%	64.86%	62.90%
HA-Net-A1	94.79%	64.99%	64.91%	63.38%	64.78%	64.50%	62.82%	66.10%	65.97%	64.09%
HA-Net-A(more)	**95.06**%	**65.45%**	**65.61%**	**64.13%**	**65.45%**	**65.22%**	**63.47%**	**66.62%**	**66.44%**	**64.95%**
Search Cost Report	HA-ENAS		**One retraining evaluation** 9.1 minutes			**Total** 9.512 GPU days	HA-ENAS (more)		**Total** 19.020 GPU days	

The best results are highlighted. The search cost of HA-ENAS is also report.

TABLE IX
THE PERFORMANCE COMPARISON AMONG HA-NET-A1, HA-NET-B1, AND HA-NET-B2 IN TERMS OF CLEAN ACCURACY AND ADVERSARIAL ACCURACY UNDER **TRANSFER ATTACKS** (AVERAGED ON 30 RUNS), WHERE THREE **WHITE-BOX ATTACK** APPROACHES ARE USED

Model	Clean	FFGSM			MI-FGSM			PGD		
	Accuracy	ViT-B/16	ResNet-50	Mixer-B/16	ViT-B/16	ResNet-50	Mixer-B/16	ViT-B/16	ResNet-50	Mixer-B/16
HA-Net-A1	94.79%	64.99%	64.91%	63.38%	64.78%	64.50%	62.82%	66.10%	65.97%	64.09%
HA-Net-B1	95.00%	58.07%	58.09%	56.19%	58.95%	59.07%	55.18%	59.36%	58.98%	56.87%
HA-Net-B2	94.09%	60.81%	60.68%	58.74%	60.62%	59.83%	58.24%	61.90%	61.37%	59.62%

The best results are highlighted.

direct attacks though its robustness is not significantly better but competitive compared to comparison networks, and the superiority of HA-Net-O1's robustness to comparison networks also demonstrates that the proposed HA-ENAS is effective in finding robust architectures to defend against direct attacks when considering the extra objective.

D. *RQ3: Ablation Study*

This section will first validate the effectiveness of the adopted objectives and the surrogate model for the proposed HA-ENAS, and then analyze the efficacy of the devised hybrid architecture-based search space in the proposed HA-ENAS. Considering many variant architectures will be created subsequently, Table X summarizes their architecture details, including each architecture's notation/name, encoding, and descriptions.

To validate the effectiveness of the adopted objectives and surrogate model, we build two variant approaches of HA-ENAS: *HA-ENAS(three)* and *HA-ENAS(w/o surro)*, where *HA-ENAS(w/o surro)* is the HA-ENAS without the surrogate model (its main body is also NSGA-II). To further investigate the effectiveness of the surrogate, two additional variants are created: *HA-ENAS(ft)* and *HA-ENAS(more)*. *HA-ENAS(ft)* is the same as HA-ENAS, but its surrogate model is fine-tuned not retrained; *HA-ENAS(more)* continues searching on the results of HA-ENAS for the same number of evaluation, but its searching does not utilize the surrogate: i.e, *HA-ENAS(more)* equal to HA-ENAS plus *HA-ENAS(w/o surro)*. Table VIII compares the adversarial accuracy of the best architectures found by HA-ENAS and four variants under nine transfer attacks, where HA-Net-S1, HA-Net-S2, HA-Net-A1(more) are found by *HA-ENAS(w/o surro)*, *HA-ENAS(ft)*, and *HA-ENAS(more)*, respectively. The

TABLE X
DETAILS OF THE ARCHITECTURES FOUND BY VARIANTS OF HA-ENAS

Model	Encoding	Descriptions
HA-Net-A1	[[2, 1, 2], [0, 1, 2, 2], [0, 0, 0, 0, 0, 0], [0, 0, 0]]	found by HA-ENAS
HA-Net-A (more)	[[2, 1, 2], [0, 2, 2, 1], [0, 0, 0, 0, 0, 0], [0, 0, 0]]	continue searching on the results of HA-ENAS, using more retraining evaluation (extra 30 generations), without the surrogate
HA-Net-B1	[[0, 1, 0], [0, 1, 1, 0], [0, 0, 0, 0, 0, 0], [0, 0, 0]]	HA-ENAS under the search space only containing Conv and Transformer blocks
HA-Net-B2	[[2, 2, 2], [0, 2, 2, 2], [0, 0, 0, 0, 0, 0], [0, 0, 0]]	HA-ENAS under the search space only containing Conv and MLP blocks
HA-Net-S1	[[0, 2, 2], [0, 0, 2, 1], [0, 0, 0, 2, 0, 0], [0, 0, 0]]	found by HA-ENAS without the surrogate i.e., without the inner loop
HA-Net-S2	[[2, 1, 2], [0, 2, 1, 2], [0, 0, 0, 0, 0, 0], [0, 0, 0]	found by HA-ENAS, whose surrogate was fine-tuned iteratively
HA-Net-O1	[[0, 1, 2], [0, 0, 2, 2], [0, 0, 2, 0, 0, 0], [0, 0, 0]]	found by HA-ENAS with three objectives: clean acc, black acc, white acc
HA-Net-H1	[[0, 0, 0], [0, 0, 0, 0], [2, 2, 2, 2, 2, 2], [1, 1, 1]]	handcrafted architectures
HA-Net-H2	[[1, 1, 1], [2, 2, 2, 2], [0, 0, 0, 0, 0, 0], [0, 0, 0]]	
HA-Net-H3	[[2, 2, 2], [1, 1, 1, 1], [0, 0, 0, 0, 0, 0], [0, 0, 0]]	
HA-Net-H4	[[2, 1, 2], [2, 2, 0, 0], [0, 0, 0, 0, 0, 0], [0, 0, 0]]	

In Encoding, 0, 1, and 2 represent the Conv, Transformer, and MLP blocks, respectively;
Black acc denotes the accuracy on transfer adversarial examples, and white acc means the accuracy from under direct attack FGSM.

overall search cost of HA-ENAS and *HA-ENAS(more)* is also reported in Table VIII.

As can be seen from Table VII, the architecture HA-Net-O1 found by *HA-ENAS(three)* indeed holds good defensive ability against direct white-box attacks, but its robustness against transfer attacks is worse than HA-Net-A1 found by HA-ENAS as shown in Table VIII. Considering the lesser meaning of resisting direct attacks in the real world and the higher cost to compute the extra objective, the proposed HA-ENAS merely adopted two objectives. Besides, the comparisons between HA-Net-S1 and HA-Net-A1 showcase retraining the surrogate model provides better results than fine-tuning, which may be because the retrained surrogate model is more accurate than that of being fine-tuned. The comparisons between HA-Net-S1 and HA-Net-A1 demonstrate the effectiveness of the employed surrogate model, which can indeed assist HA-ENAS to find more robust architectures. That is reasonable because our approach can search for 1000 more individuals per generation by the surrogate model. Finally, although the architecture HA-Net-A(more) found by *HA-ENAS(more)* holds a little performance leading over HA-Net-A1, it is unworthy for *HA-ENAS(more)* to take an extra one times the cost as HA-ENAS to make the slight performance benefit, which further indirectly validates the effectiveness of the devised surrogate model.

To analyze the efficacy of the devised search space, we compare the best architectures found by HA-ENAS under different search spaces in Table IX, where HA-Net-B1 is found under the search space only containing Conv and Transformer blocks, and HA-Net-B2 is found under the search space only Conv and MLP blocks. As can be seen, the devised hybrid architecture-based search space can indeed enable HA-ENAS to find more robust network architectures against transfer attacks.

E. RQ4: Discussions

In this section, we aim to reveal some intriguing phenomena from the architectures found by HA-ENAS and give some useful instructions to researchers on how to design robust hybrid architecture networks.

For this aim, we first plot the architecture of HA-Net-A1 in Fig. 8, and it can be found that the second half of HA-Net-A1 (i.e., stage c4 and stage c5) is composed of Conv blocks while its first half (i.e., stage c2 and stage c3) is a hybrid architecture mainly consisting of Transformer blocks and MLP blocks.

For a deep insight into the architectures finally found by the proposed HA-ENAS, we first present its obtained Pareto front containing ten non-dominated individuals on the left of Fig. 7, where the horizontal and vertical axis represents the validation clean accuracy and adversarial accuracy of architectures, respectively. Then, based on the architectures encoded by these ten individuals, we make the statistic for the ratio of different blocks (i.e. Conv blocks, Transformer blocks, and MLP blocks) at each stage and show the statistical results on the right of Fig. 7. We can observe that the second half of these architectures is still mainly composed of Conv blocks, their stage c2 mainly consists of Transformer and MLP blocks, and their stage c3 contains three blocks with a fewer ratio of Transformer blocks, which is basically consistent with the observation of HA-Net-A1. As a result, we can have an intuitive idea of designing robust networks with hybrid architectures, where the robust network maintains Conv blocks as its second half but takes Transformer blocks or MLP blocks as its first half together with Conv blocks could be used in stage c3.

To investigate the effectiveness of the aforementioned robust architecture design idea, we first manually design four hybrid architectures HA-Net-H1 to HA-Net-H4 and show their architectures in Table X: HA-Net-H1 takes Conv blocks for the first half, MLP blocks and Transformer blocks for the second half, HA-Net-H2 and HA-Net-H3 are constructed according to the above design idea, while HA-Net-H4 is handcrafted according to the probability in Fig. 7. Then, Table XI compares the performance of the best architecture found by the proposed HA-ENAS and four handcrafted architectures. It is obvious and certain that HA-Net-A1 holds the best performance in terms of clean accuracy and adversarial accuracy compared to the others. Despite slightly worse clean accuracy than HA-Net-H1, the architecture HA-Net-H2 or HA-Net-H3 or HA-Net-H4 built under the above design instruction holds highly better

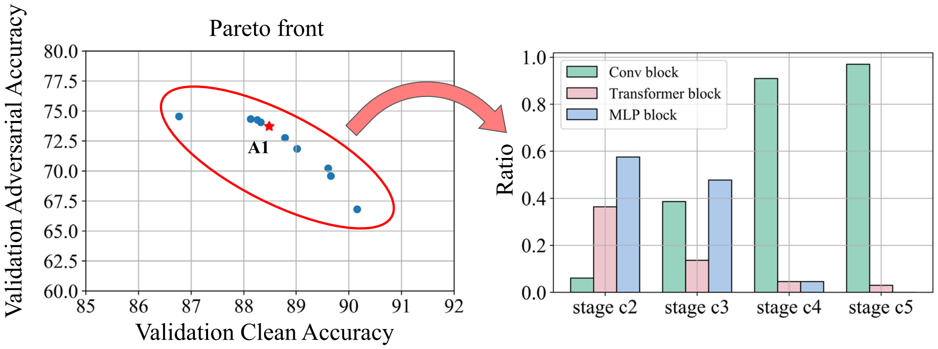


Fig. 7. Left: the Pareto front obtained by HA-ENAS, Right: the statistical results of the ratio of different blocks in each stage.

TABLE XI
THE PERFORMANCE COMPARISON AMONG HA-NET-A1, HANDCRAFTED ARCHITECTURES HA-NET-H1 TO HA-NET-H4 IN TERMS OF CLEAN ACCURACY AND ADVERSARIAL ACCURACY UNDER **TRANSFER ATTACKS** (AVERAGED ON 30 RUNS), WHERE THREE **WHITE-BOX ATTACK** APPROACHES ARE USED

Model	**Clean**	**FFGSM**			**MI-FGSM**			**PGD**		
	Accuracy	ViT-B/16	ResNet-50	Mixer-B/16	ViT-B/16	ResNet-50	Mixer-B/16	ViT-B/16	ResNet-50	Mixer-B/16
HA-Net-A1	94.79%	64.99%	64.91%	63.38%	64.78%	64.50%	62.82%	66.10%	65.97%	64.09%
HA-Net-H1	94.02%	59.07%	59.38%	57.18%	58.81%	58.63%	56.37%	57.94%	57.84%	58.10%
HA-Net-H2	93.85%	64.39%	64.79%	62.30%	64.26%	63.93%	61.94%	65.19%	65.17%	63.20%
HA-Net-H3	92.58%	62.70%	63.42%	61.30%	62.47%	62.78%	60.81%	63.59%	63.60%	61.79%
HA-Net-H4	93.41%	62.78%	62.58%	60.41%	62.54%	62.06%	59.88%	63.54%	63.45%	61.60%

The best results are highlighted.

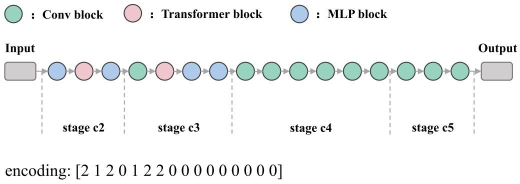


Fig. 8. The visualization of the best-found architecture HA-Net-A1.

robustness than HA-Net-H1 under whichever type of transfer attacks, which validates the effectiveness of the above robust architecture design idea to some extent.

V. CONCLUSION

This paper proposed a hybrid architecture-based evolutionary neural architecture search approach (HA-ENAS) to design hybrid architecture networks, which hold high robustness against transfer attacks from existing types of networks. To this end, a multi-stage block-wise hybrid architecture network is first devised to be able to contain different types of networks in a unified network framework, and thus a hybrid architecture-based search space is invented for HA-ENAS. To effectively and efficiently explore the search space, the robust architecture search is formulated as a MOP, and an efficient MOEA is employed to solve the MOP, where the supernet-based retraining evaluation and a surrogate model are used to reduce the search cost and accelerate the algorithm convergence. Experimental results demonstrate the high robustness of the hybrid architectures found by the proposed HA-ENAS compared to existing networks, the effectiveness of the devised search space, objectives, and the adopted surrogate model has also been validated. Besides, the observation on best-found architectures also gives an intriguing and convincing fact: the hybrid architecture networks, whose first half consists of Transformer or MLP blocks while the second half only contains convolution blocks, hold highly better robustness than other architectures.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[2] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 818–833.

[3] S. Ren, K. He, R. Girshick, X. Zhang, and J. Sun, “Object detection networks on convolutional feature maps,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 7, pp. 1476–1481, Jul. 2017.

[4] A. Shaban, S. Bansal, Z. Liu, I. Essa, and B. Boots, “One-shot learning for semantic segmentation,” in *Proc. Brit. Mach. Vis. Conf.*, London, UK, Sep. 4–7, 2017. [Online]. Available: https://www.dropbox.com/s/1odhw88t465klsz/0797.pdf

[5] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” in *Proc. 3rd Int. Conf. Learn. Representations*, San Diego, CA, USA, May 7–9, 2015. [Online]. Available: http://arxiv.org/abs/1412.6572

[6] H. Zanddizari, B. Zeinali, and J. M. Chang, “Generating black-box adversarial examples in sparse domain,” *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 6, no. 4, pp. 795–804, Aug. 2022.

[7] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, “Distillation as a defense to adversarial perturbations against deep neural networks,” in *Proc. IEEE Symp. Secur. Privacy*, 2016, pp. 582–597.

[8] C. Lyu, K. Huang, and H.-N. Liang, “A unified gradient regularization family for adversarial examples,” in *Proc. IEEE Int. Conf. Data Mining*, 2015, pp. 301–309.

[9] Z. Li, P. Xia, R. Tao, H. Niu, and B. Li, "A new perspective on stabilizing GANs training: Direct adversarial training," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 7, no. 1, pp. 178–189, Feb. 2023.

[10] W. He, J. Wei, X. Chen, N. Carlini, and D. Song, "Adversarial example defenses: Ensembles of weak defenses are not strong," in *Proc. 11th USENIX Conf. Offensive Technol.*, 2017, pp. 15–15.

[11] Y. Song, T. Kim, S. Nowozin, S. Ermon, and N. Kushman, "Pixeldefend: Leveraging generative models to understand and defend against adversarial examples," 2017, *arXiv:1710.10766*.

[12] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," 2017, *arXiv:1705.07204*.

[13] Y. Carmon, A. Raghunathan, L. Schmidt, J. C. Duchi, and P. S. Liang, "Unlabeled data improves adversarial robustness," 2019, *arXiv:1905.13736*.

[14] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. 6th Int. Conf. Learn. Representations*, Vancouver, BC, Canada, Apr. 30–May 3, 2018. [Online]. Available: https://openreview.net/forum?id=rJzIBfZAb

[15] S. Huang, Z. Lu, K. Deb, and V. N. Boddeti, "Revisiting residual networks for adversarial robustness: An architectural perspective," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 8202–8211.

[16] C. Xie and A. Yuille, "Intriguing properties of adversarial training at scale," in *Proc. 8th Int. Conf. Learn. Representations*, Addis Ababa, Ethiopia, Apr. 26–30, 2020. [Online]. Available: https://openreview.net/forum?id=HyxJhCEFDS

[17] X. Mao et al., "Towards robust vision transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 12042–12051.

[18] T. Elsken, J. H. Metzen, and F. Hutter, "Neural architecture search: A survey," *J. Mach. Learn. Res.*, vol. 20, no. 1, pp. 1997–2017, 2019.

[19] J. Dong, B. Hou, L. Feng, H. Tang, K. C. Tan, and Y.-S. Ong, "A cell-based fast memetic algorithm for automated convolutional neural architecture design," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 11, pp. 9040–9053, Nov. 2023.

[20] S. Deng, Z. Lv, E. Galván, and Y. Sun, "Evolutionary neural architecture search for facial expression recognition," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 7, no. 75, pp. 1405–1419, Oct. 2023.

[21] H. Tan et al., "RelativeNAS: Relative neural architecture search via slow-fast learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 1, pp. 475–489, Jan. 2023.

[22] Z. Lu, R. Cheng, Y. Jin, K. C. Tan, and K. Deb, "Neural architecture search as multiobjective optimization benchmarks: Problem formulation and performance assessment," *IEEE Trans. Evol. Comput.*, vol. 28, no. 2, pp. 323–337, Apr. 2024.

[23] X. Zhou, Z. Wang, L. Feng, S. Liu, K.-C. Wong, and K. C. Tan, "Towards evolutionary multi-task convolutional neural architecture search," *IEEE Trans. Evol. Comput.*, early access, Dec. 29, 2023, doi: 10.1109/TEVC.2023.3348475.

[24] E. D. Cubuk, B. Zoph, S. S. Schoenholz, and Q. V. Le, "Intriguing properties of adversarial examples," in *Proc. 6th Int. Conf. Learn. Representations*, Vancouver, BC, Canada, Apr. 30–May 3, 2018. [Online]. Available: https://openreview.net/forum?id=Skz1zaRLz

[25] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8697–8710.

[26] D. V. Vargas, S. Kotyan, and S. IIIT-NR, "Evolving robust neural architectures to defend from adversarial attacks," 2019, *arXiv:1906.11667*.

[27] M. Guo, Y. Yang, R. Xu, Z. Liu, and D. Lin, "When NAS meets robustness: In search of robust architectures against adversarial attacks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 631–640.

[28] J. Mok, B. Na, H. Choe, and S. Yoon, "AdvRush: Searching for adversarially robust neural architectures," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 12322–12332.

[29] H. Ha, M. Kim, and S. J. Hwang, "Generalizable lightweight proxy for robust NAS against diverse perturbations," *Adv. Neural Inf. Process. Syst.*, vol. 36, 2024.

[30] J. Liu and Y. Jin, "Multi-objective search of robust neural architectures against multiple types of adversarial attacks," *Neurocomputing*, vol. 453, pp. 73–84, 2021.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[32] C. Devaguptapu, D. Agarwal, G. Mittal, and V. N. Balasubramanian, "An empirical study on the robustness of NAS based architectures," 2020, *arXiv:2007.08428*.

[33] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[34] S. Bhojanapalli, A. Chakrabarti, D. Glasner, D. Li, T. Unterthiner, and A. Veit, "Understanding robustness of transformers for image classification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 10231–10241.

[35] R. Shao, Z. Shi, J. Yi, P.-Y. Chen, and C.-J. Hsieh, "On the adversarial robustness of vision transformers," 2021, *arXiv:2103.15670*.

[36] I. O. Tolstikhin et al., "Mlp-mixer: An all-mlp architecture for vision," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 24261–24272, 2021.

[37] P. Benz, S. Ham, C. Zhang, A. Karjauv, and I. S. Kweon, "Adversarial robustness comparison of vision transformer and mlp-mixer to CNNs," in *Proc. 32nd Brit. Mach. Vis. Conf.*, Nov. 22–25, 2021, p. 25. [Online]. Available: https://www.bmvc2021-virtualconference.com/assets/papers/0255.pdf

[38] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, vol. 6, pp. 14410–14430, 2018.

[39] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Proc. IEEE Eur. Symp. Secur. Privacy*, 2016, pp. 372–387.

[40] M. Andriushchenko, F. Croce, N. Flammarion, and M. Hein, "Square attack: A query-efficient black-box adversarial attack via random search," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 484–501.

[41] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 828–841, Oct. 2019.

[42] Z. Li, H. Cheng, X. Cai, J. Zhao, and Q. Zhang, "SA-ES: Subspace activation evolution strategy for black-box adversarial attacks," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 7, no. 3, pp. 780–790, Jun. 2023.

[43] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay, "A survey on adversarial attacks and defences," *CAAI Trans. Intell. Technol.*, vol. 6, no. 1, pp. 25–45, 2021.

[44] D. Su, H. Zhang, H. Chen, J. Yi, P.-Y. Chen, and Y. Gao, "Is robustness the cost of accuracy?–A comprehensive study on the robustness of 18 deep image classification models," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 631–648.

[45] D. Zhou et al., "Understanding the robustness in vision transformers," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 27378–27394.

[46] Z. Lu et al., "NSGA-Net: Neural architecture search using multi-objective genetic algorithm," in *Proc. Genet. Evol. Comput. Conf.*, 2019, pp. 419–427.

[47] H. Zhang, Y. Jin, R. Cheng, and K. Hao, "Efficient evolutionary search of attention convolutional networks via sampled training and node inheritance," *IEEE Trans. Evol. Comput.*, vol. 25, no. 2, pp. 371–385, Apr. 2021.

[48] L. He, B. Hou, J. Dong, and L. Feng, "Two-stage neural architecture optimization with separated training and search," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2023, pp. 1–8.

[49] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[50] Y. Bai, J. Mei, A. L. Yuille, and C. Xie, "Are transformers more robust than CNNs?," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 26831–26843, 2021.

[51] 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 Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[52] H. Touvron et al., "ResMLP: Feedforward networks for image classification with data-efficient training," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 5314–5321, Apr. 2023.

[53] A. Srinivas, T.-Y. Lin, N. Parmar, J. Shlens, P. Abbeel, and A. Vaswani, "Bottleneck transformers for visual recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 16519–16529.

[54] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[55] S. Hu, R. Cheng, C. He, and Z. Lu, "Multi-objective neural architecture search with almost no training," in *Proc. Int. Conf. Evol. Multi-Criterion Optim.*, 2021, pp. 492–503.

[56] J. Xu et al., "Analyzing and mitigating interference in neural architecture search," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 24646–24662.

[57] H. Chen et al., "Anti-bandit neural architecture search for model defense," in *Proc. 16th Eur. Conf. Comput. Vis.*, 2020, pp. 70–85.

[58] Y. Liu, J. Liu, Y. Jin, F. Li, and T. Zheng, "A surrogate-assisted two-stage differential evolution for expensive constrained optimization," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 7, no. 3, pp. 715–730, Jun. 2023.

[59] Y. Tian, S. Yang, L. Zhang, F. Duan, and X. Zhang, “A surrogate-assisted multiobjective evolutionary algorithm for large-scale task-oriented pattern mining,” *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 3, no. 2, pp. 106–116, Apr. 2019.

[60] Y. Tian, X. Zhang, C. Wang, and Y. Jin, “An evolutionary algorithm for large-scale sparse multiobjective optimization problems,” *IEEE Trans. Evol. Comput.*, vol. 24, no. 2, pp. 380–393, Apr. 2020.

[61] Z. Yue, B. Lin, Y. Zhang, and C. Liang, “Effective, efficient and robust neural architecture search,” in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2022, pp. 1–8.

[62] Q. Lin, Z. Fang, Y. Chen, K. C. Tan, and Y. Li, “Evolutionary architectural search for generative adversarial networks,” *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 6, no. 4, pp. 783–794, Aug. 2022.

[63] A. Krizhevsky and G. Hinton, “Learning multiple layers of features from tiny images,” Univ. Toronto, Toronto, ONT, Canada, Tech. Rep. 0, 2009.

[64] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *Proc. 3rd Int. Conf. Learn. Representations*, San Diego, CA, USA, May 7–9, 2015. [Online]. Available: http://arxiv.org/abs/1409.1556

[65] S. Zagoruyko and N. Komodakis, “Wide residual networks,” in *Proc. Brit. Mach. Vis. Conf.*, 2016.

[66] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, “Shufflenet v2: Practical guidelines for efficient CNN architecture design,” in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 116–131.

[67] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “Mobilenetv2: Inverted residuals and linear bottlenecks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4510–4520.

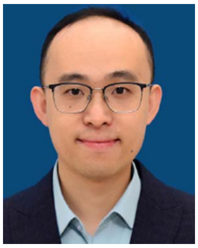
[68] L. N. Smith and N. Topin, “Super-convergence: Very fast training of neural networks using large learning rates,” in *Proc. Artif. Intell. Mach. Learn. Multi-Domain Operations Appl.*, vol. 11006, 2019, pp. 369–386.

[69] E. Wong, L. Rice, and J. Z. Kolter, “Fast is better than free: Revisiting adversarial training,” in *Proc. 8th Int. Conf. Learn. Representations*, Addis Ababa, Ethiopia, Apr. 26–30, 2020. [Online]. Available: https://openreview.net/forum?id=BJx040EFvH

[70] S. Liu, P.-Y. Chen, X. Chen, and M. Hong, “SignSGD via zeroth-order oracle,” in *Proc. Int. Conf. Learn. Representations*, 2019.

Shangshang Yang (Member, IEEE) received the B.Sc. and Ph.D. degrees from Anhui University, Hefei, China, in 2017 and 2022, respectively. He was a Visiting Ph.D. student with Bielefeld University, Bielefeld, Germany, in 2022. He is currently a Postdoctor with the School of Artificial Intelligence, Anhui University. His research interests include evolutionary multi-objective optimization, neural architecture search, intelligent education, and graph learning. He was the recipient of the 2023 International Conference on Data-driven Optimization of Complex Systems Best Paper Award.

Xiangkun Sun received the B.Sc. degree from Henan Polytechnic University, Jiaozuo, China, in 2020. He is currently working toward the master degree with the school of Artificial Intelligence, Anhui University, Hefei, China. His research interests include evolutionary neural architecture search and adversarial attack.



Ke Xu received the B.S. degree from the Hefei University of Technology, Hefei, China, in 2016, and the Ph.D degree from the Institute of Information Science, Beijing Jiaotong University, Beijing, China, in 2021. He is currently a Lecturer with the Artificial Intelligence Academy, Anhui University, Hefei. His research interests include neural network compression, high performance computing architectures for embedded applications, and computer vision.

Yuanchao Liu received the B.S. degree in automation from Northeastern University at Qinhuangdao, Qinhuangdao, China, in 2017, and the master degree in control theory and control engineering, and the Ph.D. degree in control science and engineering from Northeastern University, Shenyang, China, in 2019 and 2023, respectively. He is currently a Lecturer with the College of Information Science and Engineering, Northeastern University. His research interests include multiobjective optimization, robust optimization, dynamic optimization, and data-driven optimization.

Ye Tian (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees from Anhui University, Hefei, China, in 2012, 2015, and 2018, respectively. He is currently an Associate Professor with the School of Computer Science and Technology, Anhui University. His research interests include evolutionary computation and its applications. He was the recipient of the 2018, 2021, and 2024 IEEE Transactions on Evolutionary Computation Outstanding Paper Award, 2020 IEEE Computational Intelligence Magazine Outstanding Paper Award, and 2022 IEEE Computational Intelligence Society Outstanding Ph.D. Dissertation Award.

Xingyi Zhang (Senior Member, IEEE) received the B.S. degree from the Fuyang Normal College, Fuyang, China, in 2003, and the M.S. and Ph.D. degrees from the Huazhong University of Science and Technology, Wuhan, China, in 2006 and 2009, respectively. He is currently a Professor with the School of Computer Science and Technology, Anhui University, Hefei, China. His research interests include unconventional models and algorithms of computation, evolutionary multi-objective optimization, and logistic scheduling. He was the recipient of the 2018, 2021, and 2024 IEEE Transactions on Evolutionary Computation Outstanding Paper Award and 2020 IEEE Computational Intelligence Magazine Outstanding Paper Award