Multiple Population Alternate Evolution Neural Architecture Search

Juan Zou¹, Han Chu^{1*}, Yizhang Xia¹, Junwen Xu¹, Yuan Liu¹, Zhanglu Hou¹

¹Hunan Engineering Research Center of Intelligent System Optimization and Security, Xiangtan University

chuhan@smail.xtu.edu.cn

Abstract

The effectiveness of Evolutionary Neural Architecture Search (ENAS) is influenced by the design of the search space. Nevertheless, common methods including the global search space, scalable search space and hierarchical search space have certain limitations. Specifically, the global search space requires a significant amount of computational resources and time, the scalable search space sacrifices the diversity of network structures and the hierarchical search space increases the search cost in exchange for network diversity. To address above limitation, we propose a novel paradigm of searching neural network architectures and design the Multiple Population Alternate Evolution Neural Architecture Search (MPAE), which can achieve module diversity with a smaller search cost. MPAE converts the search space into L interconnected units and sequentially searches the units, then the above search of the entire network be cycled several times to reduce the impact of previous units on subsequent units. To accelerate the population evolution process, we also propose the the population migration mechanism establishes an excellent migration archive and transfers the excellent knowledge and experience in the migration archive to new populations. The proposed method requires only 0.3 GPU days to search a neural network on the CIFAR dataset and achieves the state-of-the-art results.

1 Introduction

Neural Architecture Search (NAS) is a method that utilizes machine learning techniques to automatically search and optimize neural network architectures. The goal of NAS is to improve the efficiency or performance of deep learning models by searching a smaller or better neural network. NAS typically employs various techniques, including reinforcement learning [Zoph and Le, 2016], gradient descent [Liu *et al.*, 2018b] and evolutionary computation [Xie and Yuille, 2017], to discover optimal neural network architectures that yield superior performance. This method has been widely used in image classification [Sun *et al.*, 2019c], object detection [Liang *et al.*, 2021] and natural language processing [Klyuchnikov *et al.*, 2022].

In evolution-based NAS, the performance of the model depends on the design of the search space. Nevertheless, common methods including the global search space, scalable search space and hierarchical search space have certain limitations. A widely adopted concept is the global search space that directly search the entire structure of the neural network to identify the most optimal configuration [Zhang et al., 2022]. However, this approach demands significant computational resources and time due to the expansive nature of the search space. The second strategy involves dividing the network into fundamental cells and constructing a more intricate network by stacking these cells together. The typical approach is cell-based NAS [Zoph et al., 2018], which first explores cells and then combines the cells by stacking and splicing the searched cells to create a complete network. The cell-based NAS reduces the search complexity and enhances the adaptability of the structure. However, [Tan et al., 2019] point out that the cell-based stacking structure compromises the diversity of network structures and does not fully take into account the characteristics and limitations of various parts of the entire network. Thirdly, [Tan et al., 2019] proposed a hierarchical search space with a fixed macro structure, it allows different layers to utilize different resolution blocks within the network structure. However, this approach incurs additional costs when searching for different layer structures. In order to balance the diversity of network structures with search costs, we introduce an alternate search approach. In a network formed by stacking N cells, We transform the search space into a space of L interconnected cells, and search the N cell spaces sequentially. This approach revisits the neural architecture search paradigm from a split perspective. The alternate search method splits a global search space into multiple small global search spaces to divide and conquer, greatly simplifying the complexity of the problem and improving search efficiency.

To address the challenges posed by a larger search space and the issue of lengthy network encoding [Zhang *et al.*, 2022] in the application of evolutionary algorithms to NAS, We propose a multi-population alternate evolutionary

^{*}Corresponding author

(MPAE) framework based on alternate search. Unlike other methods [Tian *et al.*, 2020] that utilize multiple populations to explore various equivalent Pareto-optimal solutions, the MPAE employs multiple populations to partition the extensive search space into subsets based on the network structure. Each population then samples from a different search space subset. This approach not only ensures an equal distribution of the lengthy network encoding across each population but also simplifies the search space that each individual population needs to explore.

To leverage the knowledge accumulated during the optimization process of each network module, we investigate the impact of inter-module migration within the network. Drawing inspiration from the repetitive structures commonly found in CNNs (e.g. the repetitive modules in Inception [Szegedy et al., 2017], ResNet [He et al., 2016] and NASNet [Zoph et al., 2018]), we observe that there exists a similarity in the excellent network structures of adjacent layers. Building upon this insight, we introduce the population migration mechanism, which harnesses the retained knowledge and experience of each population to expedite the evolutionary process. The population migration process includes three aspects: maintaining the migration archive, determining the number of migrating individuals for each population and selecting the migrating individuals. These migration archives are not solely selected based on accuracy, but also include potential excellent architectures based on the Pareto frontier. Based on the phenomenon that structural differences in network modules increase with distance, the populations choose different numbers of migrating individuals from the migration archive set of other populations based on their proximity. To ensure population diversity, the selection of migrating individuals is determined by the similarity values between individuals and the migrating population.

In summary, our contributions are as follows:

- Multiple Populations Alternate Evolution Framework: We proposed Multiple Populations Alternate Evolution (MPAE) framework to simplify the search space and meet the diversification needs of modules with a small search cost. MPAE uses multiple populations to split a global search space into multiple small global search spaces to divide and conquer, which greatly reduces the complexity of a single population to search.
- **Migration Mechanism:** we explore the effects caused by the migration of each module of the network to each other, and propose the population migration mechanism. The population migration mechanism constructs a migration archive set, and uses the Euclidean distance and similarity between network modules as indicators to select migrating individuals. This mechanism not only ensures the diversity of the population, but also makes full use of the knowledge retained in the optimization process of each network module to assist the search.

2 Related Work

The influence of the NAS search space on search outcomes [Yu *et al.*, 2019; Zhang *et al.*, 2020] has been found to be substantial. As NAS algorithms have evolved, the design of the search space has also evolved. Initially, in the early stages of NAS development, the common practice [Xie and Yuille, 2017; Sun et al., 2019c; Real et al., 2017] was to directly search the entire structure of the neural network. However, this global search approach proved to be resource-intensive and time-consuming. To improve search efficiency, scalable network architectures [Zoph et al., 2018; Liu et al., 2018b; Pham et al., 2018; Yang et al., 2020] were introduced. This approach involves constructing networks by repetitively stacking identical components, allowing for the extension of network components that were searched on smaller datasets to larger ones, but the scalable network architecture has sacrificed consequently the diversity of the network. And the studies [Girish et al., 2019; Zeiler and Fergus, 2014] have highlighted that convolutional networks learn distinct features at different layers. This realization emphasized the importance of block diversity in influencing the performance of network models. Consequently, a hierarchical search space [Cai et al., 2019; Tan et al., 2019; Wu et al., 2019] with a fixed macrostructure was adopted, it enabling different layers to utilize different resolution blocks within the network structure. However, these algorithms incurred an additional cost when searching for different layer structures. In contrast, our alternation evolution paradigm not only benefits from the scalable network structure but also allows for the search of different layer structures without incurring additional costs.

3 Method

3.1 Overview

We develop a multi-population alternate evolution strategy to search for neural architectures, *i.e.* MPAE. The search phase of MPAE consists of four parts, as depicted in Figure 1. The pipeline of the MPAE scheme is as follows:

Initialization: Initially, according to the *L*-layer structure of the neural network, the search space \mathcal{A} is divided into subsets \mathcal{A}_l . Then *N* candidate sub-networks are randomly selected from the search space \mathcal{A}_l to form the population pop_l . In practice, the subset \mathcal{A}_l of the search space is much smaller than the whole set \mathcal{A} , and we believe that such limited-sized subsets of the search space are naturally easier for search algorithms to handle to avoid the dilemma of large search spaces.

Population Conbination: After selecting the current evolutionary population P_i^t , the population P_i^t generates the offspring population Q_i^t through genetic operations. Then the two populations and the migration population M_i^t generated through the population migration mechanism (section 3.4) are merged into the combined population C_i^t .

Population and Migration Archive Update: All individuals in the combined population C_i^t are decoded into corresponding neural structures, inherit the supernet weights and undergo training evaluation. Then multi-objective environmental selection is used to select outstanding individuals on the Pareto front from the combined population C_i^t , and generate the next parent population P_i^{t+1} and the migration archive set M_i^{t+1} .



Step 3. Population and Migration Archive Update

Figure 1: Overall process of MPAE. Step 1 of the algorithm is to initialize multiple populations and immigrant archives. Step 2 is to generate offspring populations and merge them with parent populations and migrating populations. Step 3 involves select the parent populations for the next generation through multi-objective environmental selection and updating the immigrant archives. Step 4 is when the populations inherit the weights of the super network during evolution and update the weights of the super network after training for a few epochs. Finally, the optimal network is obtained through the search process.

SuperNet Update: The supernet parameter is warmed up for *m* epochs before starting the search. In the search phase, supernet training and population evolution are carried out alternately, and supernet will sample candidate individuals from the population to participate in training and update supernet parameters.

3.2 Initialization

The Multi-Populations Initialization is shown in Figure 2. Since neural architectures typically use feedforward structures, we represent the entire search space pool \mathcal{A} containing L layer blocks as a directed acyclic graph (DAG). Assuming that the number of possible operation combinations in each block is N, the total number of operations contained in the entire search space pool is N^L . In order to simplify the complexity of the problem, we divide the entire search space pool \mathcal{A} into L subsets \mathcal{A}_l and use L populations to represent these sub-search space pools, $\sum_{l=1}^{L} \mathcal{A}_l$. The total number of operations included in the simplified L-layer neural network is $N \times L$.

The initialization of multiple populations is the process of creating L populations respectively based on the number of layers L of the complete network. Each population P_i randomly generates N_i individuals and the generation of individuals corresponds to the network structure being encoded into a binary genome. The blocks corresponding to each layer in the search space we use are the same as the Cell in DARTS [Liu et al., 2018b]. DARTS uses a cell-level search space, where the cell uses four nodes to represent the sum of feature maps and edges to represent operations. The difference from DARTS is that the network structure parameters we define are $\{0, 1\}$ binary encode.

3.3 Supernet Training

Subnetwork Sampling: Given L populations $P_{1:L}$, sampling a complete subnetwork s_i from a supernet S is achieved by sample L individuals from the multipopulation $P_{1:L}$. Therefore, the sampling process of the subnetwork s_i can be defined as follows:

$$s_i = \sum_{l \in \mathcal{L}} decode\left(p_i^l\right) \tag{1}$$

Where L represents the layer number index set of the L-layer sub-network, and also represents L populations. Where p_i^l denotes the individual p_i sampled from the l_{th} population.

Weight Update: The weight $W_{\mathcal{S}}$ of the supernet \mathcal{S} is updated by training on the sampling subnetwork s_i . It is worth noting that the initial weight $W_{\mathcal{S}}(s_i)$ of subnetwork s_i is obtained by inheriting the weight $W_{\mathcal{S}}$ of supernet \mathcal{S} through weight sharing. The optimization of weights W_S can be expressed as follows:

$$W_{\mathcal{S}} = \underset{W_{\mathcal{S}}}{\operatorname{arg\,min}} \sum_{s_i \in \mathcal{S}} \mathcal{L}_{train} \left(\mathcal{M} \left(s_i, W_{\mathcal{S}}(s_i) \right) \right)$$
(2)

where $\mathcal{L}_{train}(\cdot)$ represents the cross-entropy loss function on the training set, and $\mathcal{M}(s_i, W_{\mathcal{S}}(s_i))$ denotes the model with subnetwork structure s_i and weights $W_{\mathcal{S}}(s_i)$. In the subnetwork sampling stage, we sample every possible architecture equally and jointly optimize the loss functions of these architectures, so each individual is sampled with $p_i \sim Bernoulli(0.5).$



Figure 2: **Multi-Populations Initialization.** Individual encode refers to the transformation of networks sampled from the search space into a two-dimensional matrix through encode strategy. The initialization of population P_i is the process of generating N_i network individuals. The initialization of multiple populations is the process of creating L populations based on the number of layers L in the complete network. It is important to note that the L individuals sampled from multiple populations together constitute a complete network.

3.4 Multi-Populations Alternative Evolutionary

In the evolutionary search phase, instead of using individuals in a single population to encode the global neural network, we simplify the encoding of the global network structure by multiple populations as a way to reduce the complexity of the search problem. The algorithm uses populations to represent different modules of the global network and alternatively optimizes each module. The details of the algorithm are described in Algorithm 1.

Algorithm 1 Multi-Populations Alternative Evolutionary Search Algorithm

- **Input:** SuperNet S, network layers size L, population size N, iterations size E_{evo} , archives size m, multi-objectives $\{\mathcal{F}_1, ..., \mathcal{F}_M\}$.
- 1: $P_{1:L} \leftarrow$ Initialize multiple populations with the size N.
- 2: $archives_{1:L} \leftarrow$ Initialize immigration archive via select *m* individuals from $P_{1:L}$.
- 3: Warm up the SuperNet S for E_{warm} epochs.
- 4: for $t = 1, ..., E_{evo}$ do
- 5: **for** l = 1, ..., L **do**
- 6: Training the individuals in P_t^l by using the sampled training strategy on D_{train}
- 7: $Q_t^l \leftarrow \text{Generate sub-population via genetic manipulation}$
- 8: $P_{t+1}^{l} \leftarrow \text{Select } N \text{ individuals from } P_{t}^{l} \cup Q_{t}^{l} \cup M_{t}^{l}$ through multi-objective environment selection
- 9: $\{\mathcal{F}_1, ..., \mathcal{F}_M\} \leftarrow$ Evaluate the multi-objectives of individuals in $P_t^l \cup Q_t^l \cup M_t^l$
- 10: $archive_l \leftarrow$ Select x individuals from P_{t+1}^l through multi-objective environment selection
- 11: end for
- 12: end for

Output: Architectures $S = \left\{ s_1^{(E_{\text{evo}})}, \dots, s_P^{(E_{\text{evo}})} \right\}$

Algorithm 1 presents the framework of our proposed MPAE approach. Initially, multiple populations are initialized using the proposed genetic encoding strategy (line 1), and immigration archives are initialized for each population.

Evolutionary processes commence after a warm-up period of the supernet for a few epochs and continue until a predefined termination criterion, such as reaching the maximum number of generations. Finally, individuals on the Pareto front are selected and decoded into their corresponding networks for final training.

During the evolution process, all populations undergo an alternate evolution in cycles. A multi-objective evaluation is first performed on all individuals in the population based on weight sharing. Subsequently, new offspring are generated through our designed genetic manipulation. The immigrant population is then generated using the proposed proximity selection operation. Finally representatives are selected from the existing individuals, newly generated offspring, and the immigrant population to form the next generation population and participate in subsequent evolution.

3.5 Migration Mechanism

Early research [Girish *et al.*, 2019; Zeiler and Fergus, 2014] shows that different layers of the network learn different features, the optimal structure of different layers is not necessarily the same. However, we consider that the structure of excellent networks in adjacent layers is similar, and the repeated structures commonly used in CNNs (such as repeated modules in Inception [Szegedy *et al.*, 2017], ResNet [He *et al.*, 2016] and NASNet [Zoph *et al.*, 2018]) also show this phenomenon. Therefore, we add a population migration mechanism to the algorithm, introduce excellent and potential individuals from other populations during the evolution of the current population, and make full use of the knowledge gained from the evolution of other populations.

The migration mechanism determines the number of individuals migrated according to the adjacent distance of each population. The adjacent distance of the population is the difference between the corresponding network layer serial numbers of each population. At the same time, the migrating individuals of a population are selected based on the degree of similarity between the individuals and the population. The degree of similarity between individual Gen_a in population

	CIFAR 10			CIFAR 100			
Algorithm	ACC(%)	# P(M)	GDs	ACC(%)	# P(M)	GDs	Search space
CNN-GA [Sun et al., 2020]	96.78	2.9	35	79.47	4.1	40	Global
SI-EvoNet [Zhang et al., 2021a]	97.31	1.84	0.458	79.16	0.99	0.813	Global
AE-CNN [Sun et al., 2019b]	95.3	2	27	77.6	5.4	36	Global
AE-CNN+E2EPP [Sun et al., 2019a]	94.7	4.3	7	77.98	20.9	10	Global
EPCNAS-C [Huang et al., 2023]	96.93	1.2	1.1	81.67	1.29	1.1	Global
AmoebaNet-A [Real et al., 2019]	96.66±0.06	3.1	3150	81.07	3.1	3150	Scalable
NASNet-A [Zoph <i>et al.</i> , 2018]	97.35	3.2	1800	82.19	3.2	1800	Scalable
BlockQNN-S [Zhong et al., 2020]	96.7	6.1	90	82.95	6.1	90	Scalable
PNAS [Liu <i>et al.</i> , 2018a]	96.59±0.09	3.2	225	82.37	3.2	225	Scalable
MdeNAS [Zheng et al., 2019]	97.45	3.8	0.16	82.39	3.8	0.16	Scalable
RENAS [Chen et al., 2019b]	97.12±0.02	3.5	6	-	-	-	Scalable
GDAS-NSAS [Zhang et al., 2021b]	97.25±0.08	3.5	0.4	81.98±0.05	3.5	0.4	Scalable
ENAS [Pham <i>et al.</i> , 2018]	97.11	4.6	0.5	81.09	4.6	0.5	Scalable
WPL [Benyahia <i>et al.</i> , 2019]	96.19	-	-	-	-	-	Scalable
Random (F=64) [Bender et al., 2018]	95.6	6.7	-	-	-	-	Scalable
SNAS [Xie et al., 2019]	97.15±0.02	2.8	1.5	79.91	2.8	1.5	Scalable
PC-DARTS [Xu et al., 2021]	97.43±0.07	3.6	0.1	82.89	3.6	0.1	Scalable
DARTS [Liu et al., 2018b]	97.00±0.14	3.3	1	82.46	3.3	1	Scalable
CARS [Yang et al., 2020]	97.38	3.6	0.4	82.72	3.6	0.4	Scalable
PDARTS [Chen et al., 2019a]	97.5	3.4	0.3	84.08	3.6	0.3	Scalable
NSGANetV1-A3 [Lu et al., 2020]	97.78	2.2	27	82.77	2.2	27	Scalable
TS-ENAS [Zou et al., 2023]	97.57	2.98	0.9	82.69	3.35	0.9	Hierarchical
Proxyless NAS [Cai et al., 2019]	97.92	5.7	1500	-	-	-	Hierarchical
MPAE-A	97.35	2.8	0.3	82.74	2.9	0.3	Our
MPAE-B	97.39	3.2	0.3	83.45	3.3	0.3	Our
MPAE-C	97.51	3.7	0.3	84.12	3.6	0.3	Our

Table 1: Comparison of MPAE with other NAS and OSNAS methods in terms of testing classification accuracy (ACC), number of parameters (P), and search cost (in terms of GPU Days, GDs) on CIFAR10 and CIFAR100 benchmark datasets. Symbol "–" indicates that no results have been reported on the dataset in the original paper.

 P_a and population P_b is expressed by the following equation:

$$\sin\left(\operatorname{Gen}_{a}, \operatorname{P}_{b}\right) = \frac{\operatorname{Gen}_{a} \times \sum_{i=1}^{D} \operatorname{Gen}_{b}^{i}}{D \cdot Len\left(Gen\right)}$$
(3)

where $\sum_{i=1}^{D} \operatorname{Gen}_{b}^{i}$ represents the first D best individual gene codes (the sum of each target value is the smallest) in the population P_{b} . The smaller the value of sim (Gen_a, P_b), the less similar the migration individual Gen_{a} selected in population P_{a} is to population P_{b} , with the aim of increasing the diversity of population P_{b} while ensuring individual fitness.

4 Experiment

In this section, we present the experimental results of our proposed algorithm compared to other competitors on benchmark classification datasets, CIFAR and ImageNet. The results include accuracy (%), parameter count (M), and search time (GDs). GDs refers to the number of days required to search for the specified neural network on a single GPU. Additionally, we categorize the other competitors into three categories based on the search space: Scalable, Global, and Hierarchical. The Scalable category primarily constructs networks by stacking the best-performing blocks obtained from the search. The Global category directly searches for complete networks on the dataset without the need for repeated block proxies. The Hierarchical category can be regarded as a special type of Global category, which searches each net-

work block separately, allowing network blocks with different structures to construct the complete network.

Evolution Details. To demonstrate the effectiveness of our method, we adopt the same cell-level search space as DARTS [Liu *et al.*, 2018b]. MPAE builds an L layer supernet to assist evolution and adopts alternate search to explore L different cell structures of the supernet. We set the supernet layers and populations size to 20 and the individuals size of population to 64. During the evolutionary search, the crossover and mutation rate were both set to 0.25. During the genetic manipulation, each node has a 0.25 probability of undergoing connection crossover and has a 0.25 probability of being randomly reassigned.

4.1 Image Classification on CIFAR

Search on CIFAR. We split the CIFAR-10 and CIFAR-100 train set into two parts, i.e., 25,000 images for updating network parameters and 25,000 for verification architectures. The split strategy is the same as DARTS [Liu *et al.*, 2018b]. We search for 500 epochs in total, and the parameter warmup stage lasts for the first 10% epochs (50). After that, we initialize the population, which maintains 64 different architectures and gradually evolve them using proposed MPAE. We use MPAE to update architectures after the network parameters are updated for ten epochs.

Evaluate on CIFAR. After the MPAE search phase, we retrained the searched architectures on the CIFAR-10 and

	ImageNet test accuracy (%)			Search Cost	Search	
Algorithm	Top-1	Top-5	# Params(M)	(GPU-days)	dataset	Search space
SI-EvoNAS [Zhang et al., 2021a]	75.8	92.59	4.7	0.458	CIFAR10	Global
AmoebaNet [Real et al., 2019]	75.7	92.4	6.4	3150	CIFAR10	Scalable
DARTS [Liu et al., 2018b]	73.3	91.3	4.7	4	CIFAR10	Scalable
NASNet-A [Zoph et al., 2018]	74	91.6	5.3	1800	CIFAR10	Scalable
PNAS [Liu <i>et al.</i> , 2018a]	74.2	91.9	5.1	225	CIFAR10	Scalable
MdeNAS [Zheng et al., 2019]	73.2	-	6.1	0.16	CIFAR10	Scalable
RENAS [Chen et al., 2019b]	75.7	92.6	5.36	1.5	CIFAR10	Scalable
CARS-C [Yang <i>et al.</i> , 2020]	75.2	92.5	5.1	0.4	CIFAR10	Scalable
GDAS-NSAS-C [Zhang et al., 2021b]	74.1	-	5.2	0.4	CIFAR10	Scalable
BayesNAS [Zhou et al., 2019]	73.5	91.1	3.9	0.2	CIFAR10	Scalable
SETN [Dong and Yang, 2019]	74.3	92	5.4	1.8	CIFAR10	Scalable
RandomNAS-NSAS-C [Zhang et al., 2021c]	74.5	-	5.4	0.7	CIFAR10	Scalable
SNAS(mild) [Xie <i>et al.</i> , 2019]	72.7	90.8	4.3	1.5	CIFAR10	Scalable
PDARTS [Chen et al., 2019a]	75.3	92.5	5.1	0.3	CIFAR100	Scalable
NSGANetV1-A3 [Lu et al., 2020]	76.2	93.0	5	27	CIFAR100	Scalable
TS-ENAS [Zou et al., 2023]	75.6	-	3.6	0.9	CIFAR10	Hierarchical
EPCNAS-C [Huang et al., 2023]	72.9†	91.5	3.02	1.17	ImageNet	Global
FairNAS [Zhang et al., 2021b]	75.3†	-	4.6	12	ImageNet	Global
PC-DARTS [Xu et al., 2021]	75.8†	92.7	5.3	3.8	ImageNet	Scalable
Proxyless NAS(GPU) [Cai et al., 2019]	75.1†	92.5	7.1	8.3	ImageNet	Hierarchical
MnasNet-A3 [Tan et al., 2019]	76.7†	93.3	5.2	1600	ImageNet	Hierarchical
MixNet-M [Tan and Le, 2019]	77†	93.3	5	1600	ImageNet	Hierarchical
MPAE-A	74.1	91.9	4.2	0.3	CIFAR10	Our
MPAE-B	75.1	92.5	4.8	0.3	CIFAR10	Our
MPAE-C	75.7	92.7	5.2	0.3	CIFAR10	Our

Table 2: Overall comparison of MPAE with other NAS and OSNAS methods on the ILSVRC2012 dataset. The MPAE model is a architecture searched on the CIFAR-10 dataset. The symbol "†" in Table 2 indicates that the network was directly searched on ImageNet. The term "search dataset" refers to the dataset used for the neural architecture search and evaluation process.

CIFAR-100 datasets with all the same training parameters as DARTS [Liu et al., 2018b]. As shown in Table I, on CIFAR-10 and CIFAR-100, MPAE outperforms most of its competitors in the Scalable category with similar parameters and a search cost of only 0.3 GDs. On the CIFAR-10 dataset, although NSGANetV1-A3 shows better classification accuracy than MPAE, the search cost of MPAE is much lower (only 0.3 GDs, much lower than 27 GDs). On CIFAR100, MPAE achieved a classification accuracy of 84.12%, surpassing its peer competitors in all Scalable categories considered in the experiment. Compared with peer competitors in the Global and Hierarchical categories, MPAE achieved competitive results on both CIFAR10 and CIFAR100 with lower search costs. Although the classification accuracy of Proxyless NAS is higher than MPAE-C on CIFAR10, MPAE-C has fewer parameters (3.7M < 5.7M) and lower search cost (only 0.3 GDs, much lower than 1500 GDs). On CIFAR100, MPAE surpassed all Global and Hierarchical competitors considered in the experiment in both accuracy (%) and search time (GDs).

4.2 Image Classification on ImageNet

Training Details. We evaluate the transferability of the architectures obtained from the CIFAR-10 dataset by training them on the ImageNet dataset. We use Nvidia A100 to train the models, and the batch size is 256. We train 250 epochs in total. The learning rate is 0.1 with a linear decay scheduler. Momentum is 0.9, and weight decay is 3e-5. Label smooth is also used with a smooth ratio of 0.1.

Evaluate on ImageNet. Table 2 summarizes the results of our model and other automated search networks on the ImageNet validation set. We categorize other search algorithms into two groups based on whether the search dataset is ImageNet or not. "Non-ImageNet" search dataset refers to methods that search for networks on small datasets and then transfer them to larger datasets. In comparison to other non-ImageNet datasets, the MPAE model exhibits significant advantages. Only the NSGANetV1-A3 [Lu et al., 2020] model has a slightly higher accuracy (0.5%) than MPAE, but its search cost is 90 times more than MPAE. Methods that require searching on the ImageNet dataset demand even more search time or computational resources. For example, although MixNet-M [Tan and Le, 2019] and MnasNet-A3 [Tan et al., 2019] models have higher accuracy than MPAE, they consume far more GPU-days than our approach (1600 GDs >> 0.3 GDs). While other methods using ImageNet as the search dataset have a search cost of around 10 GDs, their performance is similar to or lower than MPAE. Therefore, MPAE remains effective in this regard, and it ensures module diversity with a smaller search cost, thereby guaranteeing model performance.

5 Ablation Study

Effectiveness of Multi-Populations Alternate Evolution: Comparing the contributions of different NAS algorithms in terms of the search stage can be challenging and ambiguous due to the significant differences in the search space and train-



Figure 3: Search efficiency comparison between MPAE and other baselines in terms of (a) validation accuracy and (b) required compute time in GPU-Days. The search cost is measured on CIFAR-10 for most methods, except Block-QNN, where the CIFAR-100 dataset is used for.



Figure 4: Comparison graph of migrated individuals and offspring individuals at various stages of the MPAE evolutionary process.

ing procedures used during the search process. Therefore, we use multi-population coevolution and single-population global search as comparisons with our MPAE algorithm to demonstrate the efficiency of the search stage in MPAE. All three methods utilize the same search space and performance estimation strategy. The results shown in Figure 3(a) indicate that MPAE is capable of finding more accurate architectures compared to multi-population coevolution and singlepopulation global search. In addition to accuracy metrics, another important aspect of demonstrating NAS efficiency is the computational complexity of the methods. Since performing a theoretical analysis of the computational complexity for different NAS methods is challenging, we compare the computational time (GPU-days) spent by each method on the GPU to derive the reported architectures. Running MPAE once on the CIFAR-10 dataset takes approximately 0.3 GPU-days to complete (average of five runs). From the search cost comparison in Figure 3(b), we observe that our proposed algorithm is faster in identifying a set of architectures compared to many other methods and the obtained architecture set exhibits higher performance.

Effectiveness of Migration Mechanism: To demonstrate the effectiveness of the proposed population migration mechanism, we compare individuals sampled uniformly from four stages (100, 200, 300, 400 epochs) of the population evolution process. Figure 4 displays the results of comparing individuals migrated from other populations with individuals generated through population crossover and mutation at each stage. It can be observed that the migrated individuals exhibit higher generalization accuracy, while the offspring individuals have a wider search range. MPAE improves population convergence through migrated individuals, while it enhances population diversity through offspring individuals.

6 Conclusion

This paper proposed the multiple population alternate evolution framework that mitigated the conflict between search costs and module diversity by achieving module diversification with lower search costs. We first simplify the search space into multiple search subspaces and then use multiple populations to alternately search within these subspaces. This significantly reduces the complexity of the problem and improves search efficiency. We also introduce a population migration mechanism, which aims to accelerate the evolution process by utilizing the knowledge and experience retained by each population. Furthermore, our experiments demonstrate that the performance of migrated individuals generally surpasses that of offspring generated by the populations. Our MPAE algorithm achieves state-of-the-art results with only 0.3 GPU days of search, it demonstrates outstanding performance in both search speed and search efficiency.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grant No. 62276224, Grant No. 6230073173), in part by the Natural Science Foundation of Hunan Province, China (Grant No. 2022JJ40452)

References

- [Bender et al., 2018] Gabriel Bender, Pieter-Jan Kindermans, Barret Zoph, Vijay Vasudevan, and Quoc Le. Understanding and simplifying one-shot architecture search. In *International conference on machine learning*, pages 550–559. PMLR, 2018.
- [Benyahia et al., 2019] Yassine Benyahia, Kaicheng Yu, Kamil Bennani Smires, Martin Jaggi, Anthony C. Davison, Mathieu Salzmann, and Claudiu Musat. Overcoming multi-model forgetting. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, Proceedings of the 36th International Conference on Machine Learning, volume 97 of Proceedings of Machine Learning Research, pages 594– 603. PMLR, 09–15 Jun 2019.
- [Cai *et al.*, 2019] Han Cai, Ligeng Zhu, and Song Han. ProxylessNAS: Direct neural architecture search on target task and hardware. In *International Conference on Learning Representations*, 2019.
- [Chen *et al.*, 2019a] Xin Chen, Lingxi Xie, Jun Wu, and Qi Tian. Progressive differentiable architecture search: Bridging the depth gap between search and evaluation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1294–1303, 2019.
- [Chen *et al.*, 2019b] Yukang Chen, Gaofeng Meng, Qian Zhang, Shiming Xiang, Chang Huang, Lisen Mu, and Xinggang Wang. Renas: Reinforced evolutionary neural architecture search. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4787–4796, 2019.
- [Dong and Yang, 2019] Xuanyi Dong and Yi Yang. Oneshot neural architecture search via self-evaluated template network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3681–3690, 2019.
- [Girish *et al.*, 2019] Deeptha Girish, Vineeta Singh, and Anca L Ralescu. Unsupervised clustering based understanding of cnn. In *CVPR Workshops*, pages 9–11, 2019.
- [He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [Huang et al., 2023] Junhao Huang, Bing Xue, Yanan Sun, Mengjie Zhang, and Gary G. Yen. Particle swarm optimization for compact neural architecture search for image classification. *IEEE Transactions on Evolutionary Computation*, 27(5):1298–1312, 2023.
- [Klyuchnikov et al., 2022] Nikita Klyuchnikov, Ilya Trofimov, Ekaterina Artemova, Mikhail Salnikov, Maxim Fedorov, Alexander Filippov, and Evgeny Burnaev. Nasbench-nlp: neural architecture search benchmark for nat-

ural language processing. *IEEE Access*, 10:45736–45747, 2022.

- [Liang *et al.*, 2021] Tingting Liang, Yongtao Wang, Zhi Tang, Guosheng Hu, and Haibin Ling. Opanas: One-shot path aggregation network architecture search for object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10195– 10203, 2021.
- [Liu et al., 2018a] Chenxi Liu, Barret Zoph, Maxim Neumann, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy. Progressive neural architecture search. In Proceedings of the European conference on computer vision (ECCV), pages 19–34, 2018.
- [Liu *et al.*, 2018b] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. In *International Conference on Learning Representations*, 2018.
- [Lu et al., 2020] Zhichao Lu, Ian Whalen, Yashesh Dhebar, Kalyanmoy Deb, Erik D Goodman, Wolfgang Banzhaf, and Vishnu Naresh Boddeti. Multiobjective evolutionary design of deep convolutional neural networks for image classification. *IEEE Transactions on Evolutionary Computation*, 25(2):277–291, 2020.
- [Pham et al., 2018] Hieu Pham, Melody Guan, Barret Zoph, Quoc Le, and Jeff Dean. Efficient neural architecture search via parameters sharing. In *International conference* on machine learning, pages 4095–4104. PMLR, 2018.
- [Real et al., 2017] Esteban Real, Sherry Moore, Andrew Selle, Saurabh Saxena, Yutaka Leon Suematsu, Jie Tan, Quoc V Le, and Alexey Kurakin. Large-scale evolution of image classifiers. In *International Conference on Machine Learning*, pages 2902–2911. PMLR, 2017.
- [Real *et al.*, 2019] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. Regularized evolution for image classifier architecture search. In *Proceedings of the aaai conference on artificial intelligence*, volume 33, pages 4780–4789, 2019.
- [Sun *et al.*, 2019a] Yanan Sun, Handing Wang, Bing Xue, Yaochu Jin, Gary G Yen, and Mengjie Zhang. Surrogateassisted evolutionary deep learning using an end-to-end random forest-based performance predictor. *IEEE Transactions on Evolutionary Computation*, 24(2):350–364, 2019.
- [Sun *et al.*, 2019b] Yanan Sun, Bing Xue, Mengjie Zhang, and Gary G Yen. Completely automated cnn architecture design based on blocks. *IEEE transactions on neural networks and learning systems*, 31(4):1242–1254, 2019.
- [Sun et al., 2019c] Yanan Sun, Bing Xue, Mengjie Zhang, and Gary G Yen. Evolving deep convolutional neural networks for image classification. *IEEE Transactions on Evolutionary Computation*, 24(2):394–407, 2019.
- [Sun *et al.*, 2020] Yanan Sun, Bing Xue, Mengjie Zhang, Gary G Yen, and Jiancheng Lv. Automatically designing cnn architectures using the genetic algorithm for im-

age classification. *IEEE transactions on cybernetics*, 50(9):3840–3854, 2020.

- [Szegedy et al., 2017] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.
- [Tan and Le, 2019] Mingxing Tan and Quoc V. Le. Mixconv: Mixed depthwise convolutional kernels. CoRR, abs/1907.09595, 2019.
- [Tan et al., 2019] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. Mnasnet: Platform-aware neural architecture search for mobile. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 2820–2828, 2019.
- [Tian et al., 2020] Ye Tian, Ruchen Liu, Xingyi Zhang, Haiping Ma, Kay Chen Tan, and Yaochu Jin. A multipopulation evolutionary algorithm for solving large-scale multimodal multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 25(3):405– 418, 2020.
- [Wu et al., 2019] Bichen Wu, Xiaoliang Dai, Peizhao Zhang, Yanghan Wang, Fei Sun, Yiming Wu, Yuandong Tian, Peter Vajda, Yangqing Jia, and Kurt Keutzer. Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10734–10742, 2019.
- [Xie and Yuille, 2017] Lingxi Xie and Alan Yuille. Genetic cnn. In Proceedings of the IEEE international conference on computer vision, pages 1379–1388, 2017.
- [Xie et al., 2019] Sirui Xie, Hehui Zheng, Chunxiao Liu, and Liang Lin. SNAS: stochastic neural architecture search. In International Conference on Learning Representations, 2019.
- [Xu et al., 2021] Yuhui Xu, Lingxi Xie, Wenrui Dai, Xiaopeng Zhang, Xin Chen, Guo-Jun Qi, Hongkai Xiong, and Qi Tian. Partially-connected neural architecture search for reduced computational redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(9):2953–2970, 2021.
- [Yang et al., 2020] Zhaohui Yang, Yunhe Wang, Xinghao Chen, Boxin Shi, Chao Xu, Chunjing Xu, Qi Tian, and Chang Xu. Cars: Continuous evolution for efficient neural architecture search. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1829–1838, 2020.
- [Yu *et al.*, 2019] Kaicheng Yu, Christian Sciuto, Martin Jaggi, Claudiu Musat, and Mathieu Salzmann. Evaluating the search phase of neural architecture search. In *International Conference on Learning Representations*, 2019.
- [Zeiler and Fergus, 2014] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In

Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13, pages 818–833. Springer, 2014.

- [Zhang et al., 2020] Xinbang Zhang, Zehao Huang, Naiyan Wang, Shiming Xiang, and Chunhong Pan. You only search once: Single shot neural architecture search via direct sparse optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(9):2891–2904, 2020.
- [Zhang et al., 2021a] Haoyu Zhang, Yaochu Jin, Ran Cheng, and Kuangrong Hao. Efficient evolutionary search of attention convolutional networks via sampled training and node inheritance. *IEEE Transactions on Evolutionary Computation*, 25(2):371–385, 2021.
- [Zhang et al., 2021b] Miao Zhang, Huiqi Li, Shirui Pan, Xiaojun Chang, Chuan Zhou, Zongyuan Ge, and Steven Su. One-shot neural architecture search: Maximising diversity to overcome catastrophic forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(9):2921– 2935, 2021.
- [Zhang et al., 2021c] Miao Zhang, Huiqi Li, Shirui Pan, Xiaojun Chang, Chuan Zhou, Zongyuan Ge, and Steven Su. One-shot neural architecture search: Maximising diversity to overcome catastrophic forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(9):2921– 2935, 2021.
- [Zhang *et al.*, 2022] Haoyu Zhang, Yaochu Jin, and Kuangrong Hao. Evolutionary search for complete neural network architectures with partial weight sharing. *IEEE transactions on evolutionary computation*, 26(5):1072– 1086, 2022.
- [Zheng et al., 2019] Xiawu Zheng, Rongrong Ji, Lang Tang, Baochang Zhang, Jianzhuang Liu, and Qi Tian. Multinomial distribution learning for effective neural architecture search. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 1304–1313, 2019.
- [Zhong et al., 2020] Zhao Zhong, Zichen Yang, Boyang Deng, Junjie Yan, Wei Wu, Jing Shao, and Cheng-Lin Liu. Blockqnn: Efficient block-wise neural network architecture generation. *IEEE transactions on pattern analysis and machine intelligence*, 43(7):2314–2328, 2020.
- [Zhou et al., 2019] Hongpeng Zhou, Minghao Yang, Jun Wang, and Wei Pan. Bayesnas: A bayesian approach for neural architecture search. In *International conference on* machine learning, pages 7603–7613. PMLR, 2019.
- [Zoph and Le, 2016] Barret Zoph and Quoc Le. Neural architecture search with reinforcement learning. In *International Conference on Learning Representations*, 2016.
- [Zoph et al., 2018] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 8697–8710, 2018.
- [Zou *et al.*, 2023] Juan Zou, Shenghong Wu, Yizhang Xia, Weiwei Jiang, Zeping Wu, and Jinhua Zheng. Ts-enas:

Two-stage evolution for cell-based network architecture search. *arXiv preprint arXiv:2310.09525*, 2023.