Bi-fidelity Multi-objective Neural Architecture Search for Adversarial Robustness with Surrogate as a Helper-objective

Jia Liu[1] and **Yaochu Jin**[1,2]

[1]Department of Computer Science, University of Surrey, United Kingdom.
[2] Faculty of Technology, Bielefeld University, Germany
{jia.liu, yaochu.jin}@surrey.ac.uk

Abstract

Deep neural networks have been found easily fooled by adversarial attacks, which raises major concerns in security-sensitive contexts. Over the past years, considerable efforts have been made to improve the robustness of deep learning models. Recent research has investigated the adversarial robustness of neural networks from the architectural point of view and produced encouraging results. However, the cost of computation in the search for architectures is high, which becomes even worse because adversarial training process is particularly time-consuming. To address the above challenge, this paper proposes a surrogate-assisted approach to the search of robust architectures effectively and efficiently. More specifically, we leverage low-fidelity evaluations to predict the performance of architectures and introduce an additional "helper-objective", the value of which is the output of a surrogate model that is trained based on high-fidelity evaluations. Comprehensive experiments confirm the effectiveness of our approach. The discovered architectures show competitive performance on CIFAR-10, and outperform most baselines on CIFAR-100 and SVHN datasets.

1 Introduction

Deep neural networks (DNNs) have been shown to be vulnerable to adversarial examples that are intentionally crafted with imperceptible perturbations [Szegedy *et al.*, 2013]. Much effort has been made to tackle the threat of adversarial examples, such as adversarial training [Goodfellow *et al.*, 2015], defensive distillation [Carlini and Wagner, 2016], and adversary detection [Metzen *et al.*, 2017].

Despite the considerable effort on defense strategies, a majority of researchers carried out experiments based on one or two specific manually designed convolutional neural networks (CNNs). Recently, neural architecture search (NAS) has attracted increased attention and achieved outstanding performance on a variety of tasks. Early NAS algorithms [Zoph and Le, 2016] suffer from an extremely heavy computational burden since evaluating the performance of each candidate architecture requires to train the network from scratch and then test it on a validation dataset. To reduce the search cost, researchers built proxy networks with fewer layers or channels [Real *et al.*, 2019; Wu *et al.*, 2019], and trained them to solve proxy tasks of smaller scales [Cai *et al.*, 2018; Wu *et al.*, 2019; Liu and Jin, 2021]. However, the architectures obtained from proxy tasks do not perform well on the target task. Parameter sharing [Pham *et al.*, 2018; Cai *et al.*, 2019] and predictor-based evaluators [Liu *et al.*, 2018; Sun *et al.*, 2019] are two efficient strategies to estimate the performance of architectures.

Despite the remarkable progress, existing NAS methods mainly focus on improving classification accuracy and are limited by intensive computation and memory costs. Only a few studies have attempted to understand adversarial robustness from an architectural perspective.

In this work, we utilize the predicted value of a surrogate model, leveraging the knowledge of high-fidelity fitness evaluations, as a "helper-objective" to assist a multi-objective evolutionary optimization process. The proposed algorithm, named **M**ulti-**O**bjective **R**obust **A**rchitecture **S**earch based on a **S**urrogate as a **H**elper-objective (MORAS-SH), is applied to search for robust architectures effectively and efficiently. Specifically, the main contributions of this work are as follows:

- We predict the performance of candidate architectures by leveraging a combination of the parameter sharing evaluation and predictor-based evaluator to accelerate the search process.
- We adopt the concept of multi-objectivization [Knowles *et al.*, 2001] and employ an online surrogate model that predicts the high-fidelity fitness as an additional objective. This is the first attempt to use the "helper-objective" in NAS, which is the core novelty of this work.
- Experiments on benchmark datasets demonstrate that the proposed MORAS-SH method efficiently finds robust architectures with comparable classification accuracy.

The remainder of this paper is organized as follows. The next section will briefly describe the related work. In Section 3, we elaborate on the proposed approach to employing surrogate as a helper-objective in multi-objective architecture search for adversarial robustness. Experimental settings and

implementation details are presented in Section 4, followed by descriptions of the experimental results and discussion in Section 5. Finally, we summarize our findings along with future work.

2 Related Work

2.1 Adversarial Attack and Defense

Deep learning models can be misled by adversarial attacks, such as fast gradient sign method (FGSM) [Goodfellow *et al.*, 2015], basic iterative method [Kurakin *et al.*, 2016], and C&W attack [Carlini and Wagner, 2017]. One of the strongest adversarial attacks, PGD [Madry *et al.*, 2018], which combines randomized initialization with multi-step attacks, can be expressed as follows:

$$\mathbf{X}_0 = \mathbf{X} + \mathcal{U}(-\epsilon, \epsilon), \quad (1)$$

$$\mathbf{X}_{n+1} = \mathbf{X}_{,\epsilon}\{\mathbf{X}_n + \alpha \cdot \text{sign}(\nabla_{\mathbf{X}_n^*} J(\mathbf{X}_n, y))\} \quad (2)$$

where X denotes the adversarial examples, X denotes the original examples, $\mathcal{U}$ refers to a uniform distribution, ϵ is a hyper-parameter that controls the magnitude of the disturbance, α represents the step size, y denotes the true label, and $\mathbf{X}_{,\epsilon}(B)$ denotes the projection to $B(\mathbf{X}, \epsilon)$.

Extensive counter-measures have been designed to improve the robustness of deep learning models. The most effective and popular defense technique is adversarial training [Goodfellow *et al.*, 2015; Madry *et al.*, 2018], which improves the robustness of a network by training it together with adversarial examples. Adversarial training works by minimizing the weighted training loss on clean and adversarial examples. In this work, we employ PGD adversarial training (PGD-AT) to train the supernet and the architectures that are going to be evaluated for final evaluation. The PGD-AT process can be mathematically expressed as:

$$\min_{\theta} \mathsf{E}_{(\mathbf{X},y)} \quad {}_D J(PGD(\mathbf{X}, \quad), y) \quad (3)$$

where is the threat model.

2.2 Neural Architecture Evaluators

Parameter sharing and predictor-based evaluators are two commonly used techniques to efficiently evaluate architectures without training each candidate architecture from scratch.

Parameter sharing [Pham *et al.*, 2018], also known as weight sharing, is the process of building and training a super-large network within a given search space, and then the subnet directly shares the parameters from supernet. This over-parameterized supernet will contain all possible subnets. Therefore, the evaluation of subnets greatly reduces the time to evaluate candidate architectures because they share the parameters of the supernet without training them from scratch. Sample-based single-path training is a common method for training the supernet, which is trained by uniform sampling or fair multipath sampling and optimizing single paths. After training, the supernet can act as a performance estimator for different paths. When choosing a path, it can be carried out through various search strategies, such as evolutionary algorithms or reinforcement learning.

The most popular predictor in NAS is the surrogate-based predictor [Sun *et al.*, 2019] based on supervised learning. To obtain the data for training the predictor, it is necessary to train a large number of architectures initially, which is prohibitively time-consuming. The predictor then takes the architecture descriptions as inputs, and outputs the predicted performance scores. Utilizing a good predictor, promising architectures can be selected to be evaluated by the expensive evaluator. The query time is short, which allows large amount of predictions to be made during NAS.

In summary, both one-shot evaluators and predictor-based evaluators can accelerate the NAS process. However, how to combine them effectively and further reduce the computational time remains a challenging topic [Liu *et al.*, 2022].

2.3 Evolutionary Multi-objective Neural Architecture Search

Differentiable NAS, reinforcement learning and Bayesian optimization based NAS methods usually transform multi-objective NAS into a single-objective one using scalarization or an additional constraint. However, scalarization approaches were shown to be not as efficient as Pareto approaches. Multi-objective EAs are popular in solving multi-objective problems and have been shown to be successful in finding a set of Pareto optimal neural architectures in NAS [Zhu and Jin, 2020; Yang *et al.*, 2020; Hu *et al.*, 2021; Lu *et al.*, 2020].

Despite remarkable progress, the research on multi-objective NAS for the resilience of architectures against adversarial attacks [Vargas and Kotyan, 2019; Guo *et al.*, 2020; Yue *et al.*, 2020; Xie *et al.*, 2021] has been sporadic. In our previous work, we introduced MORAS [Liu and Jin, 2021] to search for architectures that are less sensitive to various adversarial attacks. However, the search process is time-consuming because each architecture in the population must be trained to obtain the fitness values. This work, by contrast, combines supernet training with surrogates to assist the evaluation process, thereby further improving the search efficiency.

3 The Proposed Approach

In this section, we develop a multi-objective architecture search for adversarial robustness with a surrogate as a "helper-objective", namely MORAS-SH. As shown in Figure 1, MORAS-SH consists of three parts, i.e., architecture evolution, architecture evaluation, and architecture training.

We employ the elitist non-dominated sorting genetic algorithm (NSGA-II) [Deb *et al.*, 2002] as the baseline for architecture evolution. To efficiently evaluate the architectures, we estimate the performance of architectures on both clean images and adversarial examples by leveraging low-fidelity fitness evaluations. To guide the search for good solutions and help maintain diversity in the population, we train a surrogate by leveraging high-fidelity evaluations to predict the performance of the candidate architectures and the predicted value is used as a "helper-objective". The low-fidelity and high-fidelity fitness values are obtained by inheriting weights W from a pre-trained supernet $\mathcal{N}$ on partial and full validation sets, respectively.

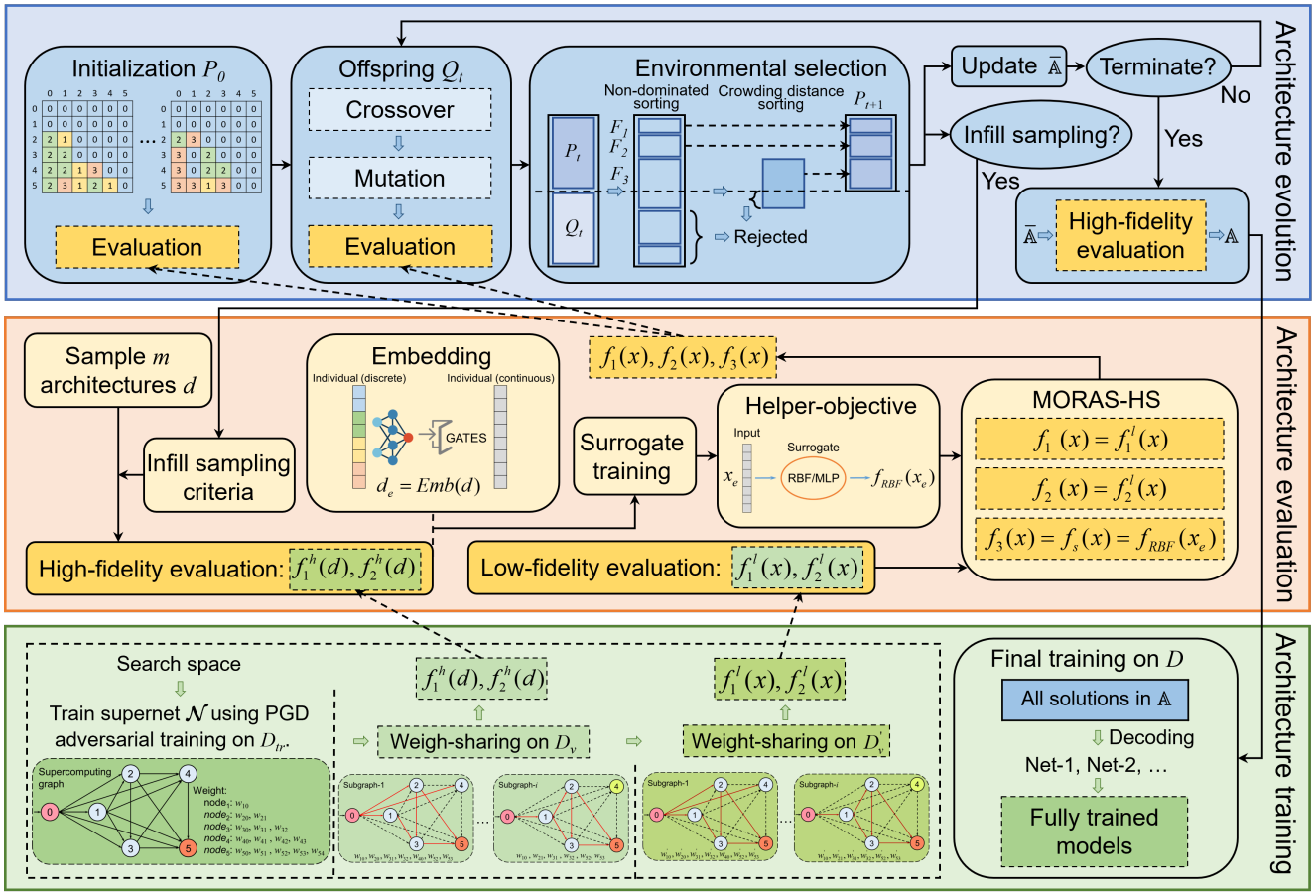

Figure 1: Overall Framework.

3.1 Search Space and Embedding

The search space we use is the same as the stage-wise search space in [Ning *et al.*, 2020a]. We use adjacency matrix encoding, which is the most common type of encodings used in current NAS research.

A surrogate model s, usually constructed by an MLP or RBF, takes a neural architecture as input and outputs a predicted score. Following [Ning *et al.*, 2020a], this work takes a graph-based neural architecture encoder called GATES [Ning *et al.*, 2020b] that maps a neural architecture into a continuous embedding space, and then concatenate the embeddings of the four stage-wise block topologies as the architecture embedding.

3.2 Multi-objectivization

Multi-objectivization [Knowles *et al.*, 2001] was intended to decompose a single-objective optimization problem into sub-objectives to reduce local optima. The concept of multi-objectivezation is developed by simultaneously optimizing the primary objective and some helper-objectives [Jensen, 2004]. Some research [Huang *et al.*, 2020] also adds a helper-objective to multi-objective optimization problems to further enhance the performance. With the help of multi-objectivization, the optimizer can perform better than focusing on the primary objectives only because the helper-objective helps maintain diversity, and guides the search away from local optima.

In this work, we include the predicted score of a surrogate as an helper-objective, where the primary objectives are the accuracy and robustness of networks. Although multi-objectivization has been used in many studies to solve difficult optimization problems, it is the first attempt to employ it in NAS and it is also novel to utilize the predicted values of a surrogate model as a helper-objective.

Both the evaluations using the low-fidelity fitness functions and surrogate models are computationally cheap yet correlated with the high-fidelity fitness function. However, neither of them is accurate enough to find a satisfactory solution to a bi-fidelity optimization problem. Moreover, the estimated performance according to the low-fidelity evaluation may be inconsistent with the one predicted by a surrogate. Hence, we use the predicted scores obtained from a surrogate as an additional objective to assist the evolutionary process with low-fidelity evaluation. We formulate NAS as the following three-objective minimization problem:

$$\min : F(x) = \{f_1, f_2, f_3\} \tag{4}$$

$$f_1(x) = f_1^l(x) = 1 - \left(\frac{1}{N}\sum \mathrm{I}(\hat{y} == y)\right) \tag{5}$$

$$f_2(x) = f_2^l(x) = 1 - (\frac{1}{N}\sum \mathbb{I}(\hat{y}_{adv} == y)) \quad (6)$$

$$f_3(x) = f_s(x) \quad (7)$$

where $f_1(x)$, $f_2(x)$ are the primary objectives, $\{f_1^l(x), f_2^l(x)\}$ denote low-fidelity fitness evaluations calculated by the error rate on the partial validation set. The candidate architectures inherit parameters from the supernet directly, so the computational cost is less expensive. $f_3(x)$ represents the "helper-objective", which equals the predicted score $f_s(x)$ of the surrogate model.

The surrogate model is trained to approximate high-fidelity fitness using data $\mathcal{S}$. Initially, we sample m solutions using the Latin hypercube sampling [Stein, 1987] and calculate their fitness using the high-fidelity evaluation, which is calculated by the error rate on the entire validation set. The inputs of the surrogate model are the values of architectures after embedding by using GATES [Ning *et al.*, 2020b]. The approximation error of the surrogate model is inevitable. Therefore, infilling samples from the current population will be added to $\mathcal{S}$ after evolving G generations. As suggested in [Wang *et al.*, 2020], we select promising and uncertain solutions as infilling samples.

3.3 Overall Framework

The MORAS-SH workflow is composed of the following three steps:

1. Supernet Training. In a predefined architecture search space (Sec. 3.1), we adversarially train a supernet $\mathcal{N}$ by using a 7-step PGD attack (PGD-7) (Sec. 2.1) on training data D_{tr}.
2. Architecture Evolution. We randomly initialize a population P_0 with n individuals (candidate topologies). For each individual, the low-fidelity evaluation is used to estimate its fitness values and the predicted score obtained from a surrogate model is used as a "helper-objective" (Sec. 3.2). The individuals in the population are gradually updated according to NSGA-II during the architecture optimization step. Concretely, we employ simulated binary crossover (SBX) and polynomial mutation (PM) [Deb *et al.*, 1995] to generate offspring. This process repeats G iterations and then k individuals from the current population are selected according to the infill criterion. The surrogate model will be updated using $\mathcal{S}$.
3. Final Training. Since we evaluate the candidate architectures during the search process by using low-fidelity evaluation with the surrogate as a helper-objective, the evaluations are of low precision. We consider this process as a pre-screening criterion. After the computation budget is exhausted, we evaluate all the non-dominant solutions in $\hat{\mathbb{A}}$ from the pre-screening criterion on the complete validation set with high fidelity to conduct secondary screening and then filter out the non-dominated solutions $\mathbb{A}$ for final adversarial training from scratch on fully training data D using PGD-AT.

4 Experimental Settings

4.1 Datasets

Three widely-studied datasets are involved in the experiments, CIFAR-10 [Krizhevsky *et al.*, 2010], CIFAR-100 [Krizhevsky *et al.*, 2009] and Street View House Numbers (SVHN) [Netzer *et al.*, 2011]. We conduct a robust architecture search on CIFAR-10, and evaluate the discovered architectures on the CIFAR-10, CIFAR-100 and SVHN datasets.

4.2 Peer Competitors

The first group of baselines includes MobileNet-V2 [Sandler *et al.*, 2018], VGG-16 [Simonyan and Zisserman, 2014], and ResNet-18 [He *et al.*, 2016], which are manually designed by human experts. The second group represents NAS-based approaches in a search space that is similar to ours, including RobNet-Free [Guo *et al.*, 2020], MSRobNet-1560 [Ning *et al.*, 2020a] and MSRobNet-1560-P [Ning *et al.*, 2020a]. The third group of competitors are conducted for ablation study. The main components of MORAS-SH include high-fidelity evaluation, low-fidelity evaluation, and surrogate modeling, which are actually part of the pre-screening of robust architectures. We conduct experiments with each components, which are termed MORAS-H, MORAS-L, MORAS-S.

4.3 Implementation Details

We used NVIDIA Titan RTX GPUs and implemented the experiments using PyTorch. PGD-7 under ℓ_1 norm with $\epsilon = 8/255$ and step size $\eta = 2/255$ is used for adversarial training.

We train the supernet with initial channel number of 44 for 400 epochs. We use an SGD optimizer with a batch size of 64, a weight decay of 1e-4, and a momentum of 0.9. The learning rate is initially set to 0.05 and decayed to 0 following a cosine schedule.

During the search process, we employ RBF and MLP as the surrogate separately. After GATES embedding, a 128-dimensional vector is fed into the surrogate. We sample 200 architectures to train an initial surrogate.

To limit the computational overhead, we set the maximum search time as three days. The portion of low-fidelity evaluation data is set to 0.2 according to [Zhou *et al.*, 2021]. To further alleviate the computational burden, we use the FGSM attack as an efficiency proxy [Ning *et al.*, 2020a] of the PGD-7 since NAS does not necessarily require accurate performance and the evaluation could be accelerated by roughly $8\times$. We use a population size of 100 and update the surrogate every 20 iterations. Ten samples will be infilled to the set $\mathcal{S}$. The probabilities for crossover and mutation are set to 0.9 and 0.02, respectively.

For better performance, we augmented the initial channels of the architectures for the final training to 55. For the final comparison on CIFAR-10, CIFAR-100 and SVHN, we adversarially train the architectures for 110 epochs on CIFAR-10/CIFAR-100 and 50 epochs on SVHN, using PGD-7 attacks with $\epsilon = 8/255$ and step size $\eta = 2/255$, and other settings are also kept the same.

To evaluate the adversarial robustness of the trained models, we apply the FGSM [Goodfellow *et al.*, 2015] with

$\epsilon = 8/255$ and PGD [Madry *et al.*, 2018] with different step numbers.

5 Experimental Results

5.1 Performance of MORAS-SH on CIFAR-10

From the set of non-dominated solutions returned after the evolution, we obtained 91 and 43 architectures by using the proposed MORAS-SH with a surrogate of RBF and MLP, respectively. After the second screening by high-fidelity evaluation, the number of non-dominated solutions both reduce to eight. We then fully train the 16 architectures and choose three architectures for each method based on their trade-offs.

Table 1 compares the performances of the architectures under various adversarial attacks. The architectures discovered by our method are referred to as MORAS-SHNets, where MORAS-SHNet-M and MORAS-SHNet-R represent the architectures obtained by MORAS-SH with an MLP and RBF as the surrogate, respectively. The results of RobNets-Free and MORobNet series are extracted from [Guo *et al.*, 2020] and [Ning *et al.*, 2020a], respectively.

As can be seen from Table 1, our MORAS-SHNet-R1 achieves 86% accuracy on clean data sets, outpacing all competitors. Under the FGSM attack, our MORAS-SHNet-M2 achieves an accuracy of 60.1%, which is also the best result among the competitors. Under the PGD-7 attack, MORAS-SHNet-R2 achieves 56.2% accuracy, the same as the MSRobNet-1560, which is also the highest. Under the PGD-20 and PGD-100 attacks, the best results are achieved by MSRobNet-1560, and our MORAS-SHNet-R2 and MORAS-SHNet-R3 are the second best.

We can see that the architectures discovered by MORAS-SHNets significantly outperform the manually designed CNNs under FGSM and strong adversarial attacks (PGD-7/10/100). With a similar number of parameters, MORAS-SHNets outperform other NAS-based peer competitors on clean samples and the samples with FGSM and PGD-7 attack. Moreover, the computational cost of MORAS-SH is smaller than the MSRobNet series since only one supernet is trained instead of eight. It illustrates that our approach can effectively and efficiently search for architectures with adversarial robustness.

5.2 Transferability to CIFAR-100 and SVHN

In line with the practice adopted in most previous NAS methods [Ning *et al.*, 2020a], we evaluate the transferability of the obtained architectures by inheriting the topology optimized for one dataset with weights retrained for a new dataset. We train MORAS-SHNets on CIFAR-100 and SVHN, and present the comparative results in Table 2.

In general, our models are consistently more robust than manually designed networks on both CIFAR-100 and SVHN. As shown in Table 2, on SVHN, our MORAS-SHNet-M3 outperforms all peer competitors in all cases, the results under different adversarial attacks are much better than the peer competitors. MORAS-SHNet-R1 outperforms others on clean CIFAR-100. However, the overall performance of our network is not as good as MSRobNet. Therefore, the transferability of this method needs to be improved. In the future, we will investigate how to enhance the transferability of the network.

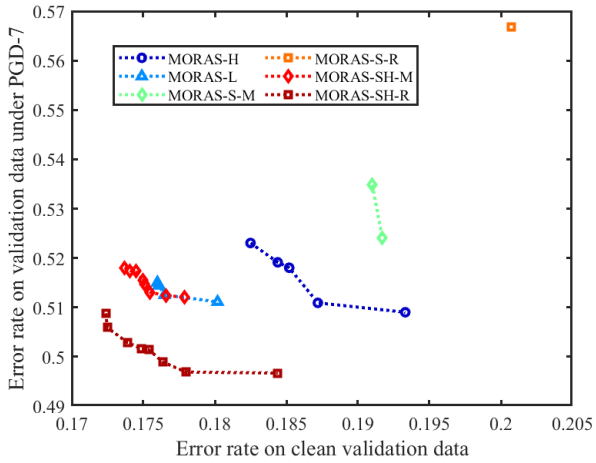

Figure 2: Pareto fronts obtained by comparative experiments. The parameters are inherited from the supernet.

5.3 Ablation Study

This section aims to disentangle the individual contribution of each main component in the proposed method. In the ablation experiment, we assume that the fitness values obtained by the parameters of the architectures inherited directly from the supernet after a complete validation set test are relatively accurate, that is, we use high-fidelity evaluation to measure the performance of the algorithms under comparison in the pre-screening process. To be fair, all experiments are terminated for three days.

Over a three-day evolutionary process, we obtained the non-dominated solutions of each experiment for pre-screening. The method we propose, with the surrogate model as a helper-objective, has a large number of solutions obtained in the pre-screening. We draw the Pareto frontier obtained after a high-fidelity evaluation of the predicted non-dominated solution obtained after three days of running each experiment on Fig. 2.

As can be seen from Fig. 2, the solutions obtained by MORAS-H, MORAS-L, MORAS-S-M and MORAS-S-R are dominated by the solutions obtained by the methods we propose. The solutions obtained by MORAS-L is comparable to our method, and in order to prove the superiority of our method, we further trained them from scratch in a complete adversarial training on CIFAR-10. We show the performance of each network after training in Figure 3. As shown in Figure 3, the architectures that MORAS-L searched for are dominated by most of the architectures we searched. This further validates the effectiveness of our surrogate model as a helper-objective.

Here we list why separately considering different parts fails to obtain promising Pareto fronts. Evaluating each candidate architecture through high-fidelity evaluations at each iteration is prohibitively expensive. With a given budget, MORAS-H can iterate only a few generations, leading to MORAS-H's

Table 1: Comparison with peer competitors under various adversarial attacks on CIFAR-10.

	Architecture	Clean (%)	FGSM (%)	PGD-7 (%)	PGD-20 (%)	PGD-100 (%)	#Para (M)	FLOPS (M)
Manually designed networks	MobileNet-V2	77.0	53.0	50.1	48.0	47.8	2.30	182
	VGG-16	79.9	53.7	50.4	48.1	47.9	14.73	626
	ResNet-18	83.9	57.9	54.5	51.9	51.5	11.17	1110
NAS-based methods	RobNet-Free	82.8	58.4	55.1	52.7	52.6	5.49	1560
	MSRobNet-1560	84.8	**60.0**	**56.2**	**53.4**	**52.9**	5.30	1588
	MSRobNet-1560-P	**85.2**	59.4	55.2	51.9	51.5	4.88	1565
Ours	MORAS-SHNet-M1	85.8	59.4	55.5	52.5	52.1	5.22	1634
	MORAS-SHNet-M2	85.4	**60.1**	55.8	52.9	52.4	5.05	1606
	MORAS-SHNet-M3	85.5	59.6	55.6	52.8	52.5	5.20	1661
	MORAS-SHNet-R1	**86.0**	59.9	55.4	52.1	51.6	5.60	1525
	MORAS-SHNet-R2	85.6	59.9	**56.2**	**53.1**	52.6	5.42	1471
	MORAS-SHNet-R3	85.1	59.9	55.8	53.0	**52.7**	5.41	1484

Table 2: Comparison with peer competitors under various adversarial attacks on CIFA-100 and SVHN.

	CIFAR-100					SVHN				
Architecture	Clean (%)	FGSM (%)	PGD-7 (%)	PGD-20 (%)	PGD-100 (%)	Clean (%)	FGSM (%)	PGD-7 (%)	PGD-20 (%)	PGD-100 (%)
MobileNet-V2	48.2	28.1	27.3	26.3	26.2	93.9	73.0	61.9	55.7	53.9
VGG-16	51.5	29.1	27.1	25.8	25.8	92.3	66.6	55.0	47.4	45.1
ResNet-18	59.2	33.8	31.6	29.9	29.7	92.3	73.5	57.4	51.2	48.8
RobNet-Free	-	-	-	-	23.9	94.2	84.0	**66.1**	**59.7**	**56.9**
MSRobNet-1560	60.8	**35.1**	**33.2**	**31.7**	**31.5**	**95.0**	77.5	64.0	57.0	54.2
MSRobNet-2000	**61.6**	34.8	32.9	31.6	**31.5**	94.9	**84.8**	65.3	58.8	55.1
MORAS-SHNet-M1	61.4	32.9	30.5	28.6	28.4	94.8	86.7	78.4	66.0	61.2
MORAS-SHNet-M2	61.2	**34.1**	30.9	29.1	28.8	94.4	84.3	65.3	58.6	55.6
MORAS-SHNet-M3	61.5	33.9	32.6	29.5	29.3	**95.8**	**90.6**	**85.7**	**73.7**	**66.3**
MORAS-SHNet-R1	**61.8**	**34.1**	30.8	28.6	28.2	94.9	85.4	64.1	57.8	54.9
MORAS-SHNet-R2	61.4	33.0	30.6	28.9	28.5	94.3	83.9	63.8	58.1	55.4
MORAS-SHNet-R3	61.4	33.0	**33.1**	**31.3**	**31.2**	94.7	77.3	61.4	55.1	52.8

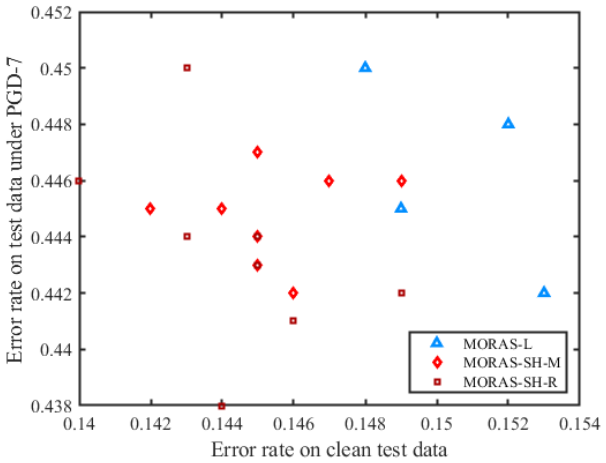

Figure 3: The performance of architectures obtained by comparative experiments after adversarial training from scratch.

inability to find better architectures in a vast search space. If using low-fidelity evaluation merely during the search process, MORAS-L can search for more generations in a limited time budget. However, the diversity of solutions is poor, and the number of non-dominated solutions obtained after the search is also small. If using the surrogate model merely in the search process, the search will be misled due to the inaccurate prediction results of the previous surrogate model. Even if the surrogate model is gradually updated with the generation increases, the range of predicted values of the updated surrogate model also changes. This means that if the surrogate model becomes more accurate, but its predictions are not as good as the previous generation of individuals, such excellent individuals will also be eliminated. In other work [Ning *et al.*, 2020b; Ning *et al.*, 2020a], researchers have trained a surrogate model with the relative ranking rather than the absolute performance values, which does not arise from the above problems. In future studies, we will introduce relative ranking into multi-objective evolutionary searches.

5.4 Discussion

To search for robust network architectures at a limited computational cost, we combined weight sharing and a surrogate-assisted approach. By using the surrogate model as an additional objective, we found that our approach can efficiently and effectively search for robust architectures compared to peer competitors on the CIFAR-10 dataset. And the network architecture is also transferable, especially on the SVHN dataset.

In terms of computational cost, we only pre-train one supernet at a time, and then inherit the parameters of the supernet when evaluating the candidate architecture, which greatly reduces the time required to evaluate the performance of the network. Moreover, we used a combination of low-fidelity evaluations and surrogate models to further speed up the search efficiency. The performance of the network obtained by our method is comparable to that of MSRobNets, but the computational cost is much less, since MSRobNets need to train eight supernets. It demonstrates that our approach can efficiently search for robust networks.

The time complexity of NSGA-II per generation is $O(MN^2)$. The number of objectives M is two if we merely consider the primary objectives. The computational complex-

ity will increase to $O(4N^2)$ if we consider accuracy and adversarial robustness predicted by the surrogate model separately. In this work, we employ a weighted sum of the clean and adversarial error rates as a label for the architecture to simplify the optimization problem and reduce the difficulty of training the surrogate model. That is, the predicted score and helper-objective are only one dimension instead of two. The weights are both set to 0.5 for the sake of simplicity. Therefore, the time complexity is $O(3N^2)$.

6 Conclusion

We employ an MOEA-based NAS approach to search for architectures that are robust to adversarial attacks. To make the procedure efficient, we propose a multi-objective architecture search for adversarial robustness with the assistance of a surrogate as a "helper-objective", namely, MORAS-SH. During evolution, MORAS-SH maximally utilizes the learned knowledge from both low- and high-fidelity fitness. Experiments results on benchmark datasets demonstrate that the proposed MORAS-SH can efficiently provide several architectures on the Pareto front. The searched models are also superior to peer competitors in terms of robustness and accuracy.

Most research on NAS for robust architectures focuses on network architectures that perform well on both clean and adversarial examples but ignoring those that perform well on clean data sets but are sensitive to attacks. Few researchers have studied what kind of network topology or parameters cause this phenomenon. It is an interesting topic to study networks that perform well on clean data but are sensitive to attacks, and it will help to further understand the intrinsic nature of neural networks. Early detection of structural factors that make networks sensitive can accelerate the discovery and design of more robust networks.

References

[Cai *et al.*, 2018] Han Cai, Tianyao Chen, Weinan Zhang, Yong Yu, and Jun Wang. Efficient architecture search by network transformation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[Cai *et al.*, 2019] Han Cai, Chuang Gan, and Song Han. Once for all: Train one network and specialize it for efficient deployment. *arXiv preprint arXiv:1908.09791*, 2019.

[Carlini and Wagner, 2016] Nicholas Carlini and David Wagner. Defensive distillation is not robust to adversarial examples. *arXiv preprint arXiv:1607.04311*, 2016.

[Carlini and Wagner, 2017] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017.

[Deb *et al.*, 1995] Kalyanmoy Deb, Ram Bhushan Agrawal, et al. Simulated binary crossover for continuous search space. *Complex systems*, 9(2):115–148, 1995.

[Deb *et al.*, 2002] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[Goodfellow *et al.*, 2015] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

[Guo *et al.*, 2020] Minghao Guo, Yuzhe Yang, Rui Xu, Ziwei Liu, and Dahua Lin. When NAS meets robustness: In search of robust architectures against adversarial attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 631–640, 2020.

[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.

[Hu *et al.*, 2021] Shengran Hu, Ran Cheng, Cheng He, Zhichao Lu, Jing Wang, and Miao Zhang. Accelerating multi-objective neural architecture search by random-weight evaluation. *Complex & Intelligent Systems*, pages 1–10, 2021.

[Huang *et al.*, 2020] Yuanjun Huang, Yaochu Jin, and Kuangrong Hao. Decision-making and multi-objectivization for cost sensitive robust optimization over time. *Knowledge-Based Systems*, 199:105857, 2020.

[Jensen, 2004] Mikkel T Jensen. Helper-objectives: Using multi-objective evolutionary algorithms for single-objective optimisation. *Journal of Mathematical Modelling and Algorithms*, 3(4):323–347, 2004.

[Knowles *et al.*, 2001] Joshua D Knowles, Richard A Watson, and David W Corne. Reducing local optima in single-objective problems by multi-objectivization. In *International conference on evolutionary multi-criterion optimization*, pages 269–283. Springer, 2001.

[Krizhevsky *et al.*, 2009] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[Krizhevsky *et al.*, 2010] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). *URL http://www. cs. toronto. edu/kriz/cifar. html*, 8, 2010.

[Kurakin *et al.*, 2016] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

[Liu and Jin, 2021] Jia Liu and Yaochu Jin. Multi-objective search of robust neural architectures against multiple types of adversarial attacks. *Neurocomputing*, 4 2021.

[Liu *et al.*, 2018] 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.*, 2022] Shiqing Liu, Haoyu Zhang, and Yaochu Jin. A survey on surrogate-assisted efficient neural architecture search. *arXiv:2206.01520*, 2022.

[Lu *et al.*, 2020] Zhichao Lu, Kalyanmoy Deb, Erik Goodman, Wolfgang Banzhaf, and Vishnu Naresh Boddeti. Nsganetv2: Evolutionary multi-objective surrogate-assisted neural architecture search. In *European Conference on Computer Vision*, pages 35–51. Springer, 2020.

[Madry *et al.*, 2018] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[Metzen *et al.*, 2017] Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. On detecting adversarial perturbations. *arXiv preprint arXiv:1702.04267*, 2017.

[Netzer *et al.*, 2011] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[Ning *et al.*, 2020a] Xuefei Ning, Junbo Zhao, Wenshuo Li, Tianchen Zhao, Yin Zheng, Huazhong Yang, and Yu Wang. Discovering robust convolutional architecture at targeted capacity: A multi-shot approach. *arXiv preprint arXiv:2012.11835*, 2020.

[Ning *et al.*, 2020b] Xuefei Ning, Yin Zheng, Tianchen Zhao, Yu Wang, and Huazhong Yang. A generic graph-based neural architecture encoding scheme for predictor-based nas. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 189–204, Cham, 2020. Springer International Publishing.

[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.*, 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.

[Sandler *et al.*, 2018] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

[Simonyan and Zisserman, 2014] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[Stein, 1987] Michael Stein. Large sample properties of simulations using latin hypercube sampling. *Technometrics*, 29(2):143–151, 1987.

[Sun *et al.*, 2019] Yanan Sun, Handing Wang, Bing Xue, Yaochu Jin, Gary G Yen, and Mengjie Zhang. Surrogate-assisted evolutionary deep learning using an end-to-end random forest-based performance predictor. *IEEE Transactions on Evolutionary Computation*, 24(2):350–364, 2019.

[Szegedy *et al.*, 2013] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[Vargas and Kotyan, 2019] Danilo Vasconcellos Vargas and Shashank Kotyan. Evolving robust neural architectures to defend from adversarial attacks. *arXiv preprint arXiv:1906.11667*, 2019.

[Wang *et al.*, 2020] Handing Wang, Yaochu Jin, Cuie Yang, and Licheng Jiao. Transfer stacking from low-to high-fidelity: A surrogate-assisted bi-fidelity evolutionary algorithm. *Applied Soft Computing*, 92:106276, 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 *et al.*, 2021] Guoyang Xie, Jinbao Wang, Guo Yu, Feng Zheng, and Yaochu Jin. Tiny adversarial mulit-objective oneshot neural architecture search. *arXiv preprint arXiv:2103.00363*, 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.

[Yue *et al.*, 2020] Zhixiong Yue, Baijiong Lin, Xiaonan Huang, and Yu Zhang. Effective, efficient and robust neural architecture search. *arXiv preprint arXiv:2011.09820*, 2020.

[Zhou *et al.*, 2021] Qi Zhou, Jinhong Wu, Tao Xue, and Peng Jin. A two-stage adaptive multi-fidelity surrogate model-assisted multi-objective genetic algorithm for computationally expensive problems. *Engineering with computers*, 37(1):623–639, 2021.

[Zhu and Jin, 2020] Hangyu Zhu and Yaochu Jin. Multi-objective evolutionary federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 31(4):1310–1322, 2020.

[Zoph and Le, 2016] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.