Supplementary Materials

Generalizable Lightweight Proxy for Robust NAS against Diverse Perturbations

A Experimental Setting

Search space Based on the cell-based neural architecture search space [46], we regard the whole network as the composition of repeated cells. Thus, we search for the optimal cell architectures and stack them repeatedly to construct the entire network. In the cell-based search phase, each cell can be represented as a directed acyclic graph (DAG), which has N nodes that represent the feature maps $z_j (j = 1, \cdots, N)$ and each edge between arbitrary node i and node j represents an operation $o_{i,j}$ chosen from the operation pool, where $o_{i,j} \in \mathcal{O} = \{o_k, k = 1, 2, \cdots, n\}$. Each feature map z_j is obtained from all of its predecessors as follows:

$$x_j = \sum_{i<j} o_{i,j}(x_i) \tag{11}$$

In this work, we utilize NAS-Bench-201 [12], and DARTS [26] search space, where different operation pools are used, which are $\mathcal{O} = \{1 \times 1 \text{ conv.}, 3 \times 3 \text{ conv.}, 3 \times 3 \text{ avg. pooling, skip, zero}\}$, and $\mathcal{O} = \{3 \times 3 \text{ conv.}, 3 \times 3 \text{ dil.conv.}, 5 \times 5 \text{ conv.}, 5 \times 5 \text{ dil. conv.}, 7 \times 7 \text{ conv.}, 3 \times 3 \text{ max pooling}, 3 \times 3 \text{ avg. pooling, skip, zero}\}$, respectively. Especially, for the NAS-Bench-201 search space, we additionally use the Jung et al. [22] dataset that includes robust accuracies of candidate neural architectures in NAS-Bench-201 search space to demonstrate the efficacy of our proposed proxy regarding searching generalized architectures against diverse perturbations and clean inputs.

Adversarial Evaluation To evaluate standard-trained models, we utilize the robust NAS-Bench-201 [22] datasets, allowing us to achieve robust accuracy. On the other hand, to evaluate adversarially-trained models, we construct our own dataset, as described in Section 4.2, enabling us to obtain robust accuracy against adversarial attacks. In all our experiments, we obtain robust accuracy on CIFAR-10 against FGSM attack with an attack size (ϵ) of 8.0/255.0 and attack step size (α) of 8.0/2550.0 while we utilize robust accuracies against FGSM attack with attack size (ϵ) of 4.0/255.0 on ImageNet16-120.

End-to-end sampling We sample 5,000 number of neural architectures on the DARTS search space to obtain end-to-end performance on CIFAR-10 and CIFAR-100. For both CIFAR-10 and CIFAR-100, we utilize the AE+warmup+move sampling strategy described in Section 4.3, where our proxy guides the sampling towards the pool of architectures with high proxy values. Specifically, for CIFAR-10, we use an init pool (e.g., warmup) of 3,000 and a sample pool for AE (e.g., move) of 50. On the other hand, we employ an init pool of 3,000 and a sample pool of 100 for CIFAR-100.

B Experimental Results

B.1 Ablation on Each Component of CRoZe

In Section 3.3, we introduce our proxy, which consists of three components: feature, parameter, and gradient consistency. We discuss the importance of considering all three components to accurately evaluate the robustness of neural architectures in a random state (Section 4.4). To further analyze the contributions of each factor, we conduct an ablation study in both the NAS-Bench-201 search space and the DARTS search space.

We find that relying solely on feature consistency in a random state is insufficient to evaluate the robustness of architectures. The proxy with only feature consistency shows a lower correlation in both standard training and adversarial training scenarios compared to CRoZe in the NAS-Bench-201 search space (Table 7a and Table 7b). This indicates that high scores obtained based on feature consistency on a single batch may not accurately reflect the performance across the entire dataset. On the other hand, when parameter or gradient similarity is added to the proxy, the correlation consistently improves, suggesting that these factors complement feature consistency by imposing

Table 7: Comparison of Spearman's ρ between the actual accuracies and the proxy values on CIFAR-10 in the NAS-Bench-201 search space. Clean stands for clean accuracy and robust accuracies are evaluated against adversarial perturbations (FGSM [16]) and common corruptions (CC. [20]). Avg. stands for average Spearman's ρ values with all accuracies.

Proxy components			Standard-Trained			
Feature	Parameter	Gradient	Clean	FGSM	CC.	Avg.
✓	–	–	0.718	0.701	0.341	0.587
✓	✓	–	0.750	0.762	0.384	0.632
✓	–	✓	0.822	0.824	0.434	0.693
–	✓	✓	0.824	0.827	0.437	0.696
✓	✓	✓	0.823	0.826	0.436	0.695

(a) Standard-Trained

Proxy components			Adversarially-Trained				
Feature	Parameter	Gradient	Clean	FGSM	PGD	HRS(FGSM)	HRS(PGD)
✓	–	–	0.602	0.295	0.329	0.442	0.404
✓	✓	–	0.677	0.343	0.431	0.542	0.527
✓	–	✓	0.707	0.405	0.489	0.587	0.573
–	✓	✓	0.731	0.422	0.507	0.610	0.595
✓	✓	✓	0.723	0.417	0.501	0.602	0.588

(b) Adversarially-Trained

Table 8: Comparisons of the final performance of the searched network in NAS-Bench-201 and DARTS search space on CIFAR-10. **Bold** and underline stands for the best and second.

Proxy components			Standard-Trained (Top-1)				Standard-Trained (Top-3)			
Feature	Parameter	Gradient	Clean	FGSM	CC.	Avg.	Clean	FGSM	CC.	Avg.
✓	–	–	93.30	44.30	55.62	64.41	92.93	37.60	54.03	61.52
✓	✓	–	93.70	45.80	56.93	65.48	93.63	48.20	55.39	**65.74**
✓	–	✓	93.70	45.80	56.93	65.48	93.63	48.20	55.39	**65.74**
–	✓	✓	93.70	45.80	56.93	65.48	93.43	41.37	55.30	63.37
✓	✓	✓	93.70	45.80	56.93	65.48	93.93	43.87	56.11	64.64

(a) NAS-Bench-201 search space

Proxy Components			Standard-Trained				
Feature	Parameter	Gradient	Clean	CC.	FGSM	HRS	Avg.
✓	–	–	94.37	72.26	16.87	28.62	53.03
✓	✓	–	**94.99**	74.06	16.82	28.58	53.86
✓	–	✓	94.30	**74.91**	16.67	28.33	53.55
–	✓	✓	94.34	74.46	15.71	26.93	52.86
✓	✓	✓	94.45	74.63	**22.38**	**36.19**	**56.91**

(b) DARTS search space

stricter constraints on the parameter space and convergence directions, respectively. While the proxy considering only parameter and gradient similarity achieves better Spearman's ρ compared to our proxy, the top-3 architectures chosen by our proxy exhibit higher average performance than those discovered by the proxy without feature consistency (Table 8a).

We further conduct ablation experiments on CIFAR-10 in the DARTS search space, which contains about 10^{19} number of candidate architectures that is significantly larger than the NAS-Bench-201 search space containing 15625 architectures. The proxy without feature consistency yielded architectures with poor robust accuracies, while the architectures selected by our proxy consistently outperformed the former on both clean and perturbed images (Table 8b). Furthermore, architecture identified solely by feature consistency exhibits better average performance compared to those discerned by proxies without feature consistency. This clearly demonstrates the influential role of feature consistency in evaluating robustness. Overall, our proposed proxy effectively searches high-performing architectures by employing consistency across features, parameters, and gradients to estimate the robustness of the given architectures within a single gradient step. The overall algorithm of CRoZe is described in Algorithm 1.

B.2 Ablation on Perturbation Type of Input

CRoZe evaluates the consistency between clean and perturbed inputs, regardless of the perturbation type, assuming that perturbed inputs retain the same semantic information as clean inputs. To empirically demonstrate the independence of our proxy from specific perturbation types applied to the input, we introduce random Gaussian noise in place of adversarial perturbations in Eq. 4 for obtaining our proxy values. As evidenced in Table 9, CRoZe consistently exhibits similar Spearman's ρ between the final performance of the standard-trained models and our proxy value in the NAS-Bench-201 search space on multiple benchmarks including CIFAR-10, CIFAR-100, and ImageNet16-120. Specifically, the gap of the average Spearman's correlation between the CRoZe using adversarial perturbations and the one with random Gaussian noise is merely 0.001 on CIFAR-100. Furthermore, when we measure the Spearmans' ρ between the final performance of adversarially-trained models and our proxy, the average Spearman's ρ values are the same between the CRoZe with random Gaussian noise and the one with adversarial perturbations (Table 10). This emphasizes that CRoZe captures the characteristics of robust architectures through the consistency of clean and perturbed inputs, irrespective of the perturbation types employed.

Table 10: Comparisons of perturbation type applied to the input. All models are adversarially-trained on CIFAR-10.

	Adversarially-Trained					
		PGD	FGSM	HRS		
	Clean	$\epsilon = 1$	$\epsilon = 8$	PGD	FGSM	Avg.
Gaussian Noise	0.718	0.503	0.415	0.590	0.603	0.566
Adversarial	0.723	0.501	0.417	0.588	0.602	0.566

Algorithm 1: **C**nsistency-based **Ro**bust **Ze**ro-cost Proxy (**CRoZe**).

Input: A single batch of given dataset $B = \{(x, y)\}$, network $f_\theta(\cdot)$ consists of M layer, which is architecture $\mathcal{A}$ with parameterized by θ

Output: Proxy value, *CRoZe*

/* Estimate robust network f_{θ^r} as done in Eq. 3. */

for $m = 1, \cdots, M$ **do**

$$\theta_m^r \leftarrow \theta_m + \beta * \frac{\nabla_{\theta_m} \mathcal{L}\big(f_\theta(x), y\big)}{\|\nabla_{\theta_m} \mathcal{L}\big(f_\theta(x), y\big)\|} * \|\theta_m\|$$

/* Generate perturbed input x' using f_{θ^r}. as done in Eq. 4 */

$$\delta = \epsilon \texttt{sign}\Big(\nabla_x \mathcal{L}(f_{\theta^r}(x), y)\Big)$$
$$x' = x + \delta$$

/* Calculate consistency in features, parameters, and gradients as done in Section 3.3 */

/* Calculate gradients of both clean network f_θ and robust network f_{θ^r} */

$$g = \nabla_\theta \mathcal{L}\big(f_\theta(x), y\big)$$
$$g^r = \nabla_{\theta^r} \mathcal{L}\big(f_{\theta^r}(x'), y\big)$$

/* Single gradient step for clean network f_θ and robust network f_{θ^r} */

$$\theta_1 \leftarrow \theta - \gamma g$$
$$\theta_1^r \leftarrow \theta^r - \gamma g^r$$

for $m = 1, \cdots, M$ **do**

/* Feature consistency */

$$\mathcal{Z}_m(f_\theta(x), f_{\theta^r}(x')) = 1 + \frac{z_m \cdot z_m^r}{\|z_m\| \|z_m^r\|}$$

/* Parameter consistency */

$$\mathcal{P}_m(\theta_1, \theta_1^r) = 1 + \frac{\theta_{1,m} \cdot \theta_{1,m}^r}{\|\theta_{1,m}\| \|\theta_{1,m}^r\|}$$

/* Gradient consistency */

$$\mathcal{G}_m(g, g^r) = \Big| \frac{g_m \cdot g_m^r}{\|g_m\| \|g_m^r\|} \Big|$$

$$\textit{CRoZe} = \sum_{m=1}^{M} \mathcal{Z}_m \times \mathcal{P}_m \times \mathcal{G}_m$$

return *CRoZe*;

Table 9: Comparison of Spearman's ρ between the actual accuracies and the proxy values on CIFAR-10, CIFAR-100 and ImageNet16-120 in NAS-Bench-201 search space. Avg. stands for average Spearman's ρ values with all accuracies within each task and CC. stands for the average of 15 different types of common corruption. All models are standard-trained.

	CIFAR-10						CIFAR-100				ImageNet16-120		
		FGSM					FGSM				FGSM		
Perturbation Type	Clean	$\epsilon = 8$	$\epsilon = 4$	$\epsilon = 2$	CC.	Avg.	Clean	$\epsilon = 4$	CC.	Avg.	Clean	$\epsilon = 4$	Avg.
Gaussian Noise	0.810	0.821	0.797	0.778	0.436	0.728	0.774	0.693	0.542	0.670	0.741	0.671	0.706
Adversarial	0.823	0.823	0.826	0.801	0.436	0.682	0.787	0.693	0.533	0.671	0.769	0.696	0.733

B.3 Ablation of the Weight Initialization Type

As our proxy assesses the robustness of the neural architecture within a single gradient step, we further investigate the sensitivity of our proxy to various weight initialization types. To validate the compatibility of our proxy with the diverse weight initialization type, we perform experiments employing Random initialization, Kaiming initialization [19], and Xavier initialization [15] on the NAS-Bench-201 search space on CIFAR-10. We measure Spearman's ρ between the final accuracies and the proxy values, where the final performances are obtained by evaluating both

Table 11: Comparison of Spearman's ρ between the final accuracies and the proxy values on the NAS-Bench-201 search space with various weight initialization methods. All models are trained with both standard and adversarial training on CIFAR-10.

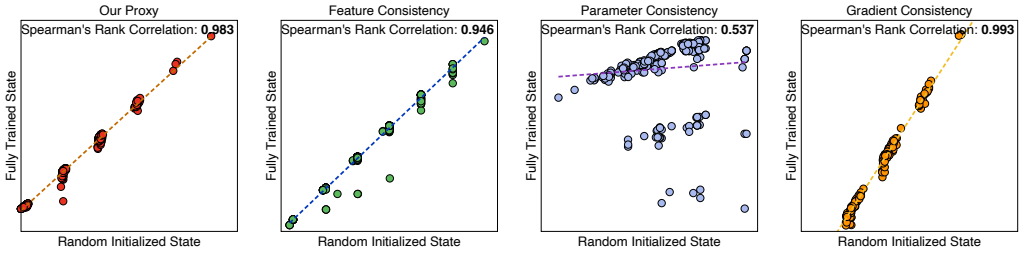
	Standard-trained					Adversarially-trained							
Weight Initialization Type	Clean	FGSM	PGD	CC.	Avg.	FGSM	PGD	CW	DeepFool	SPSA	LGV	AutoAttack	Avg.
Random	0.823	0.826	0.188	0.436	0.568	0.441	0.532	0.220	0.454	0.240	0.449	0.458	0.399
Kaiming [19]	0.812	0.818	0.189	0.430	0.562	0.428	0.512	0.217	0.443	0.227	0.426	0.436	0.384
Xavier [15]	0.816	0.822	0.190	0.433	0.565	0.428	0.513	0.217	0.442	0.227	0.425	0.436	0.384

standard-trained models and adversarially-trained models. We used 15,625 standard-trained models and 300 adversarially-trained models to ensure precise robustness assessment. Standard trained models are validated on clean images, adversarially-perturbed images (i.e., FGSM with an attack size of $8.0/255.0$ and PGD with an attack size of $1.0/255.0$), and 15 different types of common corrupted images. In contrast, advesrsarially-trained models are subjected to evaluation against 7 different types of strong adversarial attacks, including FGS, PGD, CW, DeepFool, SPSA, LGV, and AutoAttack. The results reported in the main paper are based on Random initialization.

As demonstrated in Table 11, our proxy maintains a consistently higher correlation compared to the baselines, irrespective of the weight initialization methods employed. Specifically, our proxy achieves an average correlation of 0.568 and 0.399 for standard training and adversarial training scenarios, respectively, whereas the best-performing baseline method only achieves 0.529 and 0.352 against various perturbations with the same random weight initialization. Since our approach considers the consistency of the parameters and gradients between the clean and perturbed images, our superior performance can be achieved regardless of the weight initialization method.

B.4 Assessment of CRoZe Predictiveness on Adversarially-Trained Models

Figure 7: Spearman's ρ in our proxy and each consistency component between the neural architectures with single-step trained states and fully-trained states against clean and perturbed images. All models are trained only on adversarially-perturbed images.



As we verify the predictiveness of CRoZe with 300 standard-trained models in the NAS-Bench-201 search space in Section 4.4, we further validate the predictiveness of our proxy against adversarially-trained models. To achieve adversarial robustness, we adversarially train 300 randomly sampled architectures from the NAS-Bench-201 search space on the entire CIFAR-10 dataset, following [40]. We then compute Spearman's ρ between the values obtained from the fully-adversarially-trained states and single-step trained states associated with our proxy and each consistency component.

Surprisingly, our proxy demonstrates a strong correlation of 0.983 between the proxy value obtained from a fully-trained state and a single-step trained state (Figure 7). This indicates that our proxy accurately predicts the robustness of architectures even when they are adversarially-trained. Furthermore, each individual component of the proxy also exhibits a high correlation, suggesting that a strong predictiveness of our proxy against diverse perturbations does not rely on a dominant component, but rather on the overall precise evaluation provided by the combination of components.

B.5 Versatility of CRoZe

To demonstrate the versatility of our robust zero-shot proxy, CRoZe, and its compatibility with existing clean zero-shot NAS approaches [1], we conduct additional experiments where we ensemble our proxy with other methods. Following Abdelfattah et al. [1], we re-implemented ensemble-based NAS, which involves aggregating predictions from multiple zero-shot NAS methods to calculate the

Table 12: Validation of orthogonality of CRoZe. Comparison of Spearman's ρ between the actual accuracies and the proxy values on CIFAR-10, CIFAR-100, and ImageNet16-120 in the NAS-Bench-201 search space. CC. stands for the average of 15 different types of common corruption. All models are standard-trained.

	CIFAR-10				CIFAR-100				ImageNet16-120		
Proxy Type	Clean	FGSM	PGD	CC.	Clean	FGSM	PGD	CC.	Clean	FGSM	PGD
CRoZe	0.823	0.826	0.188	0.823	0.784	0.786	0.343	0.784	0.765	0.596	0.707
Ensemble	0.803	0.681	0.543	0.771	0.793	0.739	0.444	0.798	0.749	0.638	0.296
CRoZe+Ensemble	0.894	0.872	0.633	0.894	0.894	0.851	0.415	0.878	0.810	0.688	0.259

Table 13: Comparisons of the final performance and required computational resources of the searched neural architectures in the DARTS search space on ImageNet16-120. CC. stands for the average of 15 different types of common corruption.

NAS Method	Training-free NAS	Params (M)	# GPU	Batch Size	Standard-Trained			
					Clean	CC.	FGSM	HRS
PC-DARTS [42]		5.30	8	1024	50.58	14.36	0.15	0.30
DrNAS [6]		5.70	8	512	49.63	13.42	0.21	0.42
AdvRush [31]		4.20	1	64	38.72	10.39	0.11	0.22
GradNorm [1]	✓	5.90	1	8	39.13	10.75	0.23	0.46
SynFlow [1]	✓	6.13	1	8	43.73	12.10	0.15	0.30
CRoZe	✓	5.87	1	8	47.90	13.35	0.32	0.64

proxy score for a given neural architecture. For our ensemble, we select {Snip, NASWOT} as the baseline. We then compare 1) CRoZe, 2) Ensemble: {Snip, NASWOT}, and 3) CRoZe+Ensemble: {CRoZe, Snip, NASWOT} in the NAS-Bench-201 search space on CIFAR-10, CIFAR-100, and ImageNet16-120. Robust accuracies are obtained by evaluating the standard-trained models against FGSM attack with an attack size of $8.0/255.0$ and PGD with an attack size of $1.0/255.0$.

The results presented in Table 12 indicate that CRoZe significantly enhances the predictiveness for both clean and robust accuracies, encompassing adversarial attacks and common corruption across all tasks. Notably, CRoZe+Ensemble showcases substantial improvements, achieving increases of 11.33% in clean accuracy and 28.05%, 16.57%, and 15.95% in robustness concerning FGSM attack, PGD attack, and common corruption, respectively on CIFAR-10. These results underscore the effectiveness of our proxy when combined with other proxies, enabling a more precise robust neural architecture search.

B.6 End-to-End Performance on ImageNet16-120

We further validate the final performance of the searched neural architectures by CRoZe and compare the required computational resources with existing NAS frameworks including robust NAS (AdvRush [31]), clean one-shot NAS (PC-DARTS [42], DrNAS [6]) and clean zero-shot NAS (SynFlow and GradNorm [1]), on ImageNet16-120 in the DARTS search space. Similar to Section 4.3, we sample the same number (e.g., 5,000) of candidate architectures using the warmup+move strategy with an init pool of 3,000 and sample pool of 50 for both clean-zero shot NAS and CRoZe.

The NAS Training-free methods such as GradNorm, SynFlow, and CRoZe only require a single GPU with a batch size of 8 to search for the architectures on the ImageNet16-120 dataset. In contrast, the existing clean one-shot NAS methods require 8 GPUs with much larger batch sizes. Moreover, NAS-Training-free methods consume less than 3000MB of memory, while both clean one-shot NAS and robust NAS need at least 3090 RTX GPU, which is available at 24000MB of memory. With its superior computational efficiency, CRoZe enables rapid neural architecture search and achieves the best HRS accuracy while maintaining comparable clean and common corruption accuracies. all at a much lower computational cost (Table 13). These demonstrate the effectiveness of CRoZe for rapid and lightweight robust NAS across diverse tasks (i.e., CIFAR-10, CIFAR-100, and ImageNet16-120) and perturbations (i.e., adversarial perturbations and common corruptions).