

## A FURTHER ANALYSIS ON THE STATISTICS OF *NARes*

### A.1 DISTRIBUTION ON MODEL COMPLEXITY

We demonstrate the distribution of architectures in *NARes* on model complexity in Fig. 8a. The architectures in *NARes* cover a wide range of the number of parameters and MACs with overlapping. According to the marginal distribution, various architectures of *NARes* may share similar parameters or MACs. For example, the architecture with decision vector [5, 8, 4, 16, 9, 10] has a similar number of MACs ( $\sim 10G$ ) to architectures with decision vectors [4, 8, 5, 16, 5, 12], [4, 12, 9, 10, 4, 14] and [4, 14, 5, 14, 5, 10]. Therefore, *NARes* provides a comprehensive dataset for the research community to study the relationship between model complexity and adversarial robustness (AR).

To illustrate how different depths and widths affect these two model complexity metrics, we further investigate the depth and width distribution over the number of parameters and MACs, as shown in Fig. 8b. We find the value of the depth or width factor at the later stage has more impact on #Params, while #MACs is less sensitive to the depth and width at different stages. The reason is that #MACs of a convolution layer are determined by  $(k^2 \cdot C_{in}hw \cdot C_{out})$ , where  $k$  is the kernel size,  $h$  and  $w$  are the height and width of the input feature map and  $C_{in}$ ,  $C_{out}$  are the number of input and output channels. Hence, downsampling input feature map size in half will amortize the doubling of channels at  $C_{in}$  and  $C_{out}$  in the next stage. It suggests that blocks of different stages with the same depth and width values could share similar MACs.

### A.2 ROBUST OVERFITTING

Robust overfitting is a common issue in AT, where the model performs well on the normal examples, but the accuracy on adversarial examples starts to decrease at the later stage of training, especially after the first decay of learning rate (Rice et al., 2020). In order to measure how different architectures in *NARes* are affected by the robust overfitting issue, we plot the distribution of models' best epoch and the training curve in Fig. 9. Unfortunately, robust overfitting consistently happens on all models. The adversarial accuracy on the validation set starts to decrease soon after the first decay of the learning rate, and usually, the best epoch is just before the first decay of the learning rate. This observation suggests that the robust overfitting issue cannot be fully resolved simply by searching for new architecture in our search space.

### A.3 ROBUST ACCURACY

In this section, we extend the analysis of the robustness of *NARes* in Sec. 4.1. We plot the validation accuracies in models of *NARes*, as shown in Fig. 10. The shapes of the distributions are similar to

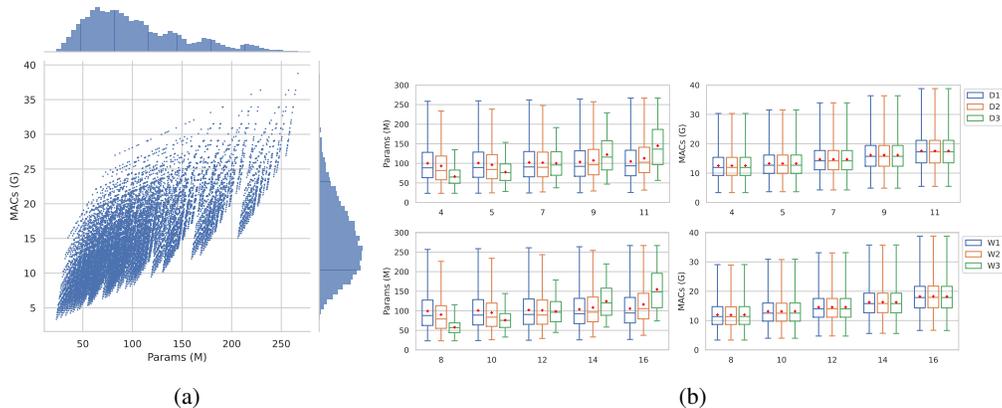


Figure 8: The distribution of *NARes* on model complexity. **(a)**: Overview of all architectures concerning the number of parameters and MACs. **(b)**: The box plot of the depth and width factor v.s. parameters and MACs across the search space; the red "+" sign represents the mean value of each group.

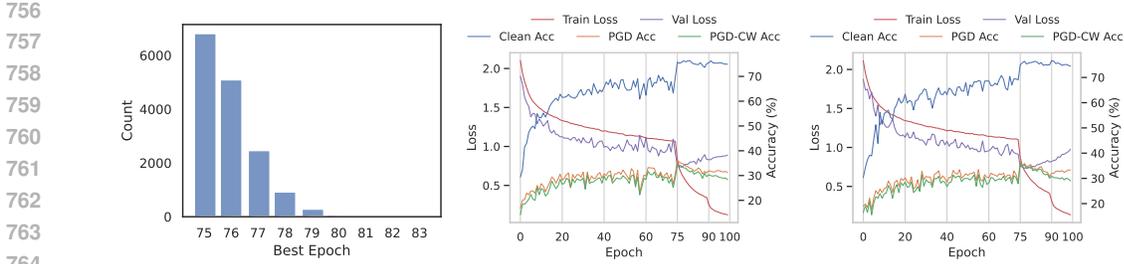


Figure 9: Robust overfitting on *NARes*. **Left**: The distribution of the best epoch of models in the search space. **Right**: The training curve of loss and validation accuracies on two models sampled from *NARes*.

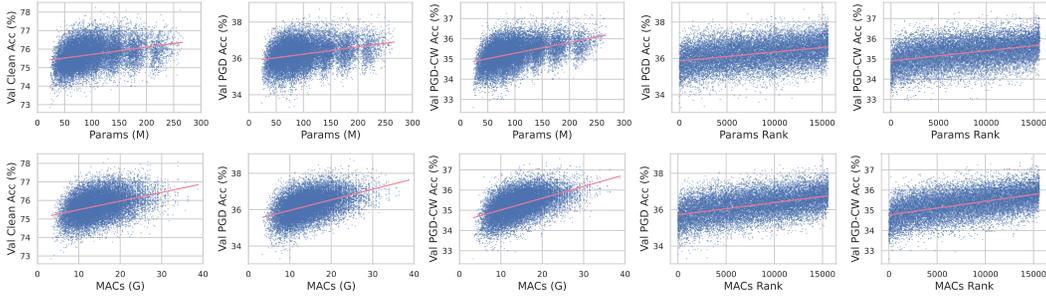


Figure 10: Statistics on the validation set of CIFAR-10.1. The clean accuracy, the PGD<sup>20</sup> accuracy, the PGD-CW<sup>40</sup> accuracy and the stable accuracy and empirical Lipschitz constant of PGD<sup>20</sup> are plotted against the number of parameters and MACs on the left side. Besides, the adversarial robustness against the rank of parameters and MACs is shown on the right side.

those on the test set in Fig. 2. However, the results indicate that both the clean and robust accuracies on the validation set are lower than those on the test set. This phenomenon has been observed in the original work of CIFAR-10.1 (Recht et al., 2018) and is attributed to the intrinsic potential distribution shift. Furthermore, Fig. 11a plots the statistics of the accuracies on the validation set and test set. According to the results, correlations of AR on the same evaluated dataset are high. While the correlations between the validation AR and the test AR are relatively small, unlike the results of other neural architecture datasets such as NAS-Bench-201 (Dong & Yang, 2019) on the clean accuracy. But from the overall view in Fig. 11b, higher validation adversarial accuracies are still a good indicator of better test adversarial robustness. Moreover, we find that the validation accuracy on the PGD-CW<sup>40</sup> attack has the highest correlation value to AR on the test set, which could be a consequence of the best epoch selection strategy by the PGD-CW<sup>40</sup> accuracy during adversarial training. We also find that the test FGSM accuracy is relatively less correlated with the validation AR on PGD<sup>20</sup> and PGD-CW<sup>40</sup> compared to clean accuracy. This could be a reason for the number of iteration steps in FGSM is one, where the decision boundary of the model is not effectively explored like the other two attacks are, hence the nature of the FGSM attack may be different from the other two attacks across different architectures.

To avoid overinterpreting the shape of the distributions, we further investigate the AR at a subset of the original search space in *NARes*. we consider two subspaces:  $\{D_{i \in \{1,2,3\}} \in \{5, 7, 9, 11\}, W_{i \in \{1,2,3\}} \in \{8, 10, 12, 14, 16\}\}$  and  $\{D_{i \in \{1,2,3\}} \in \{4, 5, 7, 9, 11\}, W_{i \in \{1,2,3\}} \in \{10, 12, 14, 16\}\}$ , which contains 8000 models respectively. Fig. 12 shows the two subspaces' AR on the test set, which is consistent with the shapes of the full search space in Fig. 2.

#### A.4 STABLE ACCURACY AND EMPIRICAL LIPSCHITZ CONSTANT ON OTHER ATTACKS

In this section, we further explore the relations of stable accuracy and empirical Lipschitz constant to AR. Fig. 13 and Fig. 14 show the statistics of the stable accuracy and empirical Lipschitz constant of

810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863

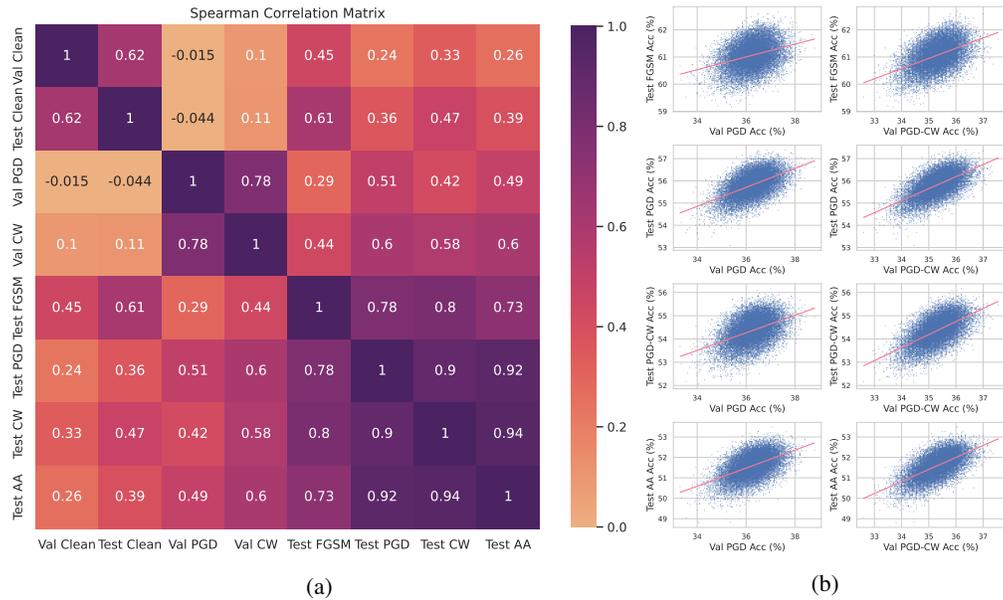


Figure 11: Statistics of accuracies on the validation set and test set. (a): The Spearman correlation matrix of the accuracies on the validation set and test set. (b): The validation adversarial robustness vs. the test adversarial robustness.

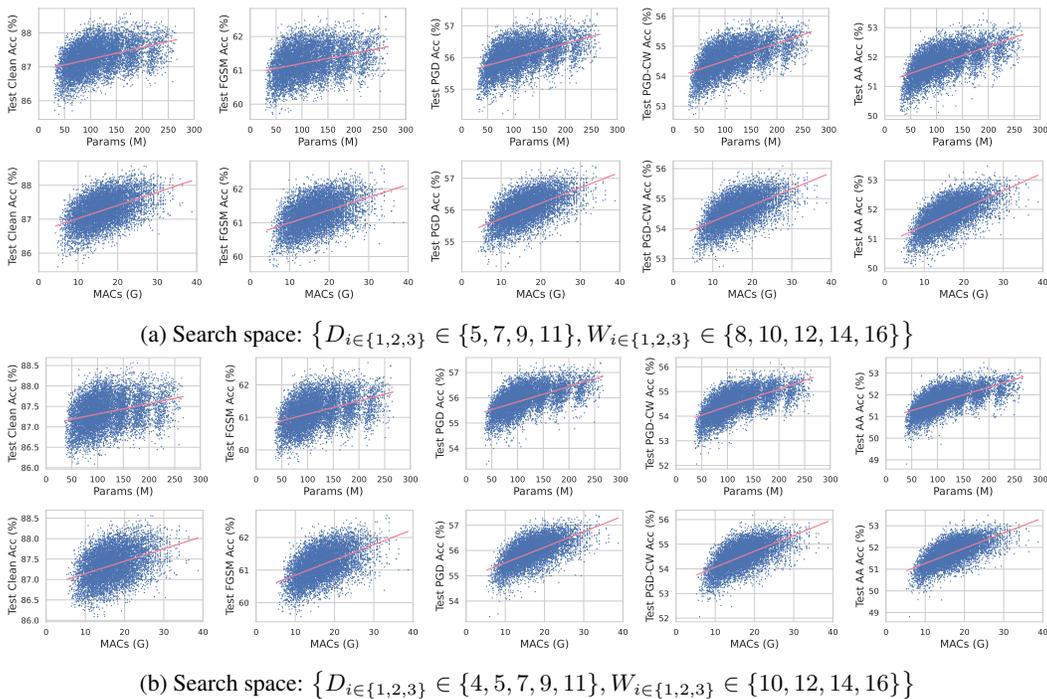


Figure 12: The clean accuracy and adversarial accuracies of different attacks in subspaces of *NARes*.

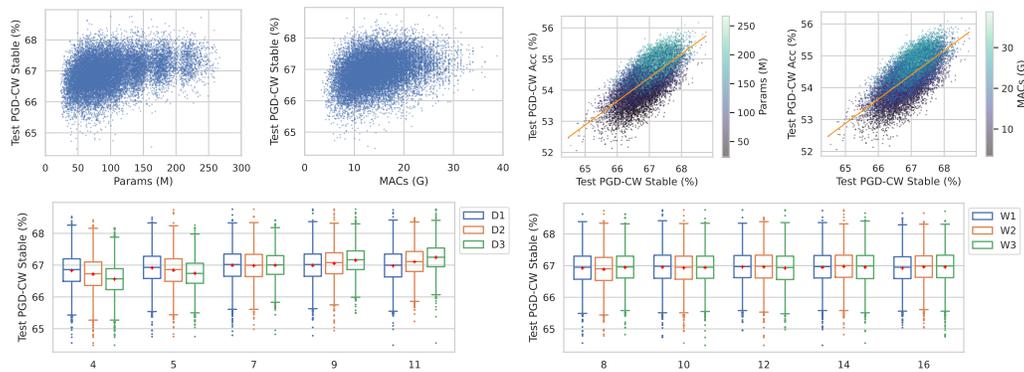


Figure 13: The statistics of PGD-CW<sup>40</sup> stable accuracy on the test set. In box plots, the red "+" sign represents the mean accuracy of each group.

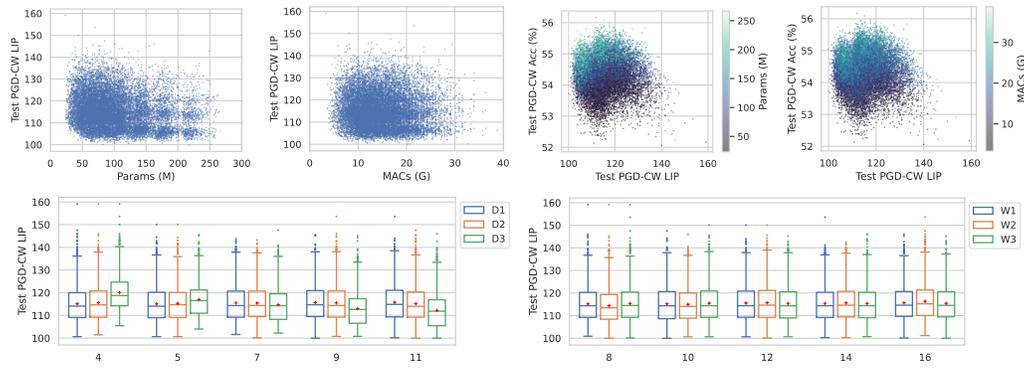


Figure 14: The statistics of PGD-CW<sup>40</sup> empirical Lipschitz constant on the test set. In box plots, the red "+" sign represents the mean accuracy of each group.

the test set under the PGD-CW<sup>40</sup> attack. And the shape and the conclusion are similar to the PGD<sup>20</sup> attack mentioned in Sec. 4.2.

Moreover, the correlation matrix in Fig. 15 suggests the empirical Lipschitz constant could be a dataset-agnostic metric, representing the intrinsic nature of the model and negative to stable accuracies. In contrast, the stable accuracies between the validation and test sets are not relatively highly correlated. However, the results also suggest that neither the stable accuracy nor empirical Lipschitz constant on the validation set is a good indicator of test adversarial robustness compared to the validation adversarial accuracies.

#### A.5 ROBUSTNESS ON COMMON CORRUPTIONS

Besides adversarial attacks, we also measure the robustness of models in *NARes* with common corruptions, as complementary robustness metrics. The Spearman correlation table over the accuracies of each corruption group, in addition to metrics of the test set, is shown in Fig. 16. On these non-worst perturbations of corruption types, its robust accuracy is only highly correlated with clean accuracy, suggesting that the worst-case adversarial robustness is quite different from common corruption robustness.

The average accuracies under different severity and types of corruption are shown in Fig. 17. We observe that the accuracies of every corruption type consistently decrease with the severity level, especially with the corruption of impulse noise, fog, and contrast. We take a few samples from these corruption types with an additional Gaussian noise as the representative of other common corruptions, and illustrate them in Fig. 18 for better understanding. Moreover, in Fig. 19, we plot the detailed correlation matrices of each corruption type at different severity levels with the accuracies on the

918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971

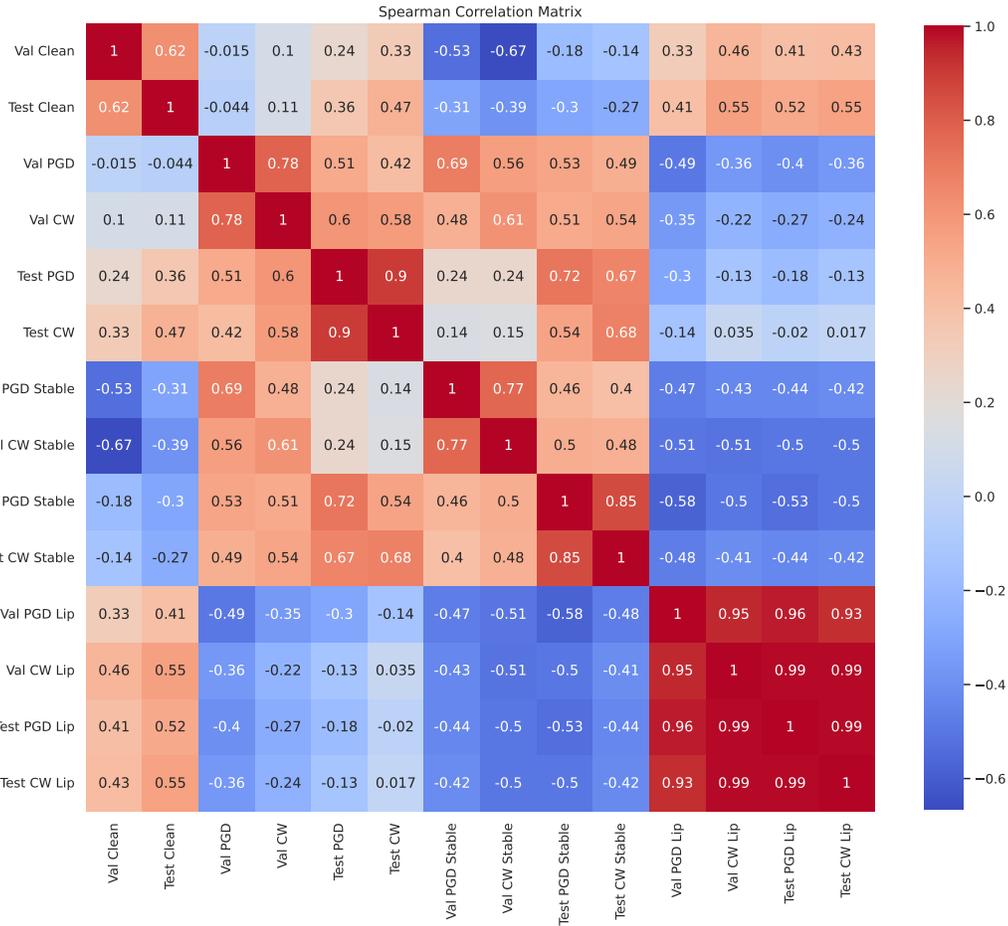


Figure 15: The Spearman correlation matrix of the accuracies, stable accuracies and empirical Lipschitz constants on the validation set and test set.



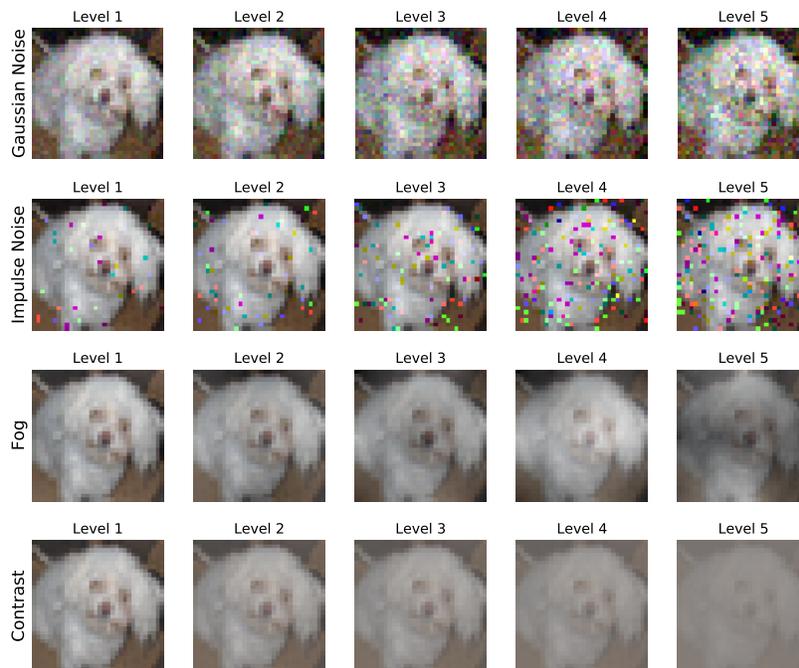
Figure 16: The Spearman correlation table about grouped corruption accuracies on CIFAR-10-C (Noise, Blur, Weather, Digital, Average) and the metrics on the CIFAR-10 test set (Test Loss, Clean, FGSM, PGD<sup>20</sup>, PGD-CW<sup>40</sup>).

972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986

Level	1	2	3	4	5	Corruption Average
Gaussian Noise	86.3	86.8	83.8	86.8	86.6	82.3
Shot Noise	84.8	86.2	80.2	85.7	85.2	82.4
Impulse Noise	83.8	86.8	86.6	82.3	86.6	84.3
Speckle Noise	86.8	86.6	82.3	86.6	84.3	83.3
Defocus Blur	82.3	86.6	84.3	83.3	86.3	85.3
Glass Blur	86.3	85.3	84.8	81.6	77.0	87.4
Gaussian Blur	87.4	86.1	82.7	82.6	86.6	85.8
Motion Blur	82.7	82.6	86.6	85.8	83.7	85.2
Zoom Blur	86.2	83.7	76.9	84.8	83.3	81.9
Snow	80.6	82.2	70.2	82.0	81.3	80.9
Frost	82.2	83.7	76.9	84.8	83.3	81.9
Fog	80.6	82.2	70.2	82.0	81.3	80.9
Brightness	78.8	79.3	64.3	78.3	75.6	76.9
Spatter	78.8	79.3	64.3	78.3	75.6	76.9
Contrast	78.8	79.3	64.3	78.3	75.6	76.9
Elastic Transform	78.8	79.3	64.3	78.3	75.6	76.9
Pixelate	78.8	79.3	64.3	78.3	75.6	76.9
JPEG Compression	78.8	79.3	64.3	78.3	75.6	76.9
Saturate	78.8	79.3	64.3	78.3	75.6	76.9
Corruption Average	78.8	79.3	64.3	78.3	75.6	76.9

987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999

Figure 17: The average accuracies of *NARes* under 19 types of corruptions with 5 severity levels on CIFAR-10-C. The average corruption accuracies over each corruption type are also shown in the rightmost column.



1021  
1022  
1023  
1024  
1025

Figure 18: Examples of some corruption types over 5 severity levels on a single image.

1026  
1027  
1028  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079

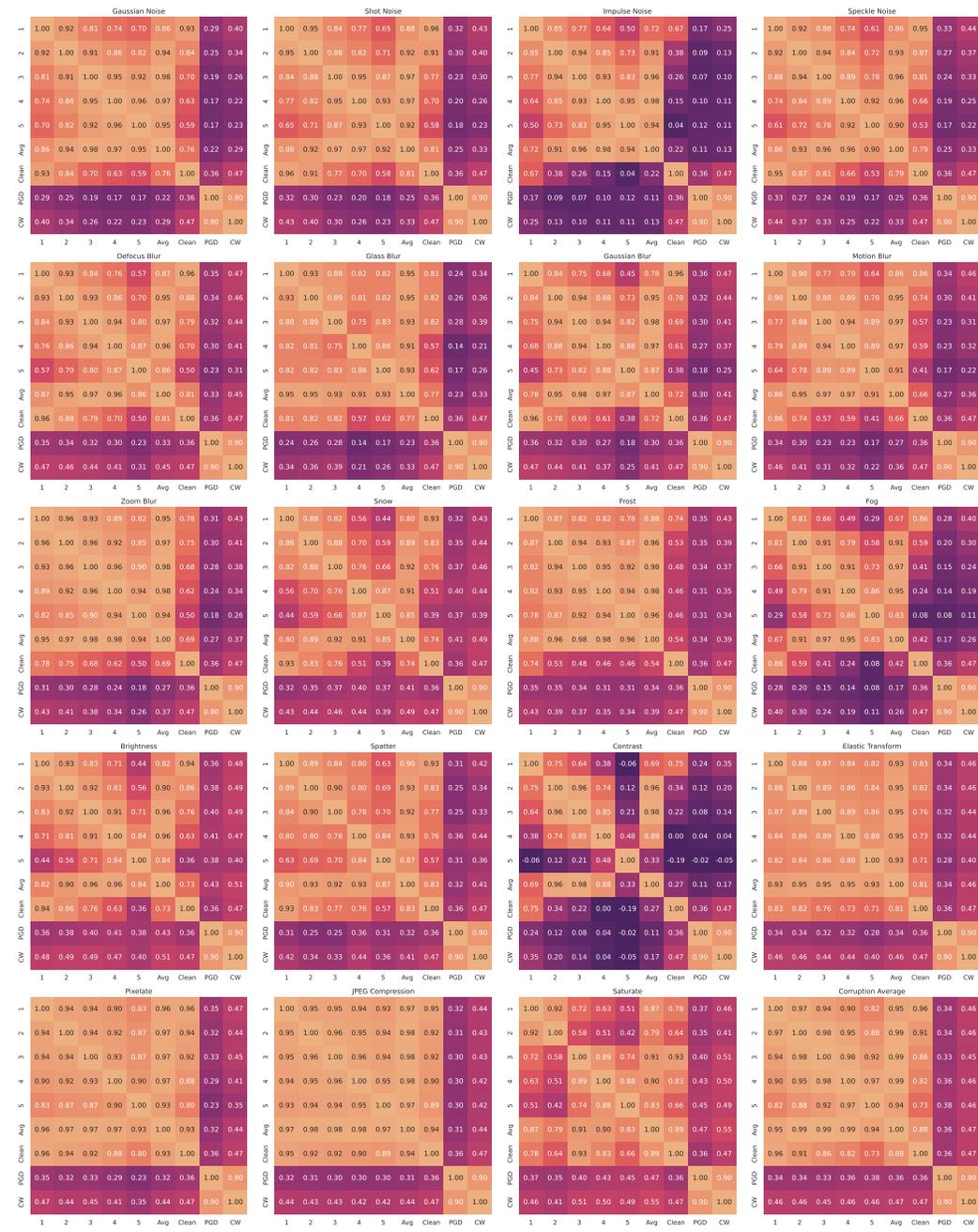


Figure 19: The Spearman correlation matrices of the accuracies of each corruption type at different severity levels, concerning the accuracies (Clean, PGD<sup>20</sup> and PGD-CW<sup>40</sup>) on the CIFAR-10 test set.

CIFAR-10 test set, where the correlations to the clean accuracy decrease with the severity level. The results also suggest that the robustness of common corruptions is not highly correlated with the adversarial robustness compared to clean accuracy, substantiating the above statement.

In summary, the robustness of common corruption is a rational complementary metric. Since the levels of corruption severity control the perturbation ranges to the original images, the images under different corruption levels naturally form datasets with different degrees of out-of-distribution (OOD) samples with the corruption shift. Therefore, *NARes* can also be a potential neural architecture dataset for OOD robustness.

Table 3: The adversarial and common corruption robustness of common models in *NARes*.

Model	Clean	FGSM	PGD <sup>20</sup>	PGD-CW <sup>40</sup>	AA-Compact*	AA*	Corruption <sup>+</sup>
WRN-28-8	85.61	59.52	53.68	52.17	49.05	49.03	77.03
WRN-28-10	87.12	60.60	54.58	52.98	50.09	50.06	78.76
WRN-34-10	86.54	59.89	54.84	52.98	50.22	50.21	78.52
WRN-34-12	86.85	60.23	55.17	53.45	50.52	50.51	78.56
WRN-46-12	87.86	62.13	56.37	55.02	52.02	51.99	79.52
WRN-46-14	87.08	61.43	56.43	55.20	52.61	52.60	78.86
WRN-58-14	87.37	61.98	56.85	55.41	52.70	52.75	79.20
WRN-70-16	87.22	61.01	56.29	54.86	52.25	52.24	79.13

\* : The accuracies of AA-Compact and AA are reported under different evaluation runs.

+ : The average accuracy over all 19 corruption types on CIFAR-10-C.

## A.6 ROBUSTNESS ON COMMON MODELS

Table 3 summarizes the metrics of commonly used architectures in adversarial robustness domains that are covered by our search space. Notably, our models achieve better robustness compared to previous works at the standard adversarial training (Madry et al., 2018; Huang et al., 2021; 2023), which indicates the effectiveness of the training strategy in *NARes*.

## A.7 GENERALIZATION TO OTHER DATASETS

Due to the expensive adversarial training cost, it is prohibitive to enumerate the entire search space on other larger datasets. To investigate the generalization of the statements in *NARes*, we further evaluate a small set of architectures on Tiny-ImageNet (Le & Yang, 2015), which contains 200 categories for classification. The training set includes 500 images for each class. And we split the original validation set in half to form a validation set and a test set. Each set contains 25 images per class. To investigate the influence of the last stage architecture ( $W_3$  and  $D_3$ ) on AR, we select the central architecture in the search space with decision vector [7, 12, 7, 12, 7, 12]; then we choose the other 24 architectures with variant  $W_3$  and  $D_3$  to form a set of 25 architectures. The training procedure is similar to that on CIFAR-10, except the input image size is 64x64.

The results are shown in Fig. 20, which has the same tendency in the overall view of CIFAR-10. We observe that generally increasing the depth and width of the last stage will improve the adversarial robustness on Tiny-ImageNet, consistent with the results of *NARes* on CIFAR-10. It substantiates the statement in Sec. 4.1 that the previous consensus of design principles might not be true. Moreover, we find a similar robust overfitting issue in Appendix A.2, as shown in Fig. 21. These observations suggest the generalization ability of *NARes* to other datasets.

# B DETAILS ON THE GENERATION OF *NARes*

## B.1 ADVERSARIAL TRAINING DETAILS

A fixed set of hyperparameters was used for all models in *NARes*. Every model was trained with the standard adversarial training with Projected Gradient Descent (Madry et al., 2018). We used 7-step PGD with step size  $2/255$  to perturb the input images and limit them into  $\ell_\infty$ -norm ball with maximum radius  $8/255$ . We used the SGD optimizer with momentum 0.9, weight decay 0.0005, and batch size 128. The initial learning rate was 0.1 and decayed by the factor of 0.1 at the epochs 75 and 90. We applied gradient clipping to the parameters to stabilize the training process by enforcing the maximum  $\ell_2$  norm of 5.0 in gradients. Each model was trained for 100 epochs on the full CIFAR-10 training set Krizhevsky (2009), which contains 50K images of size  $32 \times 32$  from 10 classes. The training set was augmented with random cropping and random horizontal flipping. To avoid the Robust Overfitting (Rice et al., 2020) during the later stage of adversarial training, we applied the early stopping strategy by recording the best PGD-CW<sup>40</sup> accuracy on a separate validation set, i.e., CIFAR-10.1 (Recht et al., 2018).

1134  
1135  
1136  
1137  
1138  
1139  
1140  
1141  
1142  
1143  
1144  
1145  
1146  
1147  
1148  
1149  
1150  
1151  
1152  
1153  
1154  
1155  
1156  
1157  
1158  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187

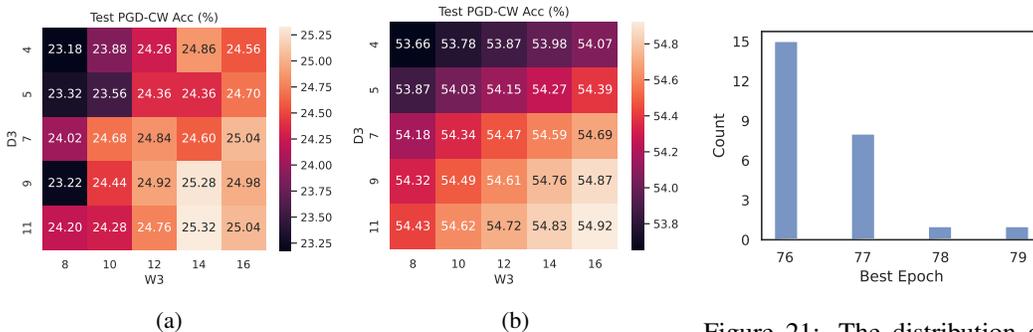


Figure 20: Adversarial robustness of architectures with different last stage settings. **(a)**: Distribution on Tiny-ImageNet under 25 architectures with  $D_{i \in \{1,2\}} = 7, W_{i \in \{1,2\}} = 12$  and  $D_3 \in \{4, 5, 7, 9, 11\}, W_3 \in \{8, 10, 12, 14, 16\}$ . **(b)**: Distribution on CIFAR-10 under 15625 architectures, i.e., the search space of *NARes*.

Figure 21: The distribution of the best epoch on Tiny-ImageNet under 25 architectures with  $D_{i \in \{1,2\}} = 7, W_{i \in \{1,2\}} = 12$  and  $D_3 \in \{4, 5, 7, 9, 11\}, W_3 \in \{8, 10, 12, 14, 16\}$ .

Table 4: Evaluated corruption types on CIFAR-10-C.

Group	Corruption Types
Noise	Gaussian, Impulse, Shot, Speckle
Blur	Defocus, Glass, Gaussian, Motion, Zoom
Weather	Brightness, Fog, Frost, Snow, Spatter
Digital	Contrast, Elastic, JPEG Compression, Pixelate, Saturate

## B.2 EVALUATION DETAILS ON ROBUSTNESS OF CORRUPTIONS

The 19 corruption types in CIFAR-10-C (Hendrycks & Dietterich, 2018) are listed in Table 4, which are classified into four groups. The dataset of CIFAR-10-C is built on the test set of CIFAR-10, and each corruption type contains 5 levels of severity. Overall, there are  $19 * 5 = 95$  accuracies on the common corruptions for each architecture.

## B.3 DECISION CHOICE OF *NARes*

Since this is the first neural network architecture dataset on macro search space for AR, the design choices for *NARes* remain unexplored. We explain some of the decisions on the creation of *NARes*.

**Using CIFAR10.1 as the validation set.** We expect that the models are able to be directly compared with other models in previous studies of adversarial training; therefore, we use the full training set of CIFAR-10 during the adversarial training and utilize a separate validation set, CIFAR-10.1. The images in CIFAR-10.1 are sampled under the same collecting strategy of CIFAR-10 from the same source, and the sub-class distribution is carefully matched (Recht et al., 2018). Although there will still be a performance drop with this validation set due to the slight distribution shift mentioned in the original work of CIFAR-10.1, we believe this validation set can better reflect the real robustness of the model from multiple perspectives. Since the validation set only affects the selection of the best model, i.e., which epoch the early stopping is applied, we can view the validation performance as a pessimistic estimate of the test set with unknown data. To study the impact of this decision, we also retrain 32 models around 10G MACs with 40K training data and 10K validation data split from the original CIFAR-10 training set (denoted as "Split"), which is commonly used in NAS domains, and compare with metrics of the same models in *NARes*. Results in Fig. 22 suggest using CIFAR-10.1 will apply the early stop sooner than "Split"; besides, the loss and clean accuracy in the validation set indicate that overfitting will also happen on CIFAR-10.1, i.e., on datasets with a slight degree of out-of-distribution (OOD). As a result, using CIFAR-10.1 as the validation set will also help prevent OOD overfitting through early stopping.

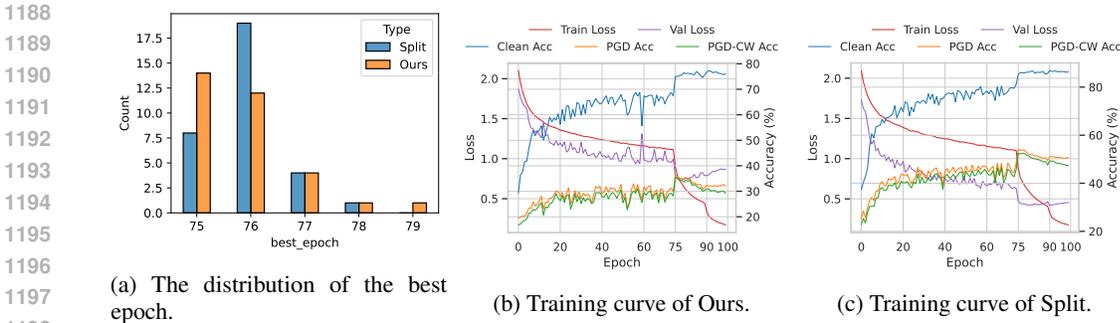


Figure 22: The statistics of NARes and the "Split" training strategy on models with  $\sim 10G$  #MACs. "Split" utilizes 40K training data and 10K validation data split from the original CIFAR-10 training set for the same standard adversarial training.

Table 5: The aggregate adversarial accuracies (from left to right) during AutoAttack on common models in NARes.

Model	APGD-CE	APGD-T	FAB-T	Square
WRN-28-8	50.95	49.03	49.03	49.03
WRN-28-10	51.92	50.06	50.06	50.06
WRN-34-10	52.30	50.21	50.21	50.21
WRN-34-12	52.51	50.51	50.51	50.51
WRN-46-12	53.79	51.99	51.99	51.99
WRN-46-14	54.46	52.60	52.60	52.60
WRN-58-14	54.54	52.75	52.75	52.75
WRN-70-16	54.10	52.25	52.24	52.24

**Not using SuperNet.** Although SuperNet is widely used in NAS domains to reduce training costs, research on whether models in the Supernet under our search space can be generalized on adversarial robustness is scarce. In addition, as mentioned by Madry et al. (2018), the adversarial robustness depends on the complex decision boundary of the model; it is unclear whether the sampled model from the Supernet can also inherit the decision boundary for adversarial robustness. Therefore, we choose to train each architecture from scratch with adversarial training, which eliminates the above concerns. And we hope NARes will help the future development of robust Supernet methods.

**Replacing AA with AA-Compact.** AutoAttack (AA) is widely used for benchmarking adversarial robustness. However, the computation cost of AA is high, so evaluating all models in NARes is expensive. Instead, we choose to use AA-Compact mentioned in Sec. 3.2 as an approximation of AA. We demonstrate the accuracies of some common models in NARes during the aggregate attacking of AA in Table 5. The other two attacks are unlikely to yield any new adversarial examples beyond those produced by APGD-CE and APGD-T, substantiating our decision. This reduces the evaluation time to approximately 1/5 that of the AA.

## C DETAILS ON NARes AS A NAS BENCHMARK

### C.1 IMPLEMENTATIONS OF NAS ALGORITHMS IN SEC. 5

**Regularized Evolution (RE):** We set the population size to 30, and the tournament size to 10. The mutation rate is set to 1.0. After each iteration, the oldest individual is replaced by a new offspring from the mutation of the parent.

**BANANAS:** The implementation and hyperparameters follow its official code<sup>4</sup>. We use five predictors as the ensemble model. Each predictor is a 20-layer feed-forward network with a width of 32. The decision variable of architecture is encoded as a one-hot binary vector with length  $5 * 6 = 30$  as the

<sup>4</sup><https://github.com/naszilla/naszilla>

Table 6: Results of different NAS algorithms on *NARes*. The algorithms search the best architecture based on (a): the clean accuracy, or (b): the PGD-CW<sup>40</sup> accuracy on the validation set, and the mean and the standard variance of robustness metrics on the best architecture are reported over 400 runs.

(a) Search objective: Clean Accuracy					
Accuracy	Optimal*	Random Search	Local Search	RE	BANANAS
Val Clean <sup>†</sup>	78.25	77.66 ± 0.25	77.67 ± 0.24	77.91 ± 0.21	<b>77.95 ± 0.20</b>
Val PGD <sup>20</sup>	38.80	36.15 ± 0.68	36.17 ± 0.68	<b>36.19 ± 0.50</b>	36.17 ± 0.51
Val PGD-CW <sup>40</sup>	37.55	35.35 ± 0.67	35.42 ± 0.65	35.59 ± 0.66	<b>35.69 ± 0.73</b>
Test Clean	88.57	87.92 ± 0.25	87.92 ± 0.26	88.07 ± 0.21	<b>88.09 ± 0.19</b>
Test FGSM	62.68	61.54 ± 0.43	61.55 ± 0.43	61.67 ± 0.43	<b>61.70 ± 0.49</b>
Test PGD <sup>20</sup>	57.39	55.97 ± 0.57	55.98 ± 0.58	56.18 ± 0.56	<b>56.22 ± 0.63</b>
Test PGD-CW <sup>40</sup>	56.17	54.71 ± 0.54	54.72 ± 0.53	54.91 ± 0.50	<b>54.97 ± 0.54</b>
Test AA <sup>‡</sup>	53.48	51.82 ± 0.53	51.83 ± 0.53	52.01 ± 0.51	<b>52.07 ± 0.55</b>
Test Corruption	80.22	79.62 ± 0.26	79.63 ± 0.25	79.78 ± 0.18	<b>79.80 ± 0.16</b>

(b) Search objective: PGD-CW <sup>40</sup> Accuracy					
Accuracy	Optimal*	Random Search	Local Search	RE	BANANAS
Val Clean	78.25	75.96 ± 0.67	75.95 ± 0.63	<b>76.12 ± 0.38</b>	76.09 ± 0.34
Val PGD <sup>20</sup>	38.80	37.69 ± 0.41	37.71 ± 0.43	38.09 ± 0.49	<b>38.17 ± 0.44</b>
Val PGD-CW <sup>40†</sup>	37.55	37.02 ± 0.18	37.06 ± 0.20	37.32 ± 0.20	<b>37.37 ± 0.20</b>
Test Clean	88.57	<b>87.31 ± 0.42</b>	87.29 ± 0.41	87.27 ± 0.32	87.26 ± 0.27
Test FGSM	62.68	61.42 ± 0.34	61.39 ± 0.34	<b>61.46 ± 0.31</b>	<b>61.46 ± 0.31</b>
Test PGD <sup>20</sup>	57.39	56.44 ± 0.30	56.46 ± 0.32	56.66 ± 0.35	<b>56.72 ± 0.35</b>
Test PGD-CW <sup>40</sup>	56.17	55.03 ± 0.33	55.04 ± 0.34	<b>55.20 ± 0.34</b>	<b>55.20 ± 0.33</b>
Test AA <sup>‡</sup>	53.48	52.34 ± 0.34	52.35 ± 0.36	52.58 ± 0.37	<b>52.61 ± 0.37</b>
Test Corruption	80.22	<b>79.01 ± 0.42</b>	78.98 ± 0.39	78.86 ± 0.36	78.79 ± 0.32

\* : "Optimal" refers to the highest achievable accuracy in the dataset of *NARes*.

† : The objective for NAS.

‡ : We use AA-Compact, a compact version of AA.

input of predictors, and the output of the predictor is the predicted objective value for that architecture. The ensemble model is trained from scratch with 500 epochs at each iteration.

## C.2 NAS BENCHMARK RESULTS ON OTHER OBJECTIVES

Besides the NAS benchmark on the objective of validation PGD<sup>20</sup> accuracy in Table 2, we also test the NAS algorithms with the objective of validation clean accuracy and PGD-CW<sup>40</sup> accuracy. The results are shown in Table 6. In either case, BANANAS achieves the best search efficiency and stability on the objective. Moreover, compared to other AR metrics, searching by clean accuracy makes it difficult for NAS algorithms to achieve higher AR; meanwhile, it makes searching easier to find architectures with higher clean accuracy and robustness of common corruption.