APPENDIX: ON THE RELATIONSHIP BETWEEN ADVERSARIAL ROBUSTNESS AND DECISION REGION IN DEEP NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

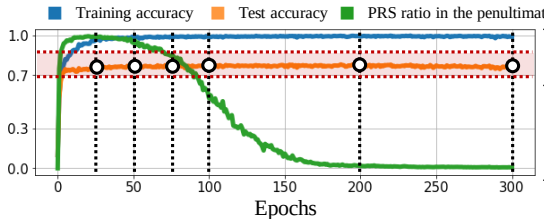
A EXPERIMENTAL SETUP

In this section, we describe the details to build the model candidates used in Figure 4 in the main paper and Figure 17 in the Appendix. We first select three different structures; (1) convolutional neural network with 6 convolutional blocks (CNN-6), (2) VGG-16, and (3) ResNet-18 and three datasets; (1) MNIST, (2) F-MNIST, and (3) CIFAR-10. In particular, we use bilinear upsampled MNIST and F-MNIST dataset ($28 \times 28 \rightarrow 32 \times 32$). The detailed structure of CNN-6 is described in Table 1. For VGG-16 and ResNet-18, we maintain the original structure[1] for the feature extraction and replace the classifier as same as CNN-6.

Table 1: The architecture of CNN-6.

	Type	Channels/size	Activation
Convolutional Layer	Conv2d [3×3]	32	ReLU
	Conv2d [3×3], MaxPool(2)	64	
	Conv2d [3×3]	128	
	Conv2d [3×3], MaxPool(2)	128	
	Conv2d [3×3], MaxPool(2)	64	
	Conv2d [3×3]	64	
Classifier	Linear	512	ReLU
		128	
		10	

We train[2] basic models with fixed five random seeds (123, 375, 574, 907 and 981) and four batch sizes (64, 128, 512 and 2048). Finally we obtain 20 basic models for each dataset and structure. For



Model	Seed	BS	MNIST 25	50	75	100	200	300
CNN-6	375	64	O	O	O	O	O	O
		128	O	O	O	O	O	O
		512	O	O	O	O	O	O
		2048	O	O	O	O	O	O

Part of Table

Figure 14: An illustrative example for the selected candidates and the corresponding part of Table 2. The example corresponds to the CNN-6 with random seed 375, batch size (BS) 128 and MNIST.

the extensive analysis on the correlation between PRS ratio and properties of network, we extract candidates from each basic model with the grid of epochs ($[25, 50, 75, 100, 200, 300]$). Then we apply the test accuracy ($acc.$) threshold (MNIST: $98\% \leq acc.$, F-MNIST: $90\% \leq acc. \leq 93\%$, and CIFAR-10: $72\% \leq acc. \leq 78\%$) to guarantee the sufficient performance. Figure 14 presents the illustrative example for candidates selection. Table 2 presents used candidates for the experi-

[1] We use the officially provided network from the Pytorch: https://pytorch.org/vision/stable/models.html
[2] Cross-entropy loss and Adam optimizer with learning rate 10^{-3} is used.

ments. We also provide the statistics of test accuracy for the candidates over each selected epoch in Appendix B.

Table 2: Used candidates of models for all experiments in the main paper. Pink box indicates the models which does not satisfy the described condition. The orange/red box indicates network A/B denoted in the main paper respectively. The blue box is described example in Appendix B. BS indicates the batch size.

			MNIST						F-MNIST						CIFAR10					
Model	Seed	BS	25	50	75	100	200	300	25	50	75	100	200	300	25	50	75	100	200	300
CNN-6	123	64	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	x
		128	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	x	o	o
		512	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
		2048	o	o	o	o	o	o	o	x	o	o	o	o	x	x	x	x	o	o
	375	64	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	x
		128	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
		512	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
		2048	o	o	o	o	o	o	o	o	o	o	o	o	x	x	x	o	o	o
	574	64	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	x
		128	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	x	o
		512	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
		2048	o	o	o	o	o	o	o	o	o	o	o	o	x	x	o	o	o	o
	907	64	o	o	o	o	o	o	o	o	o	o	o	x	o	o	o	o	o	x
		128	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	x	o	o
		512	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
		2048	o	o	o	o	o	o	o	o	o	o	o	o	x	x	x	o	o	o
	981	64	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	x
		128	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
		512	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
		2048	o	o	o	o	o	o	o	o	o	o	o	o	x	x	o	o	o	o
ResNet-18	123	64	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
		128	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
		512	o	o	o	o	o	o	x	o	o	o	o	o	o	o	o	o	o	o
		2048	o	o	o	o	o	o	x	o	o	o	o	o	x	x	x	x	x	o
	375	64	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
		128	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
		512	o	o	o	o	o	o	x	o	o	o	o	o	x	o	o	o	o	o
		2048	o	o	o	o	o	o	x	o	o	o	o	o	x	x	x	x	o	x
	574	64	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
		128	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
		512	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
		2048	o	o	o	o	o	o	x	o	o	o	o	o	x	x	x	x	x	o
	907	64	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
		128	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
		512	o	o	o	o	o	o	x	o	o	o	o	o	o	o	o	o	o	o
		2048	o	o	o	o	o	o	x	x	o	o	o	o	x	x	x	x	o	o
	981	64	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
		128	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
		512	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o
		2048	o	o	o	o	o	o	x	o	o	o	o	o	x	x	x	x	x	o
VGG-16	123	64	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	x
		128	o	o	o	o	o	o	o	o	o	o	o	o	o	x	x	x	x	x
		512	o	o	o	o	o	o	o	o	o	o	x	x	o	o	o	x	x	x
		2048	o	o	o	o	o	o	o	o	o	o	o	o	x	x	o	o	o	o
	375	64	o	x	o	o	o	o	o	o	o	o	o	x	o	o	o	x	x	x
		128	o	o	o	o	o	o	o	o	o	o	o	x	x	o	o	o	o	x
		512	o	o	o	o	o	o	o	o	o	o	o	o	x	o	o	o	o	x
		2048	o	o	o	o	o	o	o	o	o	o	o	o	x	x	x	x	x	x
	574	64	o	o	o	o	o	o	o	o	o	o	o	o	x	o	o	o	o	x
		128	o	o	o	o	o	x	o	o	o	o	x	o	o	o	o	o	o	x
		512	o	o	o	o	o	o	o	o	o	o	o	o	x	x	x	x	o	o
		2048	o	o	o	o	o	o	o	o	o	o	o	o	x	x	x	x	x	o
	907	64	o	o	o	o	o	o	o	o	o	x	o	o	o	o	o	o	x	x
		128	o	o	o	o	o	o	o	o	o	o	o	x	x	x	o	x	x	x
		512	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	x	x	x
		2048	x	x	o	o	o	o	o	o	o	o	o	o	x	o	o	o	o	o
	981	64	o	o	o	x	o	x	o	o	o	o	x	o	x	o	x	x	x	x
		128	o	o	o	o	o	o	o	o	o	o	o	o	o	x	x	x	x	x
		512	o	o	o	o	o	o	o	o	o	o	o	o	o	o	o	x	x	x
		2048	o	o	o	o	o	o	o	o	o	o	o	o	x	o	o	o	o	o

B MODEL PERFORMANCE

In this section, we provide the statistics of test accuracy for the selected candidates in Table 3. We average the test accuracy for random seeds. For example, the mean and standard deviation in the 25th epoch and batch size (BS) 64 of VGG-16 on MNIST are calculated from blue boxes in Table 2. Dash line indicates that there are no models satisfied with the described conditions in random seeds.

Table 3: Test accuracy (%) for candidates over random seeds.

		Epoch	25	50	75	100	200	300
		BS	Test Accuracy (%)					
MNIST	CNN-6	64	99.21±0.13	99.13±0.12	99.25±0.10	99.34±0.05	99.23±0.22	99.21±0.15
		128	99.18±0.09	99.21±0.08	99.37±0.08	99.36±0.05	99.43±0.05	99.44±0.05
		512	99.05±0.18	99.21±0.09	99.20±0.22	99.40±0.05	99.42±0.06	99.42±0.03
		2048	98.88±0.16	99.06±0.11	99.04±0.12	99.05±0.11	99.29±0.08	99.30±0.08
	VGG-16	64	99.11±0.31	99.06±0.43	99.21±0.39	99.05±0.55	99.28±0.18	99.31±0.20
		128	99.20±0.34	99.22±0.19	99.22±0.30	99.35±0.15	99.25±0.31	99.26±0.37
		512	99.20±0.12	99.34±0.17	99.36±0.13	99.46±0.09	99.47±0.06	99.46±0.07
		2048	99.10±0.24	99.27±0.06	99.30±0.08	99.21±0.39	99.36±0.12	99.35±0.12
	ResNet-18	64	99.03±0.12	99.24±0.05	99.28±0.05	99.30±0.08	99.30±0.08	99.30±0.10
		128	99.03±0.22	99.15±0.08	99.32±0.05	99.34±0.06	99.34±0.06	99.35±0.06
		512	98.89±0.19	99.13±0.13	99.27±0.19	99.42±0.03	99.44±0.03	99.44±0.03
		2048	98.76±0.34	99.20±0.17	99.34±0.06	99.35±0.06	99.35±0.06	99.35±0.06
F-MNIST	CNN-6	64	91.72±0.37	91.81±0.23	91.83±0.20	91.97±0.17	91.80±0.31	90.82±0.35
		128	91.52±0.19	91.77±0.28	91.95±0.19	92.04±0.26	92.16±0.14	91.73±0.82
		512	91.05±0.24	91.04±0.39	91.52±0.55	91.71±0.27	91.92±0.28	92.49±0.20
		2048	90.43±0.29	90.98±0.23	90.87±0.39	91.06±0.33	91.30±0.24	91.58±0.28
	VGG-16	64	91.92±0.22	92.30±0.65	92.67±0.17	92.11±0.27	92.63±0.23	92.41±0.48
		128	92.14±0.40	92.54±0.44	92.57±0.16	92.71±0.16	92.81±0.17	92.91±0.05
		512	91.74±0.48	92.08±0.25	92.24±0.25	92.62±0.32	92.82±0.05	92.80±0.06
		2048	91.09±0.45	92.03±0.33	92.30±0.25	92.39±0.33	92.50±0.07	92.57±0.29
	ResNet-18	64	90.60±0.24	90.96±0.28	91.02±0.12	91.05±0.15	91.04±0.17	91.01±0.44
		128	90.47±0.22	90.81±0.31	91.00±0.21	91.08±0.23	91.13±0.31	91.23±0.29
		512	90.27±0.25	90.53±0.21	90.57±0.19	90.69±0.23	90.94±0.39	91.19±0.25
		2048	-	90.17±0.23	90.56±0.26	90.41±0.13	90.61±0.33	90.49±0.23
CIFAR-10	CNN-6	64	75.76±1.32	76.39±1.06	77.28±0.45	76.88±0.45	75.76±0.51	-
		128	76.54±0.80	76.88±0.55	76.32±0.41	77.18±0.22	76.42±0.63	76.49±0.87
		512	73.17± 0.97	74.69± 1.17	74.55± 1.49	75.18±0.86	75.35±0.90	75.79±1.11
		2048	-	-	72.79±0.32	73.73±0.56	74.52±0.93	74.59±0.95
	VGG-16	64	74.38±0.96	76.10±1.06	76.73±0.54	77.37±0.61	77.86±0.11	-
		128	75.59±1.90	74.53±0.52	76.44±1.47	76.18±1.05	77.00±1.20	-
		512	75.82±1.01	76.58±1.24	76.98±0.93	76.23±0.00	75.16±1.72	74.87±0.00
		2048	-	74.37±0.71	73.74±1.58	74.94±1.01	75.79±0.77	76.26±1.34
	ResNet-18	64	75.71±0.39	75.98±0.41	76.41±0.75	76.44±0.15	76.83±0.25	76.30±0.52
		128	74.79±0.53	75.75±0.56	76.35± 0.66	76.18±0.52	76.39±0.69	76.32±0.19
		512	72.63±0.45	73.44±0.73	74.20±0.50	74.73±0.61	75.16±0.14	75.38±0.93
		2048	-	-	-	-	72.08±0.00	72.53±0.62

C FAILED ATTACK EXAMPLES

In this section, we provide more failed attack examples described in Figure 5 in the main paper. We note that the examples are only attacked successfully on Network A.

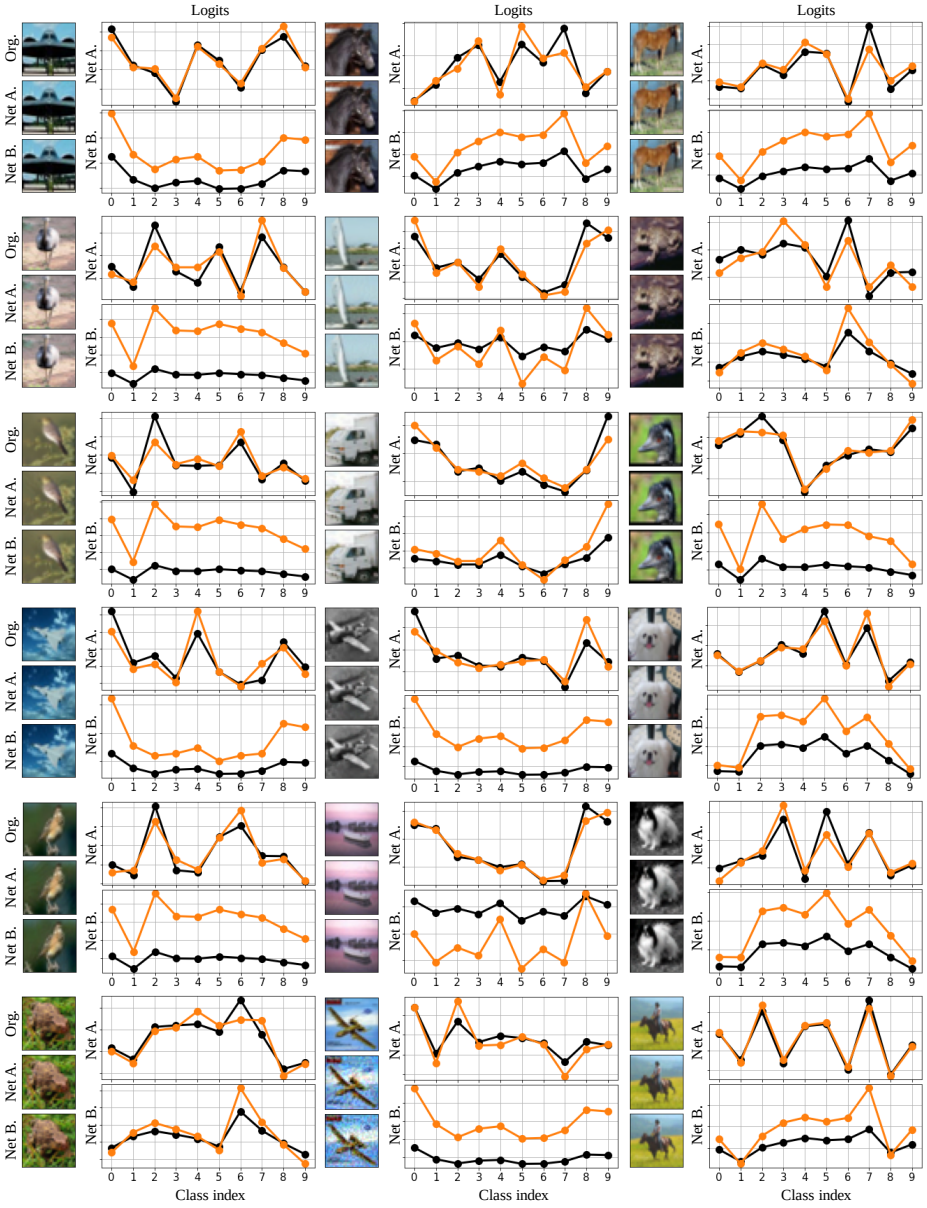


Figure 15: Randomly selected examples for successful attack on Network A but failed attack on Network B. The black line is the original predicted logits and the orange line is the changed logits after attack.

D PRS AND TRAINED FEATURES

D.1 FEATURE SPARSITY

This section explores the trained features of the models with the different PRS ratios. First, we visualize the feature maps directly for each depth of layers. Figure 16 shows illustrative examples of feature maps. We identify that the model with the low PRS ratio learns more sparse features compared to the model with the high PRS ratio.

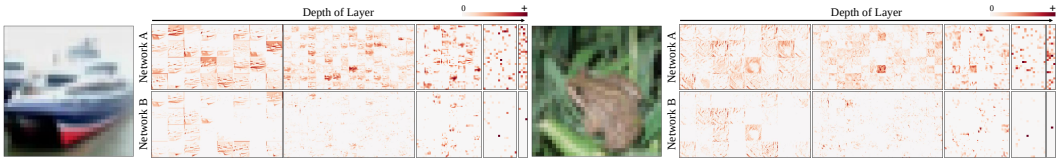


Figure 16: Visualization of feature maps for CNN-6 trained on CIFAR-10. The first row represents the feature maps for Network A and the second row represents Feature maps for Network B.

As the sparse features are considered as an independent and informative representation (Lee et al., 2007; Ranzato et al., 2007), if the PRS ratio can cause sparse feature representation in various cases, we conclude that the PRS ratio is related to the informative features. To verify our hypothesis, we measure the trend between the PRS ratio and the average sparsity for each network. The average sparsity of the model is calculated by taking the average of the ratio of zero-valued features over the training dataset for all layers.

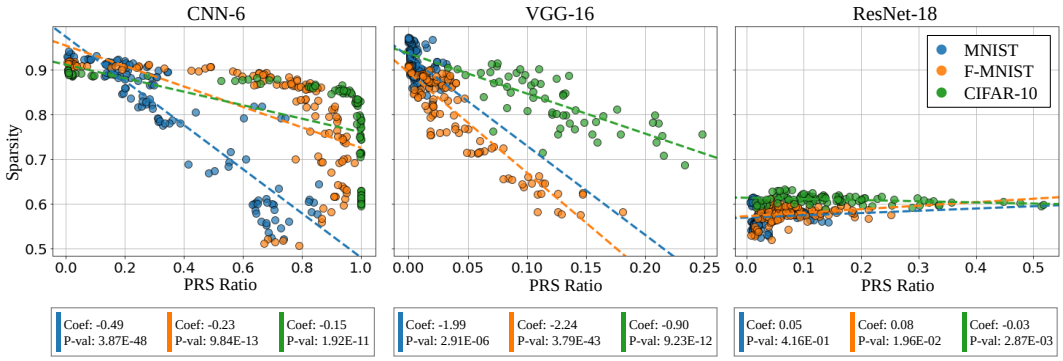


Figure 17: Relationship between the PRS ratio and the average sparsity for different networks on the various datasets. The colored dashed lines indicate the trend for each dataset.

Figure 17 shows the relationship between the PRS ratio and the average sparsity in all cases. We identify that CNN-6 and VGG-16 show an inversely correlated relationship throughout the dataset, but it is difficult to find a clear relationship for ResNet-18. We conjecture that skip connection can cause this phenomenon.

D.2 FEATURE DENSENESS

To quantify the denseness of feature representation which explains how much the samples from the same class are grouped in the feature space on the penultimate layer, we measure the ratio of same labels between K nearest neighbor samples for the given input. Let the training dataset $\mathbf{X}$ and data pair $(x_i, y_i) \in \mathbf{X}$.
The denseness for the given data pair (x, y) is defined as,

$$D(\mathbf{X}, K) = \frac{1}{|\mathbf{X}|} \sum_{x,y \in \mathbf{X}} \frac{1}{K} \sum_{i \in \mathrm{NN}_K(x, \mathbf{X})} \mathbf{I}(y_i = y),$$

where $\text{NN}_K(x, \mathbf{X})$ returns indices of K nearest neighbors for feature representation of the given input $f_{L-1:1}(x)$ from training dataset $\mathbf{X}$ and $\mathbf{I}(\cdot)$ is the indicator function. Table 4 presents the average denseness for various random seeds mentioned in Appendix A with various K. The experiment is performed on CNN-6 with a low/high PRS ratio[3]. We identify that low PRS ratio cases (a1, b1, c1) have higher denseness compared to the high PRS ratio cases in almost K. For MNIST dataset, it is difficult to discriminate the difference of denseness along the magnitude of PRS ratio.

Table 4: Average of denseness (%) in feature space for models with a low PRS ratio and a high PRS ratio. We trained CNN-6 model on three datasets, (a) MNIST, (b) F-MNIST and (c) CIFAR-10. Odd rows (a1, b1, c1) indicate the model with a low PRS ratio and even rows (a2, b2, c2) indicate the model with a high PRS ratio.

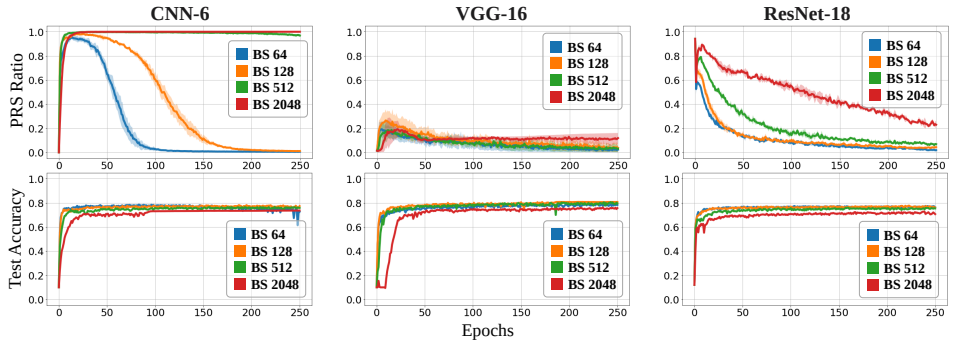
		K				
	PRS Ratio	10	50	100	200	300
(a1)	0.035±0.029	**99.9±0.1**	**99.8±0.1**	99.7±0.1	99.6±0.2	99.5±0.2
(a2)	0.669±0.088	99.9±0.1	**99.8±0.1**	**99.8±0.1**	**99.7±0.2**	**99.6±0.2**
(b1)	0.022±0.007	**98.1±0.6**	**97.2±0.8**	**96.6±0.9**	**95.7±1.0**	**94.9±1.1**
(b2)	0.946±0.018	96.9±0.3	95.5±0.3	94.8±0.4	93.8±0.4	93.2±0.4
(c1)	0.006±0.000	**94.5±0.6**	**92.0±0.7**	**90.3±0.7**	**87.7±0.7**	**85.7±0.7**
(c2)	1.000±0.000	93.9±0.7	90.4±0.8	88.3±0.9	85.5±1.0	83.6±1.1

[3]We use trained model with batch size of 128 and 2048 to control the PRS ratio

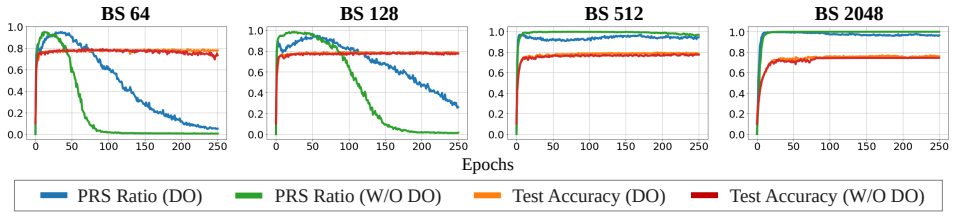
E THE FACTORS WHICH AFFECT THE PRS RATIO

We perform an ablation study on the factors which affects the PRS ratio. First, we identify the relationship between the PRS ratio and batch size (BS) which is one of the factors we used to make the candidates in section A. Fig. 18 (a) presents the PRS ratio and test accuracy for training epochs in different BS (64, 128, 512, 2048) and different networks (CNN-6, VGG-16, ResNet-18) on CIFAR-10. In Fig. 18 (a), we observe that BS is proportional to the PRS ratio (i.e., The large BS causes the high PRS ratio). Previous work (Yao et al., 2018) provides that training with large batch size can degrade the robustness of the model against the adversarial attack, which is aligned with our observation.

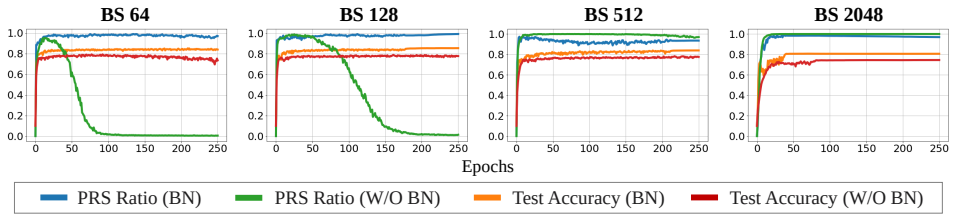
To further investigate other factors, we select two training techniques used in general: (1) Drop out (DO), and (2) Batch normalization (BN). In order to minimize the other influences such as the structural characteristics (e.g., Skip Connection of ResNet), we adopt CNN-6 as the base model in ablation study. Fig. 18 (b) presents the PRS ratio and test accuracy for training epochs according to the existence of DO ($p = 0.2$) over various BS on CIFAR-10. We find that DO tends to delay the decrease of PRS ratio in the cases of BS 64 and 128. However, if the network has high PRS ratio (BS 512 and 2048), DO does not affect to change of the PRS ratio. Fig. 18 (c) represents the result of the training with the existence of BN for various BS on CIFAR-10. We find that the models with BN have the high PRS ratio in all cases. When we consider the relationship between the PRS ratio and robustness, BN can be considered as the factor which causes adversarial vulnerability. Previous work (Benz et al., 2021) describes the negative effect of BN on the robustness.



(a) Influence of batch size for the PRS ratio in the training with CIFAR-10.



(b) Influence of drop out for the PRS Ratio in the training with CIFAR-10.



(c) Influence of batch normalization for the PRS Ratio in the training with CIFAR-10.

Figure 18: The ablation study for the factors which affect to the PRS ratio.

F ROBUST ACCURACY FOR VARIOUS ATTACKS

F.1 PRS RATIO AND ROBUSTNESS FOR ANOTHER ATTACKS

We measure the robust accuracy under the FGSM attack (Goodfellow et al., 2014) and Auto Attack(AA) (Croce & Hein, 2020) on two networks with different PRS ratio. The models are selected at the 200th epoch with random seed 907, except for VGG-16 on CIFAR-10 (the 75th epoch) to guarantee the performance. Table 5 presents the robust accuracy for various attacks. We identify that the network with low PRS ratio is more robust than higher one in almost attack methods. As can be seen from the AA results for "Standardly trained model" in RobustBench[4], the models show zero robust accuracy against AA known as the most powerful adversarial attack. In our experiment, as entire candidates in Table 2 of Appendix A are standardly trained, the robust accuracy under AA is mostly close to zero. As a result, it is not comparable the trend of relationship between PRS ratio and robustness under AA. As an alternative, we provide the experiment for various defense methods under AA in Appendix F.2.

Table 5: Robust accuracy for various attack methods (Higher is better). The magnitude of ϵ is as follow: MNIST = 0.3, F-MNIST = 0.1, and CIFAR10 = 0.0313 on L_∞ norm. We denote standard test accuracy by Clean.

Model	Dataset	BS	PRS Ratio	Clean	FGSM	PGD	AA
CNN-6	MNIST	128	0.201	99.46	**83.54**	**39.79**	0.00
		2048	0.679	99.39	37.11	11.81	0.00
	fMNIST	128	0.195	92.24	**52.63**	**34.05**	0.00
		2048	0.977	91.05	28.39	2.67	0.00
	CIFAR10	128	0.022	77.60	**44.71**	**41.17**	**0.39**
		2048	1.000	75.66	40.10	25.92	0.00
VGG-16	MNIST	128	0.001	99.18	54.22	36.36	0.00
		2048	0.024	99.35	**55.59**	**36.68**	0.00
	fMNIST	128	0.012	92.66	**45.09**	**30.60**	**0.84**
		2048	0.060	92.44	41.31	17.14	0.00
	CIFAR10	128	0.012	77.90	**44.40**	**33.68**	**0.08**
		2048	0.106	75.05	39.92	30.03	0.00
ResNet-18	MNIST	128	0.013	99.34	**51.15**	**24.88**	0.00
		2048	0.078	99.37	0.40	0.00	0.00
	fMNIST	128	0.027	90.75	**31.56**	**19.94**	**0.60**
		2048	0.078	90.29	26.55	12.98	0.00
	CIFAR10	128	0.073	75.40	**40.27**	**29.87**	**0.85**
		2048	0.334	72.08	32.61	24.51	0.00

F.2 PRS REGULARIZER WITH A LARGE DATASET

We provide the comparison of robust accuracy for each regularizer under PGD, Square Attack (Andriushchenko et al., 2020) and Auto Attack on CIFAR-100. We also consider ShuffleNet-V2 (Ma et al., 2018) which is a new architecture to validate consistency of our proposed method. In Table 6, we identify that proposed regularizer can improve the robust accuracy consistently compared to the standard training. We also note that our PRS regularizer can improve the robust accuracy without adversarial examples which require the expensive computation cost. Although L_{PRS} cannot beat the adversarial training (AT), We also confirm that L_{PRS}+AT still shows better results than AT to alleviate the drop of clean accuracy.

[4]RobustBench is to track the real progress in adversarial robustness: https://robustbench.github.io/

Table 6: Comparison of robust and test (clean) accuracy on ShuffleNet-V2 under PGD, Square Attack (SA) and AutoAttack (AA) on $L_\infty(\epsilon$=8/255) for CIFAR-100.

Model	Type	Clean	PGD_{20}	SA	AA	Time/Epoch (s)
ShuffleNet V2	Standard	66.17	2.52	0.37	0.21	31.27
	AT	61.36	36.23	34.72	30.64	134.22 (+102.95)
	$\mathcal{L}_{MR}$	**67.04**	16.22	14.15	12.62	33.27 (+2.00)
	$\mathcal{L}_{PRS}$	66.03	15.12	14.18	12.40	33.34 (+2.07)
	$\mathcal{L}_{MR}$+AT	64.42	**41.76**	40.48	34.36	143.58 (+112.31)
	$\mathcal{L}_{PRS}$+AT	65.82	41.56	**41.07**	**35.00**	145.05 (+113.78)

G ADVERSARIAL TRAINING AND PRS RATIO

To verify the relationship between adversarial training (AT) and the PRS ratio, we perform the comparison between standard training (ST) and AT.

We first train the ResNet-18 with CIFAR-10 dataset using standard training as the pre-trained model (0th - 150th epoch). We then train two networks from this pre-trained model using (1) standard training, and (2) AT based on the PGD on L_∞ with $\epsilon = 0.0313$. After training, we compare the PRS ratio and the robust accuracy against PGD attack. Fig. 19 presents the PRS ratio and the robust accuracy in the training. We observe that the PRS ratio drops and the robust accuracy increases at the same epoch (160th epoch), while those of ST almost maintain.

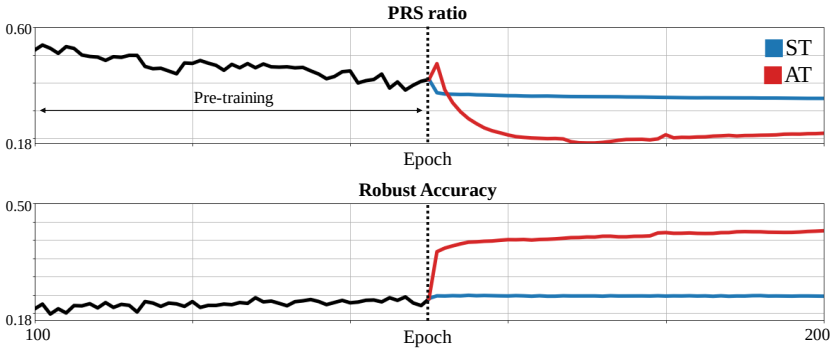


Figure 19: PRS ratio and robust accuracy for two training schemes. Black line indicates the pre-trained network. Red/Blue line indicates standard training and adversarial training, respectively.

H ROBUSTNESS FOR INCLUSION/EXCLUSION TEST SAMPLES

In this section, we provide additional robustness evaluation for different networks and datasets mentioned in Section 3.3.

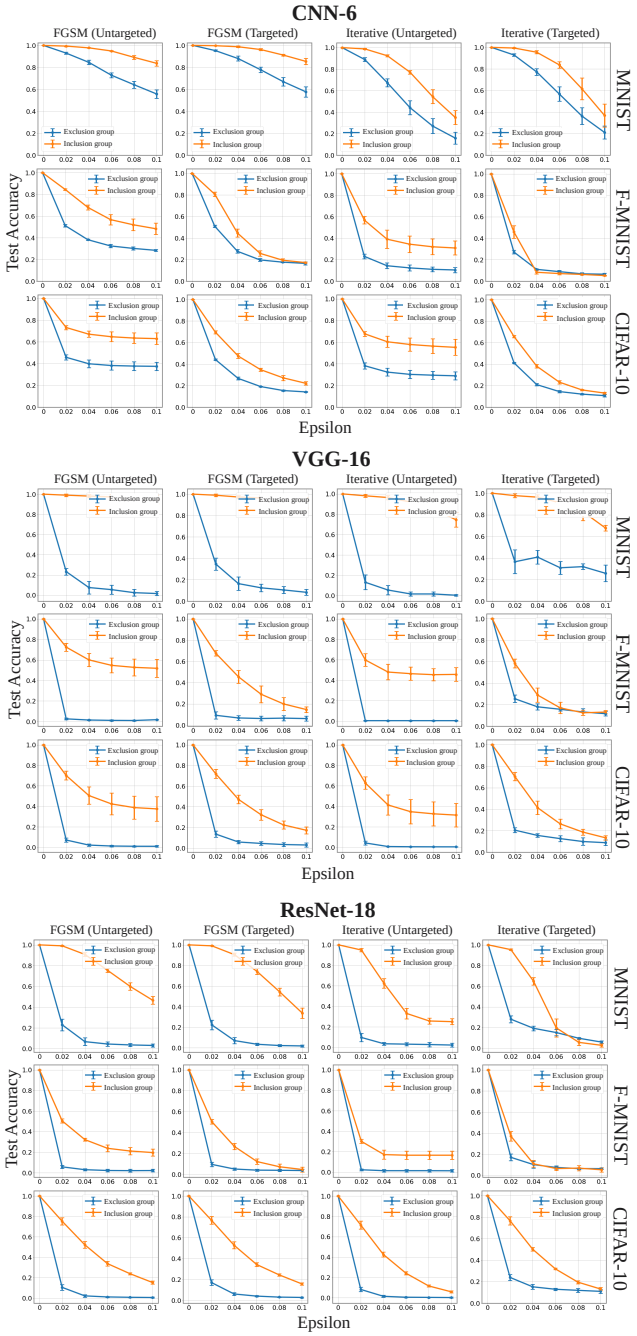
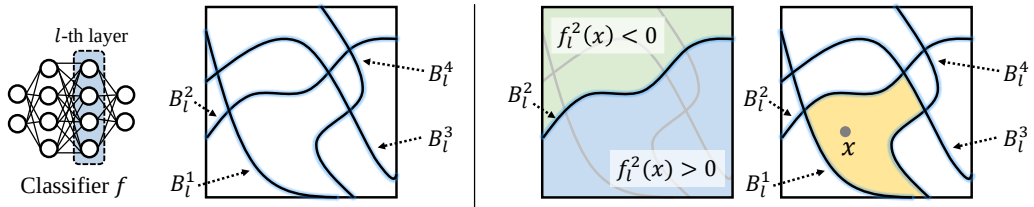


Figure 20: Test accuracy under adversarial attacks on L_∞ norm for exclusion/inclusion group on CNN-6, VGG-16, and ResNet-18. The blue/orange line shows the exclusion/inclusion group.

I AN ILLUSTRATIVE EXAMPLE FOR INTERNAL DBS, DRS AND PRS

In this section, we provide an illustrative example to clarify the concept of internal decision boundary (DB) and decision region (DR) described in the main paper. For the given classifier f, we can consider activation values for each internal neuron in l-th layer. According to the definition of internal DB, we can connect points which have zero-value activation in the input space. In Figure 21 (a), we can identify that the internal DBs in the input space. We note that as the l-th layer has four internal neurons, the network represent four internal DBs in the input space. Figure 21 (b) shows the half-space of one internal DB B_l^2 and internal DR comprised of the intersection of each half-space (yellow shaded area). If the given training sample x resides in this DR, we can call it as popuplated region (PR).

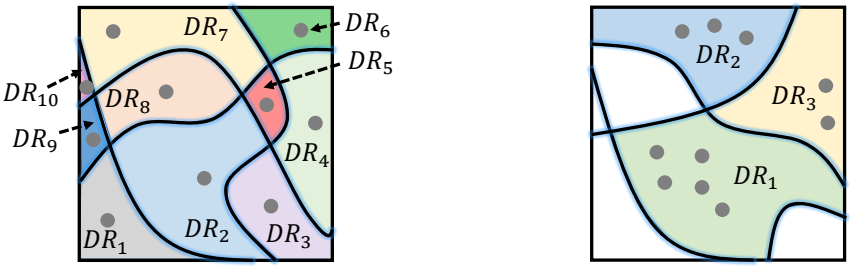


(a) Internal Decision Boundary (B_l^i) for $f_{l:1}$

(b) Decision Region for $f_{l:1}(x)$

Figure 21: An illustrative example for internal DB, DR and PRS in 2-dimensional input space.

Figure 22 shows the conceptual PRS configuration for Network A and B which discussed in the main paper. We note that Network B has lower PRS ratio than Network A. In this conceptual visualization, the PRS of Network B can be represented as $PRS(\mathbf{X}, f, l) = \{DR_1, DR_2, DR_3\}$, while $PRS(\mathbf{X}, f, l) = \{DR_1, \cdots, DR_{10}\}$ on Network A. If we suppose $|X| = 10$, the PRS ratio of Network A is 1 ($10/|X|$), and PRS ratio of Network B is 0.3 ($3/|X|$).



Network A

Network B

Figure 22: An illustrative comparison of PRS between two networks given dataset X. Each gray dot represents a data x for given dataset X.

J A TOY EXAMPLE FOR PRS REGULARIZER

In this section, we provide a 2D binary classification example with standard training and PRS regularizer L_{PRS}. The adversarial robustness is generally interpreted by the concept of margin in the deep neural networks (Sokolić et al., 2017; Madry et al., 2018; Ilyas et al., 2019). We conjecture that our PRS ratio is related to the concept of margin. In the bounded input space, increasing the distance to internal DBs expands the volume of internal DRs, and it eventually reduces the number of DRs. To verify our hypothesis, we provide a 2D binary classification toy example with a simple fully-connected ReLU network (2-200-200-2) with standard training and L_{PRS} in Figure 23. From Figure 23, we identify that L_{PRS} merges PRs directly, and this aligns with the behavior leading to the increase of margin to internal DBs. We believe that this empirical observation can be a bridge to connect the concept of margin and PRS.

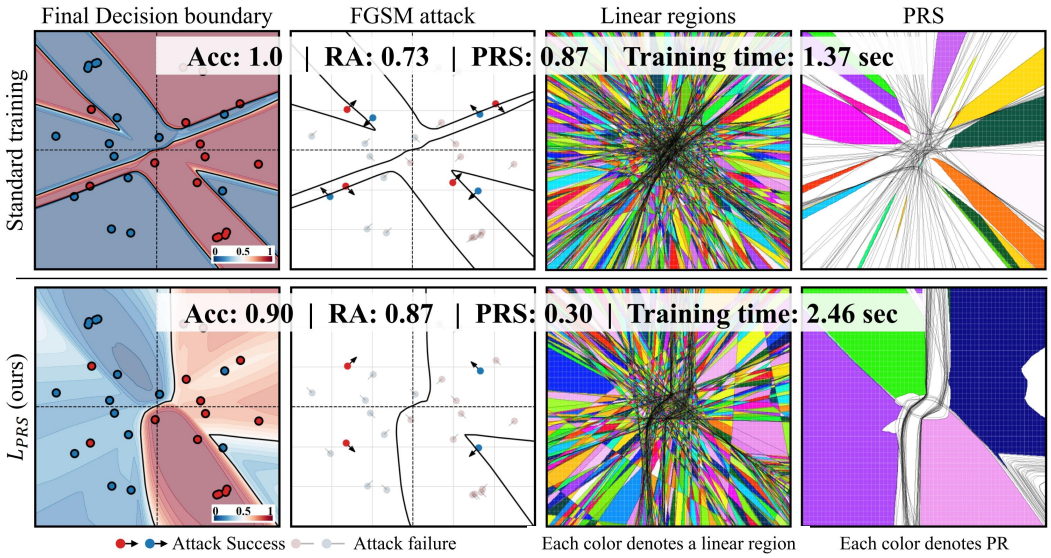


Figure 23: 2D binary classification example with a simple fully-connected ReLU network (2-200-200-2) with Standard training and PRS regularizer.

REFERENCES

Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *European Conference on Computer Vision*, pp. 484–501. Springer, 2020.

Philipp Benz, Chaoning Zhang, and In So Kweon. Batch normalization increases adversarial vulnerability and decreases adversarial transferability: A non-robust feature perspective. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7818–7827, 2021.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pp. 2206–2216. PMLR, 2020.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *arXiv preprint arXiv:1905.02175*, 2019.

Honglak Lee, Chaitanya Ekanadham, and Andrew Ng. Sparse deep belief net model for visual area v2. *Advances in neural information processing systems*, 20:873–880, 2007.

Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 116–131, 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.

Marc Ranzato, Christopher Poultney, Sumit Chopra, Yann LeCun, et al. Efficient learning of sparse representations with an energy-based model. *Advances in neural information processing systems*, 19:1137, 2007.

Jure Sokolić, Raja Giryes, Guillermo Sapiro, and Miguel RD Rodrigues. Robust large margin deep neural networks. *IEEE Transactions on Signal Processing*, 65(16):4265–4280, 2017.

Zhewei Yao, Amir Gholami, Qi Lei, Kurt Keutzer, and Michael W Mahoney. Hessian-based analysis of large batch training and robustness to adversaries. *arXiv preprint arXiv:1802.08241*, 2018.