

A. Performance of Baseline with Further Fine-tuning

We investigate the baseline performance with extended fine-tuning. Figure 3 shows our baseline training, validation, and test accuracy over 300 epochs for ResNet-34 on the CIFAR-100 dataset. We find that the performance plateaued at around epoch 60, and fine-tuning beyond 100 epochs did not yield further performance improvements.

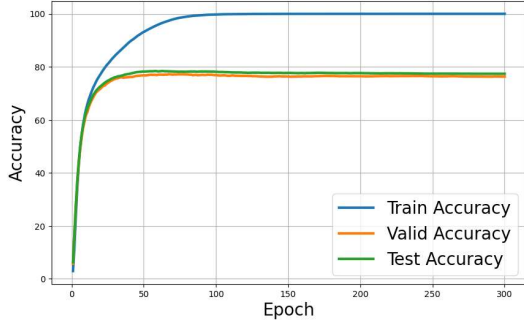


Figure 3. This graph shows our baseline training, validation, and test accuracy over 300 epochs for the ResNet-34 model on the CIFAR-100 dataset. The accuracy results are averaged over three random seeds.

B. Performance of Adversarial Correction (Extension)

B.1. Adversarial Correction of ResNets

We also apply another variant of BI method, stated below: **Targeted Variant of Basic Iterative (VBI)**: In addition to the standard untargeted BI method, we created a targeted variant called VBI. Unlike BI (Eq. 1), which moves away from the true label, VBI (Eq. 7) operates in the opposite direction, moving towards the true label by negating the sign of the gradient sign function.

$$\mathbf{X}_{N+1}^{\text{VBI}} = \text{Clip}_{\mathbf{X}, \epsilon} \left\{ \mathbf{X}_N^{\text{VBI}} - \alpha \text{sign}(\nabla_{\mathbf{X}} J(\mathbf{X}_N^{\text{VBI}}, y_{\text{true}})) \right\}. \quad (7)$$

The comprehensive performance results of our pipeline on FP32 models are presented in Tab. 5, an extended version of Tab. 1. Additionally, the complete performance results of our pipeline on quantized Int8 models can be found in Tab. 6, which is an extended version of Tab. 3. The attacks all seem to have similar performance improvements, except for some directed attacks where we use different directions to reduce the effectiveness of incorrect classifications versus correct ones. Specifically, we can move away along the gradient of the highest probability class of incorrect samples to weaken the accuracy of the incorrect input (*i.e.*, BIH attack) or move toward the true labels (*i.e.*, VBI attack).

Moving toward the true labels yields higher attack correction rates but similar performance compared to moving away along the gradient of the highest probability class. Table 6 also shows that increasing the number of attack iterations to increase the attack success rate for the VBI attack does not correlate closely with the correction rate. However, gradient-based attacks, such as DDN and VBI, have better overall performance than non-gradient-based attacks, such as salt and pepper.

The adversarial correction does indeed reduce the training loss. In Fig. 4b and Fig. 4d we can see that both targeted (VBI) and untargeted (LL) adversarial attacks can successfully reduce the logit level of the initially maximum probability incorrect label as compared with the logit level of the true label, resulting in correction.

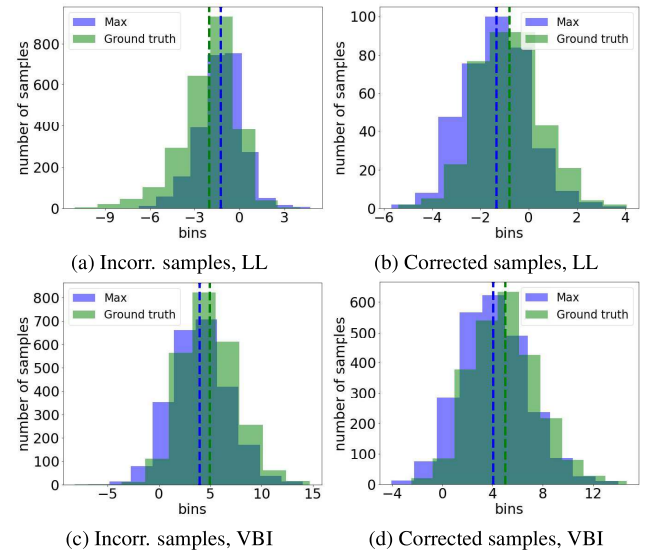


Figure 4. The incorrect class (max) and true class logits change for uncorrected (a,c) and corrected (b,d) samples of CIFAR-100 after applying the corrective LL (a,b) and VBI (c,d) attacks on the ResNet-34. The vertical dashed lines indicate the mean values of incorrect class (max logit) and true class logits change.

B.2. Adversarial Correction vs. Adversarial Perturbation

In the Feng and Tu theory [13], all that is needed in the first step of the input space training (IST) is to perturb the input so as to reduce the loss. It is not necessary to actually change the input so as to have the network give the correct answer; all that is required is that the loss be reduced.

In the experiments shown in Tab. 1, we defined T_a as the set of successfully corrected samples in step 3 of our adversarial correction approach. If we instead consider T_a to include all perturbed samples, regardless of whether their outputs are corrected, T' will have the same size as the original training dataset. We refer to this as a variant approach. In

Table 5. Accuracy (%) of FP32 baseline models (BL), which is fine-tuned on the CIFAR train domains, and accuracy of baselines after applying our approach (denoted as BL-IST) by using different attacks to generate adversarial domains. The data is reported as an average of three seeds.

Model	Approach	CIFAR-10						CIFAR-100					
		Corr. rate	T'	Train	Valid	Test	Δ Acc	Corr. rate	T'	Train	Valid	Test	Δ Acc
ResNet-18 (11.19M)	BL (FP32)	-	-	99.61 $\pm$ 0.56	93.73 $\pm$ 0.43	93.29 $\pm$ 0.37	-	-	-	99.00 $\pm$ 1.19	76.84 $\pm$ 0.12	77.04 $\pm$ 0.08	-
	D-CORAL	-	99.96 $\pm$ 0.04	99.69 $\pm$ 0.36	95.91 $\pm$ 0.20	95.57 $\pm$ 0.13	+2.28	-	98.86 $\pm$ 0.94	98.84 $\pm$ 0.98	80.14 $\pm$ 0.47	80.27 $\pm$ 0.74	+3.23
	D-CORAL + LL	55/176	100.00	99.80 $\pm$ 0.28	96.17 $\pm$ 0.08	95.93 $\pm$ 0.15	+2.64	70/451	100.00	99.20 $\pm$ 0.92	80.99 $\pm$ 0.21	80.93 $\pm$ 0.46	+3.90
	D-CORAL + BIH	99/176	100.00	99.86 $\pm$ 0.19	96.16 $\pm$ 0.28	95.87 $\pm$ 0.24	+2.58	51/451	100.00	99.36 $\pm$ 0.72	80.75 $\pm$ 0.54	80.99 $\pm$ 0.45	+3.96
	D-CORAL + VBI _{iter1}	121/176	100.00	99.59 $\pm$ 0.46	96.09 $\pm$ 0.22	95.97 $\pm$ 0.12	+2.68	226/451	100.00	99.47 $\pm$ 0.59	80.81 $\pm$ 0.18	80.92 $\pm$ 0.56	+3.89
	D-CORAL + VBI	175/176	100.00	99.97 $\pm$ 0.04	96.19 $\pm$ 0.15	95.77 $\pm$ 0.06	+2.48	446/451	100.00	99.80 $\pm$ 0.20	80.37 $\pm$ 0.98	80.54 $\pm$ 0.80	+3.50
	D-CORAL + DDN	176/176	100.00	100.00	96.21 $\pm$ 0.28	95.84 $\pm$ 0.07	+2.55	451/451	99.98 $\pm$ 0.01	99.98 $\pm$ 0.01	80.79 $\pm$ 0.45	80.82 $\pm$ 0.35	+3.79
	D-CORAL + SP	45/176	100.00	99.79 $\pm$ 0.29	96.17 $\pm$ 0.12	95.80 $\pm$ 0.08	+2.51	43/451	100.00	99.16 $\pm$ 1.00	80.63 $\pm$ 0.54	80.89 $\pm$ 0.61	+3.86
ResNet-34 (21.30M)	BL (FP32)	-	-	99.43 $\pm$ 0.67	94.71 $\pm$ 0.05	94.22 $\pm$ 0.06	-	-	-	94.36 $\pm$ 2.24	78.12 $\pm$ 0.79	78.41 $\pm$ 0.10	-
	D-CORAL	-	99.92 $\pm$ 0.03	99.81 $\pm$ 0.10	96.78 $\pm$ 0.08	96.40 $\pm$ 0.05	+2.18	-	95.38 $\pm$ 1.56	95.26 $\pm$ 1.64	82.99 $\pm$ 0.48	82.98 $\pm$ 0.07	+4.57
	D-CORAL + LL	25/80	99.98 $\pm$ 0.02	99.89 $\pm$ 0.07	96.53 $\pm$ 0.16	96.31 $\pm$ 0.12	+2.09	370/2538	100.00	96.05 $\pm$ 1.39	83.13 $\pm$ 0.08	82.69 $\pm$ 0.12	+4.28
	D-CORAL + BIH	46/80	99.99 $\pm$ 0.01	99.94 $\pm$ 0.06	96.53 $\pm$ 0.23	96.36 $\pm$ 0.07	+2.14	655/2538	100.00	97.31 $\pm$ 1.19	83.04 $\pm$ 1.19	83.31 $\pm$ 0.06	+4.90
	D-CORAL + VBI _{iter1}	53/80	99.99	99.97 $\pm$ 0.01	96.62 $\pm$ 0.07	96.26 $\pm$ 0.12	+2.04	1207/2538	99.99	97.40 $\pm$ 0.92	83.39 $\pm$ 0.44	83.11 $\pm$ 0.23	+4.70
	D-CORAL + VBI	80/80	100.00	100.00	96.71 $\pm$ 0.22	96.26 $\pm$ 0.12	+2.04	2490/2538	100.00	99.21 $\pm$ 0.12	83.34 $\pm$ 0.36	83.26 $\pm$ 0.45	+4.85
	D-CORAL + DDN	80/80	100.00	100.00	96.71 $\pm$ 0.22	96.71 $\pm$ 0.05	+2.49	2538/2538	99.98 $\pm$ 0.01	99.97 $\pm$ 0.01	83.55 $\pm$ 0.53	83.64 $\pm$ 0.06	+5.23
	D-CORAL + SP	23/80	99.98 $\pm$ 0.01	99.90 $\pm$ 0.09	96.52 $\pm$ 0.12	96.22 $\pm$ 0.05	+2.00	118/2538	100.00	95.74 $\pm$ 1.48	83.33 $\pm$ 0.37	83.25 $\pm$ 0.29	+4.84
ResNet-50 (23.57M)	BL (FP32)	-	-	99.81 $\pm$ 0.14	95.36 $\pm$ 0.36	94.32 $\pm$ 0.59	-	-	-	98.81 $\pm$ 0.73	80.01 $\pm$ 0.65	79.74 $\pm$ 0.19	-
	D-CORAL	-	99.92 $\pm$ 0.03	99.78 $\pm$ 0.04	96.65 $\pm$ 0.16	96.61 $\pm$ 0.12	+2.29	-	99.84 $\pm$ 0.01	98.34 $\pm$ 0.38	83.70 $\pm$ 0.14	83.89 $\pm$ 0.22	+4.15
	D-CORAL + LL	46/131	99.96 $\pm$ 0.01	99.84 $\pm$ 0.01	96.57 $\pm$ 0.19	96.31 $\pm$ 0.11	+1.99	60/775	99.99 $\pm$ 0.01	98.58 $\pm$ 0.36	83.29 $\pm$ 0.43	83.11 $\pm$ 0.48	+3.37
	D-CORAL + BIH	69/141	99.95 $\pm$ 0.03	99.85 $\pm$ 0.02	96.41 $\pm$ 0.11	96.11 $\pm$ 0.16	+1.79	261/775	99.98 $\pm$ 0.02	98.69 $\pm$ 0.26	82.86 $\pm$ 0.50	83.03 $\pm$ 0.43	+3.29
	D-CORAL + VBI _{iter1}	79/231	99.99 $\pm$ 0.01	99.89 $\pm$ 0.02	96.61 $\pm$ 0.10	96.18 $\pm$ 0.26	+1.86	304/775	99.99	99.02 $\pm$ 0.21	83.59 $\pm$ 0.46	83.00 $\pm$ 0.17	+3.26
	D-CORAL + VBI	130/131	99.99 $\pm$ 0.01	99.96 $\pm$ 0.01	96.56 $\pm$ 0.12	96.50 $\pm$ 0.18	+2.18	741/775	99.98 $\pm$ 0.02	99.57 $\pm$ 0.14	82.96 $\pm$ 0.23	82.87 $\pm$ 0.07	+3.13
	D-CORAL + DDN	131/131	99.97 $\pm$ 0.04	99.97 $\pm$ 0.04	96.61 $\pm$ 0.22	96.35 $\pm$ 0.12	+2.03	775/775	99.98 $\pm$ 0.01	99.98 $\pm$ 0.01	83.29 $\pm$ 0.42	83.03 $\pm$ 0.07	+3.29
	D-CORAL + SP	17/131	99.97 $\pm$ 0.01	99.82 $\pm$ 0.01	96.61 $\pm$ 0.27	96.30 $\pm$ 0.15	+1.98	45/775	99.99 $\pm$ 0.01	98.58 $\pm$ 0.35	83.00 $\pm$ 0.19	83.25 $\pm$ 0.32	+3.51
EffNetV2-M (52.99M)	BL (FP32)	-	-	99.96 $\pm$ 0.06	97.66 $\pm$ 0.13	97.15 $\pm$ 0.14	-	-	-	99.88 $\pm$ 0.08	86.63 $\pm$ 0.73	86.88 $\pm$ 0.46	-
	D-CORAL	-	99.97 $\pm$ 0.01	99.95 $\pm$ 0.01	98.21 $\pm$ 0.08	97.76 $\pm$ 0.14	+0.61	-	99.72 $\pm$ 0.11	99.62 $\pm$ 0.14	87.73 $\pm$ 0.59	87.36 $\pm$ 0.57	+0.48
	D-CORAL + LL	3/9	100.00	99.98	98.14 $\pm$ 0.09	97.82 $\pm$ 0.08	+0.67	17/54	99.95 $\pm$ 0.05	99.87 $\pm$ 0.01	88.05 $\pm$ 0.22	87.52 $\pm$ 0.45	+0.64
	D-CORAL + BIH	6/9	100.00	99.99 $\pm$ 0.01	98.20 $\pm$ 0.09	97.82 $\pm$ 0.09	+0.68	23/54	99.97 $\pm$ 0.05	99.91 $\pm$ 0.09	87.96 $\pm$ 0.32	88.00 $\pm$ 0.10	+1.12
	D-CORAL + VBI _{iter1}	7/9	99.99 $\pm$ 0.01	99.99 $\pm$ 0.01	98.18 $\pm$ 0.11	97.82 $\pm$ 0.12	+0.67	29/54	99.94	99.87 $\pm$ 0.05	88.00 $\pm$ 0.03	87.77 $\pm$ 0.06	+0.89
	D-CORAL + VBI	8/9	99.99 $\pm$ 0.01	99.99 $\pm$ 0.01	98.13 $\pm$ 0.12	97.80 $\pm$ 0.14	+0.65	46/54	99.95 $\pm$ 0.01	99.88 $\pm$ 0.04	88.09 $\pm$ 0.18	87.76 $\pm$ 0.16	+0.88
	D-CORAL + DDN	9/9	100.00	100.00	98.18 $\pm$ 0.09	97.86 $\pm$ 0.06	+0.71	54/54	99.92 $\pm$ 0.04	99.92 $\pm$ 0.04	87.98 $\pm$ 0.18	87.81 $\pm$ 0.10	+0.93
	D-CORAL + SP	4/9	99.99	99.98	98.13 $\pm$ 0.05	97.70 $\pm$ 0.12	+0.55	18/54	99.95 $\pm$ 0.03	99.87 $\pm$ 0.07	87.85 $\pm$ 0.04	87.89 $\pm$ 0.19	+1.01

Table 6. Accuracy (%) of quantized (Int8) ResNets of various sizes obtained after applying PTSQ on its baseline, and the accuracy of Int8 ResNets using our approach.

Model	Approach	CIFAR-10						CIFAR-100					
		Corr. rate	T'	Train	Valid	Test	Δ Acc	Corr. rate	T'	Train	Valid	Test	Δ Acc
ResNet-18	BL (FP32)	-	-	99.61 $\pm$ 0.56	93.73 $\pm$ 0.43	93.29 $\pm$ 0.37	-	-	-	99.00 $\pm$ 1.19	76.84 $\pm$ 0.12	77.04 $\pm$ 0.08	-
	BL (INT8)	-	-	98.08 $\pm$ 0.71	93.01 $\pm$ 0.56	92.42 $\pm$ 0.17	-	-	-	96.74 $\pm$ 3.02	75.45 $\pm$ 1.29	76.06 $\pm$ 0.94	-
	D-CORAL	-	99.93 $\pm$ 0.01	99.23 $\pm$ 0.05	95.30 $\pm$ 0.39	95.18 $\pm$ 0.09	+2.76	-	99.52 $\pm$ 0.14	96.67 $\pm$ 0.56	79.15 $\pm$ 0.53	79.15 $\pm$ 0.26	+3.09
	D-CORAL + BIH	158/736	99.99 $\pm$ 0.01	99.46 $\pm$ 0.04	95.53 $\pm$ 0.29	95.48 $\pm$ 0.18	+3.07	300/1966	99.98 $\pm$ 0.02	97.36 $\pm$ 0.55	79.25 $\pm$ 0.30	79.53 $\pm$ 0.58	+3.47
	D-CORAL + SP	128/736	100.00	99.48 $\pm$ 0.02	95.46 $\pm$ 0.20	95.29 $\pm$ 0.06	+2.93	189/1966	99.98 $\pm$ 0.01	97.31 $\pm$ 0.66	79.27 $\pm$ 0.73	79.79 $\pm$ 0.49	+3.73
ResNet-34	BL (FP32)	-	-	99.43 $\pm$ 0.67	94.71 $\pm$ 0.05	94.22 $\pm$ 0.06	-	-	-	94.36 $\pm$ 2.24	78.12 $\pm$ 0.79	78.41 $\pm$ 0.10	-
	BL (INT8)	-	-	98.16 $\pm$ 0.35	93.63 $\pm$ 0.10	93.36 $\pm$ 0.09	-	-	90.32 $\pm$ 2.30	76.20 $\pm$ 0.39	77.13 $\pm$ 0.45	-	-
	D-CORAL	-	99.92 $\pm$ 0.04	99.31 $\pm$ 0.17	96.19 $\pm$ 0.10	96.08 $\pm$ 0.20	+2.72	-	99.44 $\pm$ 0.55	92.59 $\pm$ 0.33	81.89 $\pm$ 0.63	81.94 $\pm$ 0.45	+4.81
	D-CORAL + BIH	250/771	99.97 $\pm$ 0.02	99.39 $\pm$ 0.12	96.29 $\pm$ 0.08	96.05 $\pm$ 0.07	+2.69	689/4607	99.99 $\pm$ 0.01	93.77 $\pm$ 0.33	82.19 $\pm$ 0.14	82.18 $\pm$ 0.20	+5.05
	D-CORAL + SP	272/771	99.95 $\pm$ 0.05	99.45 $\pm$ 0.05	96.25 $\pm$ 0.06	95.83 $\pm$ 0.19	+2.47	479/4607	100.00	93.55 $\pm$ 0.53	81.99 $\pm$ 0.24	82.12 $\pm$ 0.20	+4.99

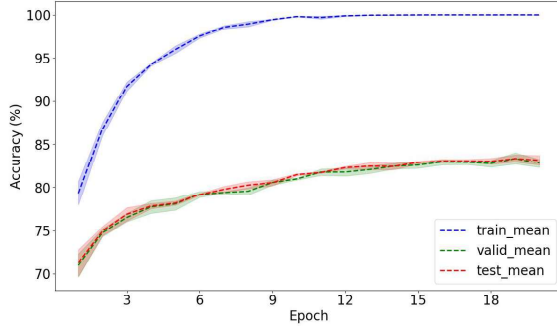
our original approach, the accuracy of the original network on T' reaches 100% when considering only the successfully perturbed samples and the original correctly detected samples. Inspired by [34], we can view T' as an *easy* dataset, given its 100% accuracy, while considering T as a *hard* dataset. As shown in Tab. 7, the variant approach results in a smaller performance improvement compared to our original method. This decline may be due to adversarial perturbations increasing the loss rather than decreasing it for uncorrected inputs, leading to a negative impact relative to the baseline. From this, we conclude that only the corrected input samples should be retained.

Table 7. Accuracy (%) of ResNet FP32 baselines after applying our approach using the LL attack to generate adversarial domains for CIFAR datasets. Note that “D-CORAL + LL (V)” is a variant approach in which T_a in Step 3 incorporates all perturbed samples of T_w .

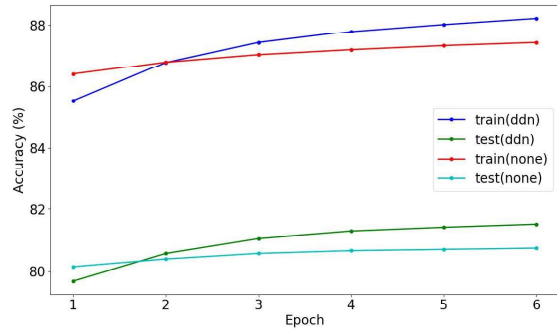
Model	Approach	CIFAR-10		CIFAR-100	
		# T'	Test	# T'	Test
ResNet-18	BL (FP32)	-	93.32	-	77.09
	D-CORAL + LL	44,972	95.77 (+2.45)	44,879	80.48 (+3.39)
	D-CORAL + LL (V)	45,000	95.51 (+2.19)	45,000	79.56 (+2.47)
ResNet-34	BL (FP32)	-	94.24	-	78.53
	D-CORAL + LL	44,993	96.36 (+2.12)	42,903	82.76 (+4.23)
	D-CORAL + LL (V)	45,000	96.18 (+1.94)	45,000	80.81 (+2.28)

B.3. Adversarial Correction with Longer Domain Adaptation Process

Figure 5 demonstrates the performance of our pipeline on the training, validation, and test datasets of CIFAR-100 using ResNet-34. By applying Deep CORAL, we add an extra loss term (*i.e.*, CORAL loss), which helps reduce overfitting by acting as a regularizing term. This approach is less likely to overfit compared to training the baseline model.



(a) ResNet-34 on CIFAR-100



(b) TinyVIT-21M on ImageNet-1K

Figure 5. This graph shows our pipeline training, validation, and test accuracy with the DDN attack using (a) the ResNet-34 model on the CIFAR-100 dataset and (b) the TinyVIT-21M model on the ImageNet-1K. The accuracy results are averaged over three random seeds.

B.4. Adversarial Correction vs. Adversarial Training

While there are similarities in our approach and adversarial training, there are significant differences. Adversarial training takes training set images which the network classifies correctly and alters (attacks) these images so that the network gives the wrong answer. These images, with the correct label, are used to augment the training set. Our approach, on the other hand, takes training set images which the network classifies incorrectly, and alters (attacks) these images so that the network gives the right answer. Instead of augmenting the training set with these examples, we replace the initially wrong images with the adversarially corrected images. The advantage of our approach is that it improves accuracy as

compared to standard adversarial training, since we are providing guidance for the network on how to do better on images it had trouble with. However, both approaches, as shown in the paper, provide robustness to adversarial attacks.

B.5. Adversarial Correction Against Noisy Labeled Datasets

We investigate the performance of our AdCorDA approach under 20% symmetrical noisy labeled setting on CIFAR datasets. We first established new CIFAR baselines using standard fine-tuning, then applied our AdCorDA approach. Table 8 compares the performance of our approach using ResNet-34 on CIFAR datasets with and without noisy labels. The results demonstrate that our approach remains highly effective under noisy label settings, outperforming the noisy CIFAR-10 baseline by 4.77% in the DA-only case and the noisy CIFAR-100 baseline by 8.77% in the DDN attack case.

Table 8. Accuracy (%) of baseline and our AdCorDA approach with Deep CORAL on CIFAR datasets using ResNet-34 with and without 20% noisy labels.

Dataset	Approach	Corr. Rate	Train	Test
CIFAR-10-noisy	BL (FP32)	-	76.26	91.07
	D-CORAL	-	78.98	95.84 (+4.77)
	D-CORAL + DDN	10685/10685	78.22	93.40 (+2.33)
CIFAR-100-noisy	BL (FP32)	-	72.92	72.77
	D-CORAL	-	75.07	81.31 (+8.54)
	D-CORAL + DDN	12188/12188	77.40	81.54 (+8.77)
CIFAR-10	BL (FP32)	-	99.43	94.22
	D-CORAL	-	99.81	96.40 (+2.18)
	D-CORAL + DDN	80/80	100.00	96.71 (+2.49)
CIFAR-100	BL (FP32)	-	94.36	78.41
	D-CORAL	-	95.26	82.98 (+4.57)
	D-CORAL + DDN	2538/2538	99.97	83.64 (+5.23)

C. Visualization of Adversarial Correction and Attacks

C.1. Visualization of Adversarial Attacks

In recent years, there has been an increasing amount of research aimed at developing techniques to deceive neural networks. These techniques, known as adversarial attacks, involve making malicious yet subtle changes in the input to fool the network [5, 14, 21, 22, 29]. Adversarial attacks make malicious yet subtle changes in the input to fool the network, as shown in Fig. 7. These changes are often imperceptible to the human eye, making it difficult to distinguish between the original image and the adversarially altered one. Our adversarial correction approach, on the other hand, takes training set images which the network classifies incorrectly, and alters (attacks) these images so that the network gives the right answer. Similarly, the differences between the misclassified image and the corrected image are not visually noticeable (see Fig. 9).

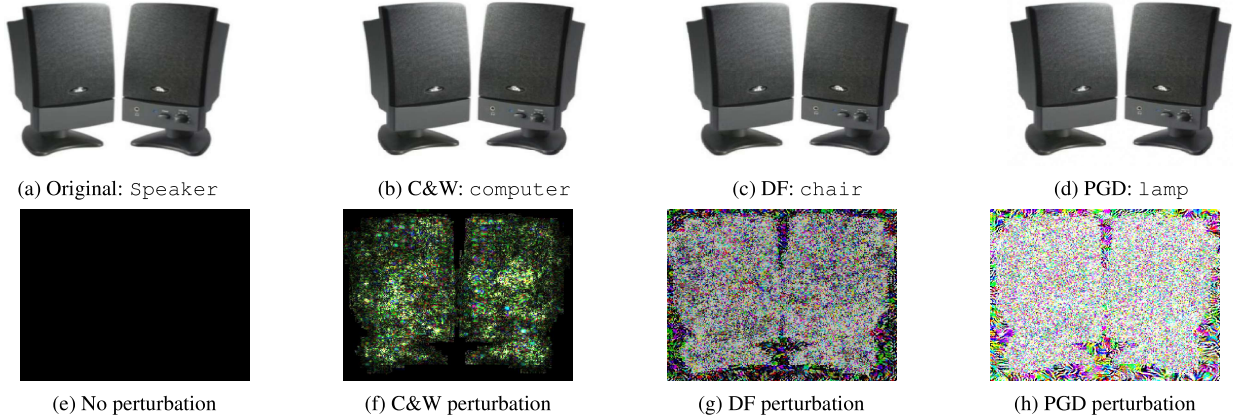


Figure 7. Subplots (a)-(d) show misclassified images of a speaker from the Amazon domain by ResNet-50 under C&W, DF, and PGD adversarial attacks. Subplots (e)-(h) show the corresponding perturbations generated under attacks, magnified by a factor of 500.

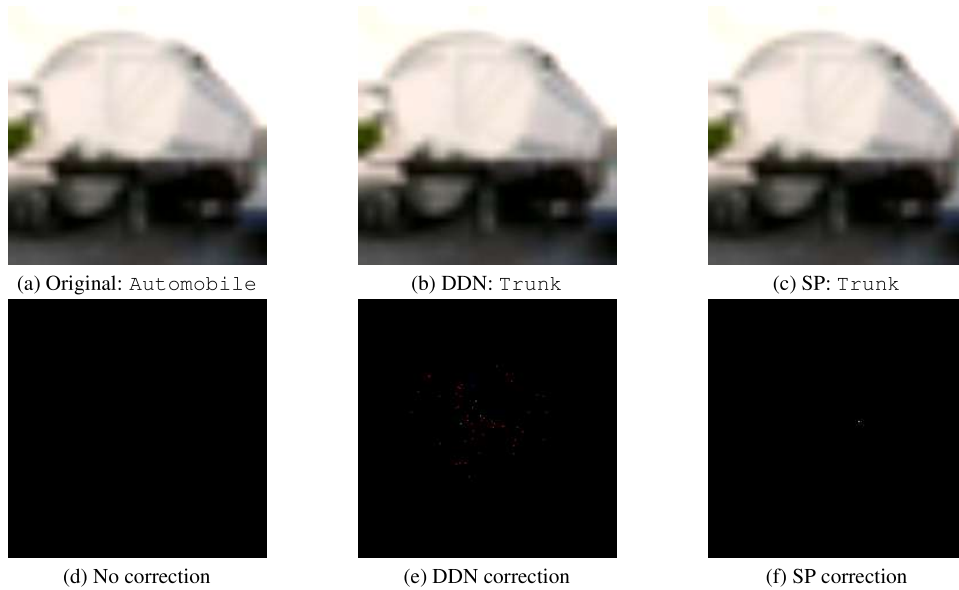


Figure 9. Subplots (a)-(c) show misclassified images of a trunk from the CIFAR-10 dataset by ResNet-34 and its corrected images under DDN and Salt and Pepper noise attacks. Subplots (d)-(f) show the corresponding perturbations generated under adversarial correction, magnified by a factor of 1000.

C.2. Grad-CAM Visualization of Adversarial Correction

To help visualize the impact of the adversarial correction technique on misclassified images, we employ Gradient-weighted Class Activation Mapping (Grad-CAM) [33] to provide visual explanations. Grad-CAM utilizes gradient-based localization to identify important regions in an image that contribute to the prediction of the model concept. In our study, Fig. 10a is an example initially misclassified as “automobile” by ResNet-34. However, applying the DDN attack, the image can be correctly identified as a “horse”. To better understand the differences between the Grad-CAM

of the original (Fig. 10c) and its corrected image (Fig. 10d), we present a visualization in Fig. 10b. This visualization clearly illustrates that incorrect detection was primarily influenced by the contextual information surrounding the object rather than the object itself, demonstrating that by modifying the contextual information surrounding the image using adversarial attack, correct classification becomes possible.

D. Experiments Setup (Extension)

The efficiency of our method is evaluated on the basis of the reconstructed baseline result. The data transformation strategy we use in the data loading process should match

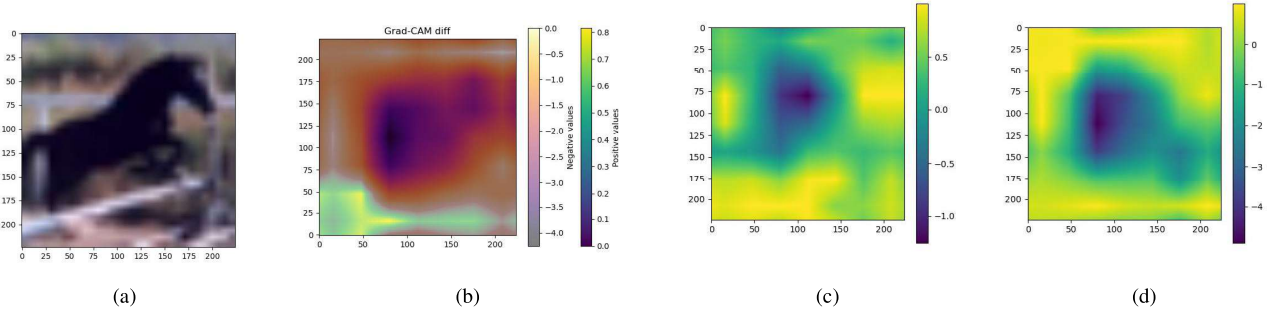


Figure 10. Evaluation of ResNet-34 on CIFAR-10 dataset. (a) misclassified images, (b) the difference between the Grad-CAM images for the original and adversarially corrected inputs using DDN attack. This illustrates the shift in focus of the network for the two images, (c) the Grad-CAM image for the original incorrect image, (d) the Grad-CAM image for the adversarially corrected image.

the pre-trained model to produce the same result. In our experiments, we find that including random crops in data transformation can cause problems when applying adversarial attacks. Since random crop cuts a random region of the image, it causes the image we want to attack to differ from the original image. This can cause unsuccessful attacks and unstable performance of our method. It is worth mentioning that we need to shuffle the dataset T' before conducting domain adaptation training. This ensures a mixture of training samples, including those from the original dataset for which the trained network gets the correct answers and the incorrect samples successfully perturbed.

In detail, we shuffle datasets before applying random crops and save the coordinates of the random crop regions in a file with the order of images in the training dataset if a random crop is included in the transformation process. For the correct dataset (a subset of the train dataset with correct predictions), we reload the random crop file and apply the recorded random crops in the domain adaptation training step. The correct dataset indices are saved following the order of the train dataset when we generate the model baseline so we can use these indexes to obtain random crop regions for the correct dataset. For the incorrect dataset (a subset of the train dataset predicted as incorrect) that needs to be attacked, we use the same strategy to obtain random crop indexes and apply them to images before applying attacks. Then we check the correction rate by comparing the labels before and after the attack to ensure that the labels are different. We also test the prediction of our baseline model on the correct dataset and altered dataset (incorrect dataset processed with attack). The accuracy for the correct dataset should be 100% and the accuracy for the altered dataset should be near 100% while using the DDN attack.

In the domain adaptation training step, we load the file only based on the current batch size. To be more precise, in the domain adaptation training, for each step, when the index is in the correct index list, the source image, target image,

and source label are just the data from the train dataset. When the index is in the incorrect index list, the source image and source label processed with attacks should be taken from the altered dataset, and the target images are taken from the train dataset.

Memory overflow is another problem we face in the domain adaptation step. The loading of the entire file that stores all the attacked images can take a significant amount of memory, especially when the dataset is extremely large, such as ImageNet-1K. Therefore, we individually save these images into a *.npz* file, containing one attacked image and its correct label. Then, we design a customized dataloader that loads the correct images and the altered images as our source dataloader.

E. Consumption of Computing Resources

The adversarial correction process does take some time, but it's not as long as the training time. The computer resources needed to reproduce our experiment results are summarized in the Tab. 9. For large datasets, if we assume the performance decreases on larger datasets, we will have more corrections to do. However, the increase in time is expected to be near linear or slightly more than linear, not quadratic.

Table 9. Compute resources used for each dataset with a batch size of 16. The resources and time may vary slightly depending on the selected model. Note that BL refers to baseline training, AC to adversarial correction, DA to the domain adaptation stage without adversarial correction, and DA+AC to domain adaptation with adversarial correction.

Dataset	Experiment	CPU			GPU			Computing Time
		Number	Memory (GB)	Cores	Number	Memory (GB)	Type	
CIFAR-10	BL	1	3	32	1	2	V100	2.0 min/epoch
	AC	1	3	32	1	2	V100	0.4 min/batch
	DA	1	15	32	1	2	V100	1.2 min/epoch
	DA + AC	1	20	32	1	2	V100	1.2 min/epoch
CIFAR-100	BL	1	3	32	1	2	V100	2.5 min/epoch
	AC	1	3	32	1	2	V100	0.5 min/batch
	DA	1	15	32	1	2	V100	1.5 min/epoch
	DA + AC	1	20	32	1	2	V100	1.5 min/epoch
CINIC-10	BL	1	5	32	1	2	V100	12 min/epoch
	AC	1	30	32	1	2	V100	0.5 min/batch
	DA	1	38	32	1	2	V100	6 min/epoch
	DA + AC	1	40	32	1	2	V100	15 min/epoch
ImageNet-10	AC	1	125	32	1	40	A100	0.2 min/batch
	DA	1	125	32	1	40	A100	3 min/epoch
	DA + AC	1	125	32	1	40	A100	4 min/epoch
ImageNet-100	AC	1	125	32	1	40	A100	0.2 min/batch
	DA	1	125	32	1	40	A100	20 min/epoch
	DA + AC	1	125	32	1	40	A100	20 min/epoch
ImageNet-1K	AC	1	40	32	1	12	V100	1 min/batch
	DA	1	45	32	1	12	V100	300 min/epoch
	DA + AC	1	50	32	1	12	V100	300 min/epoch