Supplementary Material: Boosting Adversarial Robustness using Feature Level Stochastic Smoothing

Sravanti Addepalli*, Samyak Jain*, Gaurang Sriramanan*, R. Venkatesh Babu
Video Analytics Lab, Department of Computational and Data Sciences
Indian Institute of Science, Bangalore, India

1. Training Details

We present details on the proposed training regime in this section. The proposed algorithm is presented in Algorithm-1 of the main paper.

We use coarse hyperparameter tuning (in exponents of 10) to fix the coefficients weighting each loss term in Eq.1 and Eq.2 of the main paper. We set the weights of the Cross-Entropy loss (ℓ_{CE}) and the KL divergence between latent space encodings of clean and adversarial samples (KL_2) to 1. We assign weight of 1 to the KL divergence between the softmax predictions of a clean image with and without sampling (KL_4) in L13 of Alg.1, so as to minimize the rejection of benign samples. Further, we set the weight of the KL divergence term that enforces encoder representations to obey the standard Gaussian distribution (KL_1) to 0.01. Lastly, we utilise a weight of 0.1 for the loss between the softmax predictions of adversarial images with and without latent space sampling (KL_3). We use the same set of hyperparameters across all datasets.

We train the network for 120 epochs, using a cyclic learning rate schedule. The maximum learning rate is set to 0.1. We use Stochastic Gradient Descent (SGD) optimizer without momentum, and use a weight decay of 5e-4.

We additionally perform early stopping using 7-step PGD adversarial samples on the designated validation set of each dataset, as explained in Section-3. We select the model which achieves the highest accuracy on adversarial samples in the *No Sampling* case ($Acc_{adv,NS}$). We use early stopping for identifying the best models while training the baseline models as well.

2. Rejection scheme

In this section, we discuss details on the rejection scheme described in Section-4.3 of the main paper. During test time, for each input image, the classifier predicts N outputs after sampling N times from the Gaussian distribution at the output of the encoder. We define the output of the *Smoothed Classifier* to be the most frequently predicted class among the N samples. Further, the sample is rejected if the frequency of the predicted class is below a predefined threshold f. We select the threshold for rejection such that not more than 10% of the samples are correctly classified and rejected [12]. Based on this criteria, the threshold for the proposed method is set to 32 for CIFAR-10 and 23 for CIFAR-100. Therefore, for CIFAR-10 dataset, an input sample is accepted only if the classifier predicts the same output class at least 33 times out of the 100 outputs.

Output of the ideal *Smoothed Classifier* SC, is an expectation of the classifier outputs $C(x, \epsilon)$ over the random variable ϵ, which is sampled from the Standard normal distribution.

$$SC(x) = \mathbb{E}_{\epsilon \sim \mathcal{N}(0,I)}\, C(x, \epsilon) \tag{1}$$

The above output $SC(x)$ is a deterministic value for every image x. However, in the absence of an analytical expression for this expectation, we consider sample statistics over all N outputs obtained after sampling. This sample mean is not a deterministic value, and can vary during test time. A simple trick to make the outputs during test time deterministic is to pre-sample N vectors from $\mathcal{N}(0, I)$, and use the same set of vectors during test time. Another option is to increase the value of N during test time, which makes the predictions more stable to repeated evaluations. The plot of $Acc_{adv,10\%}$ vs. N is shown in Fig.1(a). It can be seen that as the value of N increases, the output becomes more stable. For all experiments in this paper, we consider $N = 100$.

The metrics reported for any finite value of N hold true with a fixed probability. In order to find this probability, we use the method of hypothesis testing as described in the work by Cohen *et al*. [5, 8]. For each test sample, we find the p-value of the two-sided hypothesis test, $n_A \sim Binomial(n_A, n_B, p)$, where n_A and n_B represent the number of times the top-2 classes are predicted. We determine the value α, which serves as an upper bound to this p-value for all test samples. Using this method, we find that for $N = 100$, the probability of predicting an incorrect class for a sample which is claimed to be correct is 0.0045. This

*Equal contribution.
Correspondence to: Sravanti Addepalli (sravantia@iisc.ac.in)

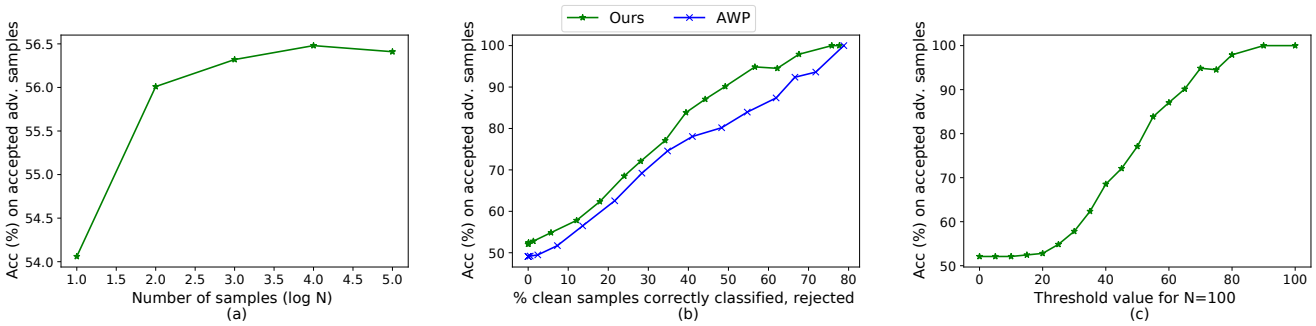

Figure 1. Plots on CIFAR-10 dataset (a) $Acc_{adv,10\%}$ is reported across variation in the number of times the latent vector is sampled during test time (N). The accuracy becomes stable with increase in N. N is varied from 10^1 to 10^5. (b) Plot of $Acc_{adv,10\%}$ against variation in the percentage of clean samples that are correctly classified and rejected. Proposed method achieves improved accuracy compared to the strongest baseline, AWP [14]. (c) $Acc_{adv,10\%}$ across variation in threshold value used for rejection. Threshold value of 32 is selected to limit the percentage rejection of samples which are clean and correctly classified to 10%.

Table 1. **CIFAR-**10: Performance (%) of models under an ensemble of 5 attacks : PGD, APGD-CE, APGD-DLR, FAB, SQUARE. FC denotes the percentage of samples which are always correctly classified and accepted for all attacks. FW denotes the percentage of samples which are accepted and incorrectly classified by at least one of the attacks. MPR denotes the max $\%$ rejected samples.

	$Acc_{nat,NS}\uparrow$	$Acc_{adv,10\%}\uparrow$	FC↑	FW↓	MPR
RS standard training [5]	**86.32**	16.17	14.47	75.00	58.16
PNI-W (Noise) [7]	85.48	34.69	25.31	47.67	64.79
Trades+RS (Noise) [3]	75.14	49.28	41.48	42.69	44.52
PGD (Noise) [10]	84.37	51.63	41.88	39.23	45.32
TRADES (Noise) [15]	80.15	50.83	42.24	40.85	40.92
PGD (Conf) [10]	83.80	59.40	44.03	30.09	41.20
Trades (Conf) [15]	81.77	62.10	44.15	26.94	41.19
AWP (Conf) [14]	80.58	63.38	44.06	25.45	40.55
CCAT [12]	89.92	0.00	0.00	**1.85**	100.00
FLSS (**Ours**)	80.22	**72.40**	**51.10**	19.46	**37.69**

value corresponds to the criterion of rejecting at most 10% samples which are clean and correctly classified. We set the value of N to be 100 since the probability of incorrect prediction is sufficiently low, however this can be further reduced by increasing the value of N.

As shown in Fig.1(c), the proposed *Smoothed Classifier* can achieve a wide range of accuracies based on the threshold selected, at the cost of an increase in the fraction of samples which are clean, correctly classified and rejected. The plot in Fig.1(b) shows that for any given fraction of clean samples which are correctly predicted and rejected, the proposed method achieves higher adversarial accuracy on the accepted samples ($Acc_{adv,10\%}$), when compared to the AWP [14] baseline.

3. Details on Datasets and Model Architecture

To evaluate the proposed approach, we use the benchmark datasets, CIFAR-10 and CIFAR-100. CIFAR-10 [9] is a ten class dataset consisting of RGB images of dimension 32×32, and is commonly used to benchmark results on adversarial robustness of deep networks. The original training set consists of $50,000$ images, which we split into $49,000$ images to comprise the training set, and a hold-out set of $1,000$ images to serve as the validation set. Using this validation set, we perform early-stopping to identify the best model parameters for the proposed approach as well the baseline defense methods. Further, to demonstrate the scalability of the proposed approach to datasets with higher number of classes, we present results on CIFAR-100, which is a 100-class dataset with RGB images of dimension 32×32. We use a class-balanced validation set of size 2500 for CIFAR-100.

We use ResNet-18 architecture to report the performance of the proposed approach as well as other existing defenses such as AWP [14], TRADES [15] and PGD [10]. However, since we use the pre-trained model for CCAT [12], the architecture is ResNet-20, as used by the authors. The CNN backbone of ResNet-18 forms the encoder network of the proposed approach. We require an additional layer with 1×1 convolutions to compute the 512 dimensional mean and variance vectors in the proposed approach when compared to the baselines. Therefore, the architecture of the proposed approach uses 5% additional parameters when compared to the baselines. The MLP in Fig.2 of the main paper consists of 3 layers. However, to ensure that the architecture of the proposed method is close to the ResNet-18 architecture used for baselines, we use 1 layer in the MLP head. We obtain slight gains (close to 0.5%) in robust accuracy by including an additional layer in the MLP head.

4. Experimental results

We present more details on evaluation of the proposed method, and comparison of results against baselines in this section. We use Nvidia DGX workstation with V100 GPUs for our training and evaluation.

Table 2. **Ablations on CIFAR-10**: Performance (%) of models under an ensemble of 6 attacks : PGD, APGD-CE, APGD-DLR [6], PGD-CW [4], GAMA-PGD and GAMA-MT [11]. FC denotes the percentage of samples which are always correctly classified and accepted for all attacks. FW denotes the percentage of samples which are accepted and incorrectly classified by at least one of the attacks. MPR denotes the max % rejected samples.

	Description	$Acc_{nat,NS}\uparrow$	$Acc_{nat,0\%}\uparrow$	$Acc_{nat,10\%}\uparrow$	$Acc_{adv,NS}\uparrow$	$Acc_{adv,0\%}\uparrow$	$Acc_{adv,10\%}\uparrow$	FC ↑	FW ↓	MPR
P	Our Proposed Approach	80.51	77.68	89.63	50.64	51.00	56.16	43.16	33.69	47.42
S1	1-step training (Skip Eq.2 of main paper)	72.16	65.81	86.31	53.16	48.10	58.19	37.47	26.92	53.30
S2	1-step training (Combine Eq.1, 2 of main paper)	80.76	78.31	89.25	50.51	49.45	54.22	42.35	35.75	46.97
A1	Set coefficient of KL1 to 0	80.08	77.42	89.41	53.49	49.17	54.84	41.66	34.30	47.86
A2	Set coefficient of KL2 to 0	84.31	84.07	91.88	40.60	41.31	41.92	36.81	50.98	30.05
A3	Set coefficient of KL3 to 0	79.12	77.71	87.95	53.41	48.91	52.42	42.64	38.70	46.08
A4	Set coefficient of KL4 to 0	81.05	77.99	88.62	50.92	50.93	53.85	44.81	38.40	44.02
A5	Increase coefficient of KL1 from 0.01 to 0.1	81.33	74.51	89.97	50.11	50.56	55.25	41.53	33.62	47.06
A6	Increase coefficient of KL3 from 0.1 to 1	80.34	76.78	89.02	50.70	51.16	55.04	43.95	35.90	45.43
A7	Replace KL2 with $0.1 \cdot KL(adv\|\|\mathcal{N}(0,1))$	85.01	84.09	92.90	43.73	44.01	45.64	39.42	46.94	49.58

4.1. Ablation Experiments

We present results on various ablation experiments in Table-2, to highlight the importance of different components of the proposed algorithm. In S1, we skip training on the second step (Eq.2 of the main paper), which is important for improving the accuracy on clean samples. While this results in a 2% boost in adversarial accuracy after rejection ($Acc_{adv,10\%}$), the clean accuracy drops with respect to the proposed approach (P). In S2, we train on a combination of the losses in Eq.1 and 2 of the main paper. While this saves the computation time for one additional back propagation, the adversarial accuracy drops. It is possible that careful tuning of hyperparameters can retrieve the best accuracy using the combined loss.

We further set each of the KL divergence terms to 0, one at a time, in A1-A4. While all these experiments result in sub-optimal results, setting KL2 to 0 causes a very significant drop in accuracy on adversarial samples, since it directly relates to the robustness objective. We also try to replace KL2 with $0.1 \cdot KL\big(E(\widetilde{x})||\mathcal{N}(0,I)\big)$, which also leads to sub-optimal robustness. As seen in A6, increasing the coefficient of KL1 results in a higher clean accuracy, at the cost of a drop in accuracy of adversarial samples. Increasing KL3 from 0.1 to 1 (A6) also leads to a drop in robustness.

4.2. Combining FLSS with CCAT

We present the performance of the state-of-the-art Adversarial Detection method CCAT [12] against random perturbations sampled from a Bernoulli distribution of varying magnitudes (denoted by δ) in Table-3. Since the goal of Adversarial Detection methods is to be able to detect images which do not belong to the original distribution, they are seen to be overly sensitive to even perturbations of small magnitude. We note that even at $\delta = 1/255$, the network rejects 78.6% of the images at a rejection threshold of 1%. The accuracy of a normally trained ResNet-18 network is indeed very high (92.4%) against images corrupted with such low magnitude ($\delta = 1/255$) random noise sampled from a Bernoulli distribution. In order to address this sensitivity, we propose to combine FLSS with CCAT such that every image that is rejected by CCAT is re-evaluated for acceptance by FLSS. This reduces the rejection to 25.4%, while maintaining a high accuracy of 89.49% after rejection. For the combined model we consider a threshold corresponding to 1% rejection of correctly classified clean samples for CCAT, and 10% for FLSS, which applies to the samples rejected by CCAT. Therefore the effective criteria for the combined model is very close to 1%.

Table 3. **Performance (%) of CCAT [12]** against random Bernoulli noise perturbations of varying magnitude (denoted by δ). The rejection is done such that not more than 1% of the clean samples are correctly classified and rejected. FC denotes the percentage of samples which are always correctly classified and accepted for all attacks. FW denotes the percentage of samples which are accepted and incorrectly classified by at least one of the attacks. R denotes the % rejected samples. CCAT rejects a high fraction of samples even at low perturbation magnitudes.

Noise (δ)	$Acc_{0\%}\uparrow$	$Acc_{1\%}\uparrow$	FC↑	FW ↓	R
$1/255$	75.82	97.99	20.97	0.43	78.60
$2/255$	40.04	97.43	1.14	0.03	98.83
$3/255$	26.03	90.90	0.10	0.01	99.89
$4/255$	21.06	66.66	0.02	0.01	99.97
$5/255$	17.75	0.00	0.00	0.01	99.99
$6/255$	16.21	-	0.00	0.00	100.00
$7/255$	16.03	-	0.00	0.00	100.00
$8/255$	15.96	-	0.00	0.00	100.00

We further evaluate the performance of CCAT, FLSS and a combination of both approaches against an enhanced Maximum Margin attack proposed by Stutz *et al.* [12] in Fig.2. As noted by the authors, this attack causes a higher rate of misclassification on accepted samples (FW) for CCAT when compared to other attacks. As seen in Fig.2(a), with a threshold corresponding to 1% rejection of clean samples that are correctly classified, FW is 36.5%. This reduces to 28.56% when the rejection threshold is increased to 10%. For the same threshold of 10%, the proposed method

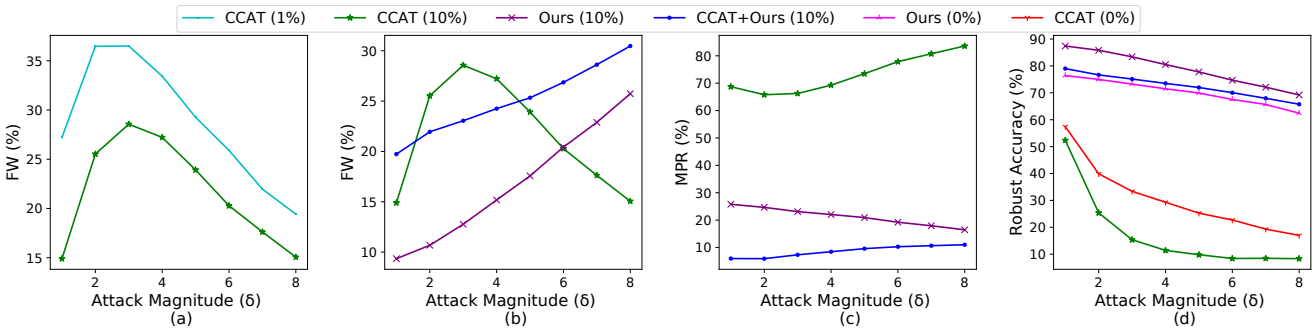

Figure 2. Performance against an enhanced Maximum-Margin attack [12] on CIFAR-10 dataset against CCAT and FLSS (Ours) models (a) Plot of FW on CCAT against attack magnitude (δ) at different FPR values (b) Comparison of FW between CCAT, FLSS (Ours) and CCAT + FLSS (Ours) (c) Maximum Percentage Rejection (%) for the same models (d) Comparison of Robust accuracy with and without rejection

has a worst case FW of 25.73% at $\delta = 8/255$ as shown in Fig.1(b). This is achieved at a significantly lower Maximum Percentage Rejection (MPR) as shown in Fig.2(c). While FW increases to 30.47% when combined with FLSS, this is again achieved at a very low MPR. We present the Robust Accuracy after rejection ($Acc_{adv,10\%}$) in Fig.2(d). Since the CCAT model is not adversarially trained, the accuracy is very low. However, when combined with FLSS, there is a significant boost in accuracy.

4.3. Evaluation against Black-Box attacks

We evaluate the proposed method against transfer based black-box attacks on CIFAR-10 and CIFAR-100 in Tables-5 and 6 respectively. Here, we consider FGSM and PGD 7-step transfer-based attacks, as well as the query-based Square attack [1]. For the transfer attacks, the source model considered is a normally trained model of the same architecture as the target network. We note that FGSM black-box attack is stronger than PGD 7-step attack, while the Square attack is the strongest as expected, since it performs zeroth-order optimization on the model. We also observe that the proposed method performs significantly better than AWP on the strongest attack (Square), for both 0% as well as 10% rejection rates. We note that black-box attacks are significantly weaker than the white-box attacks reported in the main paper. This shows the absence of gradient masking in the proposed method.

4.4. Evaluation using Random Restarts

For reliable evaluation of the proposed defense, we present results against multiple random restarts and multiple steps of the PGD attack on CIFAR-10 and CIFAR-100 datasets in Tables-7 and 8. The purpose of multiple restarts is to find the worst adversary in the δ-ball around each image. It is to be noted that a naive implementation of a series of attacks on a single image could lead to an inadvertent drop in accuracy due to the stochasticity of predictions. In fact, even if a single clean image is repeatedly evaluated (n times) on the proposed classifier, it is likely to be rejected at least once as $n \to \infty$. We capture the same by reporting the probability of correct predictions using hypothesis test-

Table 4. **EOT**: Accuracy (%) of models against Expectation over Transformation (EOT) attack [2] on CIFAR-10 and CIFAR-100 datasets. The base attack considered is PGD-100. EOT-k represents the use of k computations to approximate the expected value of the gradient. We report results for $k = 10$, 50 and 100. The accuracy of the proposed approach is stable to EOT attacks.

	CIFAR-10		CIFAR-100	
	$Acc_{adv,0\%}$	$Acc_{adv,10\%}$	$Acc_{adv,0\%}$	$Acc_{adv,10\%}$
Standard Attack	54.00	65.65	29.16	47.83
EOT - 10	54.40	66.61	30.16	53.96
EOT - 50	54.60	66.82	30.00	53.95
EOT - 100	54.60	67.12	30.08	54.01

Table 5. **CIFAR-**10: Accuracy (%) of models against FGSM, PGD-7 and Square Black-Box attacks. Attack source for FGSM and PGD-7 is a normally trained model of the same architecture.

	$Acc_{nat,0\%}\uparrow$	$Acc_{nat,10\%}\uparrow$	$Acc_{adv,0\%}\uparrow$	$Acc_{adv,10\%}\uparrow$
AWP - FGSM	80.58	89.14	78.01	87.23
Ours - FGSM	77.68	89.63	75.98	88.96
AWP - PGD 7	80.58	89.14	78.86	88.02
Ours - PGD 7	77.68	89.63	76.32	89.12
AWP - Square	80.58	89.14	55.52	78.69
Ours - Square	77.68	89.63	70.40	85.16

Table 6. **CIFAR-**100: Accuracy (%) of models against FGSM, PGD-7 and Square Black-Box attacks. Attack source for FGSM and PGD-7 is a normally trained model of the same architecture.

	$Acc_{nat,0\%}\uparrow$	$Acc_{nat,10\%}\uparrow$	$Acc_{adv,0\%}\uparrow$	$Acc_{adv,10\%}\uparrow$
AWP - FGSM	58.21	74.31	56.52	73.01
Ours - FGSM	47.50	74.35	46.03	73.21
AWP - PGD 7	58.21	74.31	56.78	73.56
Ours - PGD 7	47.50	74.35	46.15	73.62
AWP - Square	58.21	74.31	30.96	53.27
Ours - Square	47.50	74.35	40.35	70.75

Table 7. **CIFAR-**10: Accuracy (%) of models under attacks with varying number of steps and restarts. Accuracy with no rejection and 10% rejection is reported for each attack.

	$Acc_{0\%} \uparrow$	$Acc_{10\%} \uparrow$	$Acc_{0\%} \uparrow$	$Acc_{10\%} \uparrow$	$Acc_{0\%} \uparrow$	$Acc_{10\%} \uparrow$
No. of steps	100	100	1000	1000	100	100
No. of restarts	1	1	1	1	10	10
AWP [14]	53.92	66.36	53.87	66.30	53.92	66.30
FLSS (**Ours**)	54.61	67.89	54.55	67.88	54.57	67.72

Table 8. **CIFAR-**100: Accuracy (%) of models under attacks with varying number of steps and restarts. Accuracy with no rejection and 10% rejection is reported for each attack.

	$Acc_{0\%} \uparrow$	$Acc_{10\%} \uparrow$	$Acc_{0\%} \uparrow$	$Acc_{10\%} \uparrow$	$Acc_{0\%} \uparrow$	$Acc_{10\%} \uparrow$
No. of steps	100	100	1000	1000	100	100
No. of restarts	1	1	1	1	10	10
AWP [14]	31.28	45.66	31.24	45.86	31.11	45.51
FLSS (**Ours**)	29.16	48.00	29.08	47.88	29.01	47.62

Table 9. **Adaptive Attacks on CIFAR-**10: Performance (%) of the proposed model against various adaptive attacks. MPR denotes the maximum percentage rejected samples.

	$Acc_{adv,0\%} \uparrow$	$Acc_{adv,10\%} \uparrow$	$FC \uparrow$	$FW \downarrow$	MPR
A1: FA (KL, targ)	58.50	69.71	47.20	20.50	41.90
A2: FA (MSE, targ)	59.20	71.57	48.60	19.30	43.60
A3: FA (KL, untarg)	62.70	74.61	50.60	17.20	39.90
A4: FA (MSE, untarg)	62.90	74.88	49.50	16.60	41.00
A5: FA (MSE + min var, targ)	58.60	70.57	47.50	19.80	43.20
A6: Diverse CE (sample)	70.20	84.27	55.20	10.30	40.30
RA1: Max entropy	66.80	84.19	45.80	8.60	50.80
RA2: Max entropy + Min CE	77.60	88.38	62.40	8.20	33.30
RA3: Output diversify (all classes)	73.60	86.12	59.60	9.60	34.60
RA4: Maximize variance	73.80	85.67	60.40	10.10	31.60
Ensemble of 6 attacks	50.40	55.39	41.10	33.10	49.50
PGD	53.30	67.40	42.40	20.50	43.60

ing in Section-2. However, since the goal in this section is to merely find an adversary in the δ-ball of each image, we consider a fixed set of noise vectors sampled from $\mathcal{N}(0, I)$ for each attack in Tables-7 and 8. The same is considered for all other evaluations in the paper which use a series of attacks on a single image.

We observe from the first two partitions of Tables-7 and 8 that there is only a marginal drop in robust accuracy between the PGD-100 and PGD-1000 step attacks. This indicates that the robust accuracy saturates and does not deteriorate further as the number of steps used in the PGD attack is increased. Further, from the first and third partitions, we observe that the robust accuracy is preserved even with multiple random restarts of the PGD attack, thereby indicating the absence of the gradient masking effect.

4.5. Evaluation against Adaptive attacks

We evaluate our proposed model against various adaptive adversarial attacks, which are constructed specifically for the defense at hand, as recommended by Tramer *et al.* [13]. The results for CIFAR-10 are reported in Table-9. We use a 1000-sample balanced subset for reporting our results.

We broadly consider two kinds of attacks. The first set of attacks, A1 to A6 are implemented with the objective of fooling the model, whereas the second set of attacks, RA1 to RA4 are crafted to encourage the model to reject the image. For the proposed model, we claim improved adversarial accuracy on accepted samples ($Acc_{adv,10\%}$), while maintaining a reasonable limit on the maximum percentage rejection (MPR). Therefore performance against both types of attacks needs to be considered. In most of the attacks (unless specified), the image is sampled once during the forward propagation for attack generation.

Since the proposed training method relies on the properties of the feature space of an image, we consider a wide range of feature level attacks (FA) at the output of the encoder, to generate an adaptive adversary. In A1 and A2, we craft an adversary that closely resembles a different image from a target class in the feature space. A1 crafts an adversary to minimize the KL divergence between the encoder outputs of the two images, whereas A2 crafts an adversary to minimize the MSE between the predicted mean vectors. For A1 and A2, we consider a random image from each of the 10 classes, and report the worst case accuracy across all attacks. We find that A1 is the strongest among all the adaptive attacks considered, yielding an accuracy of 69.71% on accepted adversarial samples for CIFAR-10 dataset. A3 and A4 are untargeted versions of the attacks A1 and A2 respectively. These attacks generate an adversary to maximize the KL divergence and MSE with respect to the corresponding clean images in the latent space.

A5 optimizes an objective which is similar to A2, however, it also attempts to find an adversary with minimum variance at the output of the encoder. Having a low variance enforces the outputs of all random samples at test time to be similar, thereby encouraging the adversary to be accepted even if it is incorrectly classified. In A6, each of the 100 outputs of the test image are encouraged to be predicted as a fixed random target class, by minimizing cross-entropy loss of each of the softmax vectors with respect to this target. The overall loss which is optimized is the sum of all 100 cross-entropy losses.

The next set of adaptive attacks consider the objective of increasing MPR (Maximum Percentage Rejection). RA1 generates an adversary that maximizes entropy of the output probability vector. This would possibly diversify the prediction of the image when sampled multiple times, thereby resulting in rejection of the image. In RA2, we consider the same objective of maximizing entropy, but additionally enforce that the image is correctly predicted, by minimizing cross-entropy of the image with respect to the true class. We find that RA1 is the strongest attack, leading to 50.8% rejection. In RA3, each of the 100 sampled outputs of the test image are encouraged to be predicted as a different random class. This is done by minimizing cross-entropy loss of each

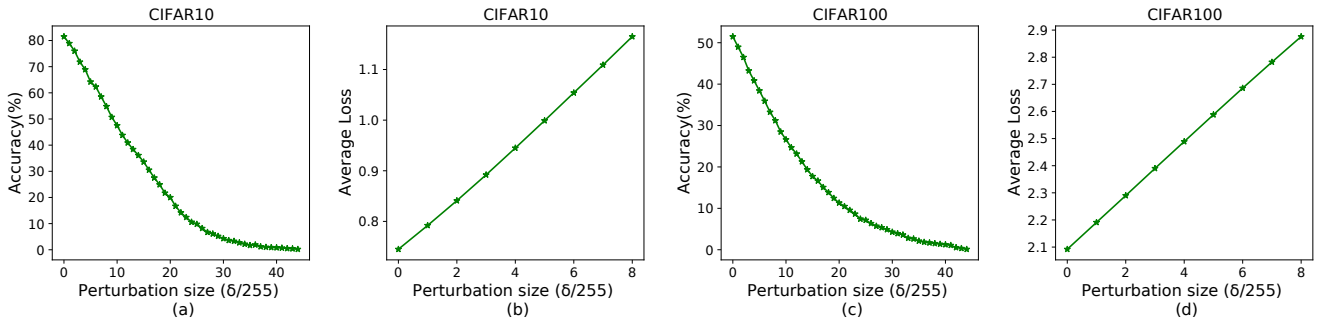

Figure 3. Accuracy and loss on adversaries in a no-sample case (a) Accuracy ($Acc_{adv,NS}$) against PGD-7 step attack on CIFAR-10 dataset. (b) Loss on FGSM adversaries for CIFAR-10 dataset. (c) Accuracy ($Acc_{adv,NS}$) against PGD-7 step attack on CIFAR-100 dataset. (d) Loss on FGSM adversaries for CIFAR-100 dataset. We note that $Acc_{adv,NS}$ goes to 0 for higher perturbation magnitudes, and loss on FGSM samples increases monotonically with δ, indicating the absence of gradient masking.

of the softmax vectors with respect to a different random target, and finding the gradient of sum of all losses. This attack is marginally weaker than RA1 and RA2, possibly because of inconsistency in the loss, resulting in a weak gradient direction. In RA4, an adversary is generated by maximizing variance at the output of the encoder. Higher variance can possibly lead to inconsistent predictions, thereby leading to higher rejection rate. However, we find this to be weaker than the other attacks.

Overall, we find that the ensemble of 6 attacks considered in Table-1 of the main paper are significantly stronger than attacks which can possibly exploit the specific nature of the defense. The attack RA-1 however is able to increase the Rejection rate higher than the ensemble of attacks considered for the main evaluations in the paper. We therefore consider the RA-1 attack on the CIFAR-100 dataset as well. This increases the MPR on CIFAR-100 from 65.61% to 72.28%, while maintaining a significantly high accuracy on adversarial samples before and after rejection.

4.6. Sanity Checks to ensure absence of Gradient Masking

The plots in Fig.3 show that for both CIFAR-10 and CIFAR-100, accuracy against PGD 7-step attack goes to 0 as the magnitude of perturbation (δ) increases. Also, loss on FGSM adversaries for small perturbation magnitudes increases monotonically. Both of these trends indicate the absence of gradient masking in the proposed method [2]. We consider the *No Sampling* case here, since this is primarily a check to verify the efficacy of the defense, and hence the rejection scheme need not be considered.

References

[1] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. *arXiv preprint arXiv:1912.00049*, 2019.

[2] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018.

[3] Avrim Blum, Travis Dick, Naren Manoj, and Hongyang Zhang. Random smoothing might be unable to certify ℓ_∞ robustness for high-dimensional images. *arXiv preprint arXiv:2002.03517*, 2020.

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

[5] Jeremy M Cohen, Elan Rosenfeld, and J Zico Kolter. Certified adversarial robustness via randomized smoothing. *arXiv preprint arXiv:1902.02918*, 2019.

[6] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. *arXiv preprint arXiv:2003.01690*, 2020.

[7] Zhezhi He, Adnan Siraj Rakin, and Deliang Fan. Parametric noise injection: Trainable randomness to improve deep neural network robustness against adversarial attack. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 588–597, 2019.

[8] Kenneth Hung, William Fithian, et al. Rank verification for exponential families. *The Annals of Statistics*, 47(2):758–782, 2019.

[9] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009.

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

[11] Gaurang Sriramanan, Sravanti Addepalli, Arya Baburaj, and R Venkatesh Babu. Guided Adversarial Attack for Evaluating and Enhancing Adversarial Defenses. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[12] David Stutz, Matthias Hein, and Bernt Schiele. Confidence-calibrated adversarial training: Generalizing to unseen attacks. In *Proceedings of the International Conference on Machine Learning*, 2020.

[13] Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. *arXiv preprint arXiv:2002.08347*, 2020.

[14] Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems*, 33, 2020.

[15] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P Xing, Laurent El Ghaoui, and Michael I Jordan. Theoretically principled trade-off between robustness and accuracy. *arXiv preprint arXiv:1901.08573*, 2019.