Towards Robust Detection of Adversarial Examples

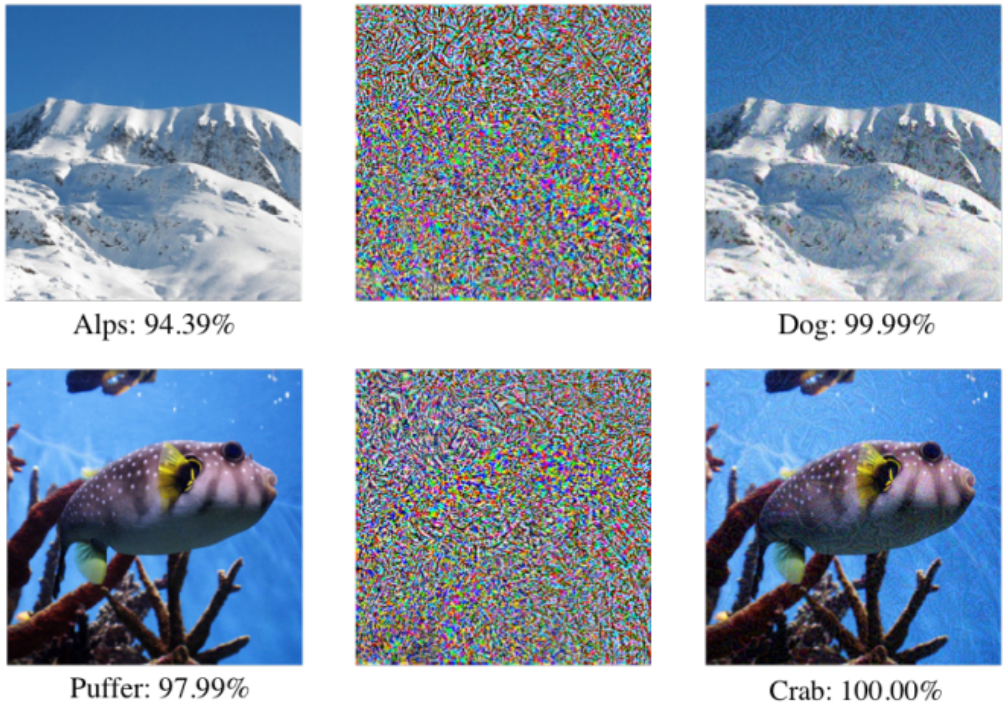
Tianyu Pang, Chao Du, Yinpeng Dong and Jun Zhu

Department of Computer Science and Technology
Tsinghua University

NeurIPS | 2018

TSAIL

Adversarial Examples



From Dong et al. (CVPR 2018)

How to Defend Adversarial Attacks?

Possible strategy one:

To correctly classify adversarial examples

- Optimal
- Difficult to achieve
- Computationally expensive (adversarial training)

How to Defend Adversarial Attacks?

Possible strategy two:

To detect and filter out adversarial examples

- Suboptimal
- Little computation
- Methods borrowed from anomaly detection

We Detect Adversarial Examples, and How?

Design new detectors:

- Kernel density detector (Feinman et al. 2017)
- LID detector (Ma et al. ICLR 2018)
- ……

We Detect Adversarial Examples, and How?

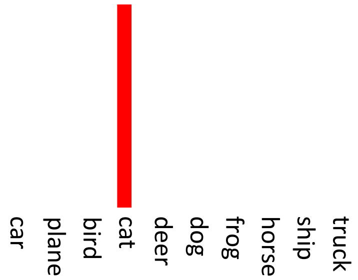
Design new detectors:

- Kernel density detector (Feinman et al. 2017)
- LID detector (Ma et al. ICLR 2018)
- ……

Train the models to better collaborate with existing detectors

Reverse Cross Entropy

Cross-Entropy (CE):

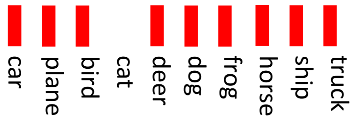


1_y: One-hot label

$\{0, 0, 0, \mathbf{1}, 0, 0, 0, 0, 0, 0\}$

$$\mathcal{L}_{CE} = -\mathbf{1}_y \cdot \log(\mathrm{F})$$

Reverse Cross-Entropy (RCE):



R_y: Reverse label

$\{\frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \mathbf{0}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}, \frac{1}{9}\}$

$$\mathcal{L}_{RCE} = -\boldsymbol{R}_y \cdot \log(\mathrm{F})$$

The RCE Training Method

Phase 1: Reverse Training

Training the model by minimizing the RCE loss

Phase 2: Reverse Logits

Negating the logits fed to the softmax layer to give predictions

Theoretical Analysis

Theorem 2. *(Proof in Appendix A) Let (x, y) be a given training data. Under the L_∞-norm, if there is a training error $\alpha \ll \frac{1}{L}$ that $\|\mathbb{S}(Z_{pre}(x, \theta_R^*)) - R_y\|_\infty \leq \alpha$, then we have bounds*

$$\|\mathbb{S}(-Z_{pre}(x, \theta_R^*)) - 1_y\|_\infty \leq \alpha(L-1)^2,$$

and $\forall j, k \neq y$,

$$|\mathbb{S}(-Z_{pre}(x, \theta_R^*))_j - \mathbb{S}(-Z_{pre}(x, \theta_R^*))_k| \leq 2\alpha^2(L-1)^2.$$

Property 1: Consistent and Unbiased

When the training error $\alpha \longrightarrow 0$, the prediction tends to the one-hot label

Property 2: Tighter Bound

The difference between any two non-maximal elements decreases as $O(\alpha^2)$

The Insights of RCE Training

We first define the non-maximal entropy (non-ME) as:

$$\mathrm{nonME(x)} = -\sum_{i \neq y} \hat{F}(x)_i \log\left(\hat{F}(x)_i\right),$$

where $\hat{F}(x)_i$ is the normalized non-maximal predictions.

The Insights of RCE Training

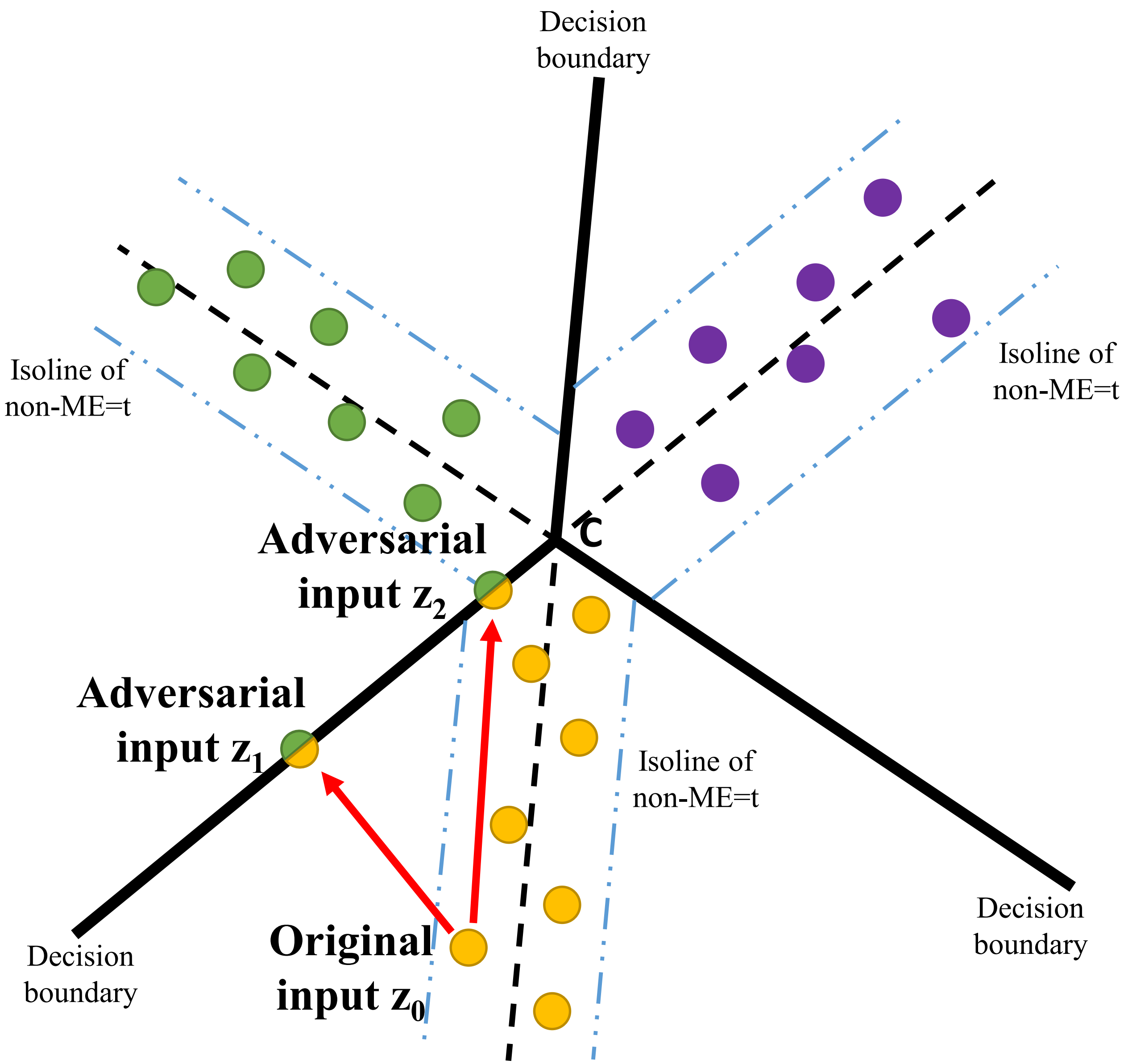
We first define the non-maximal entropy (non-ME) as:

$$\mathrm{nonME(x)} = -\sum_{i \neq y} \hat{F}(x)_i \log\left(\hat{F}(x)_i\right),$$

where $\hat{F}(x)_i$ is the normalized non-maximal predictions.

RCE training encourages the maximal prediction to tend to 1, while maximizing the non-ME.

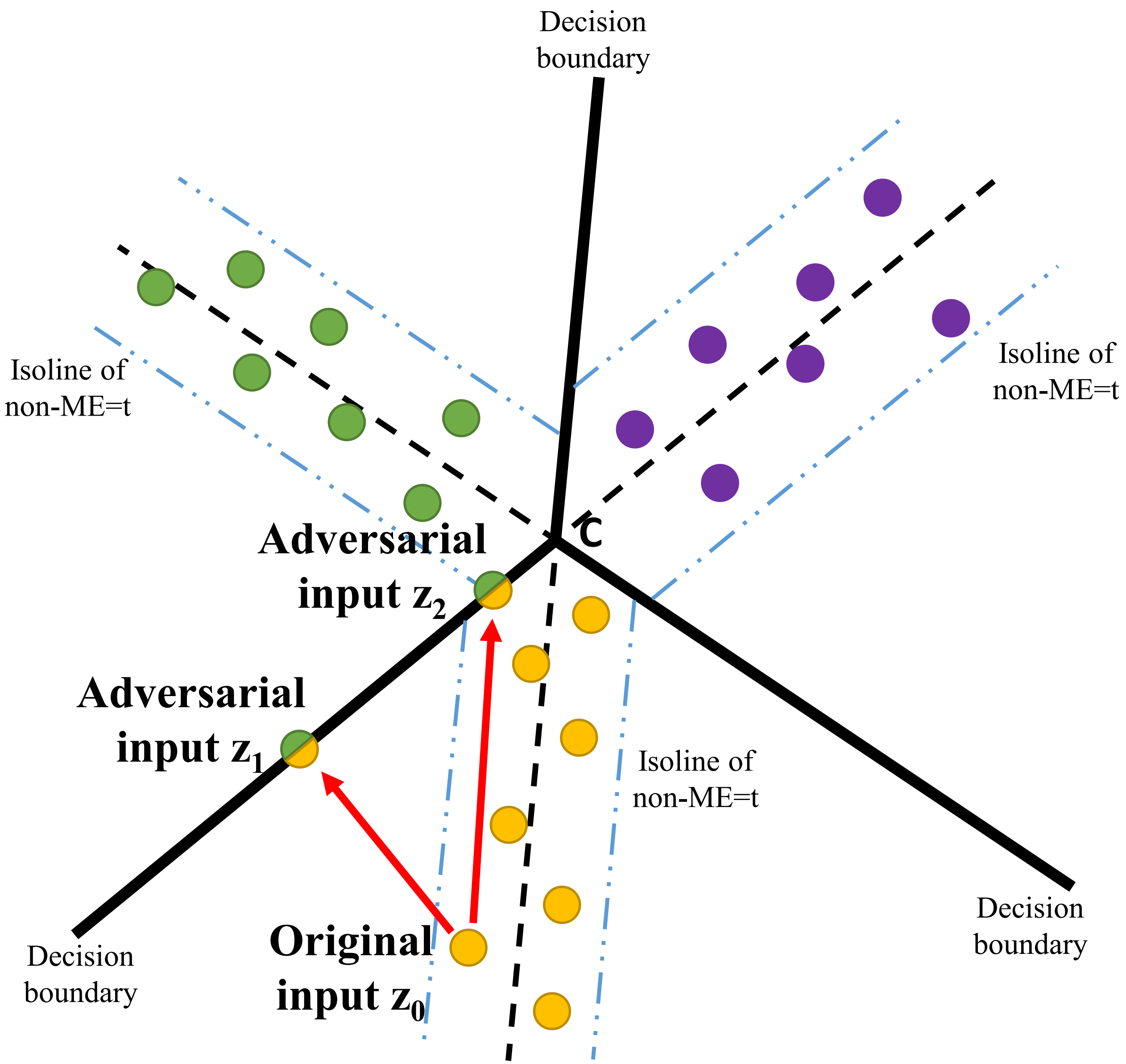
The Insights of RCE Training



The left plot is the decision domain in 2-d feature space for 3 classes (each class with one color)

When the non-ME of the returned predictions are maximized, the learned features for each class with tend to locate near the black dash lines, where the points on the dash lines have the maximal non-ME.

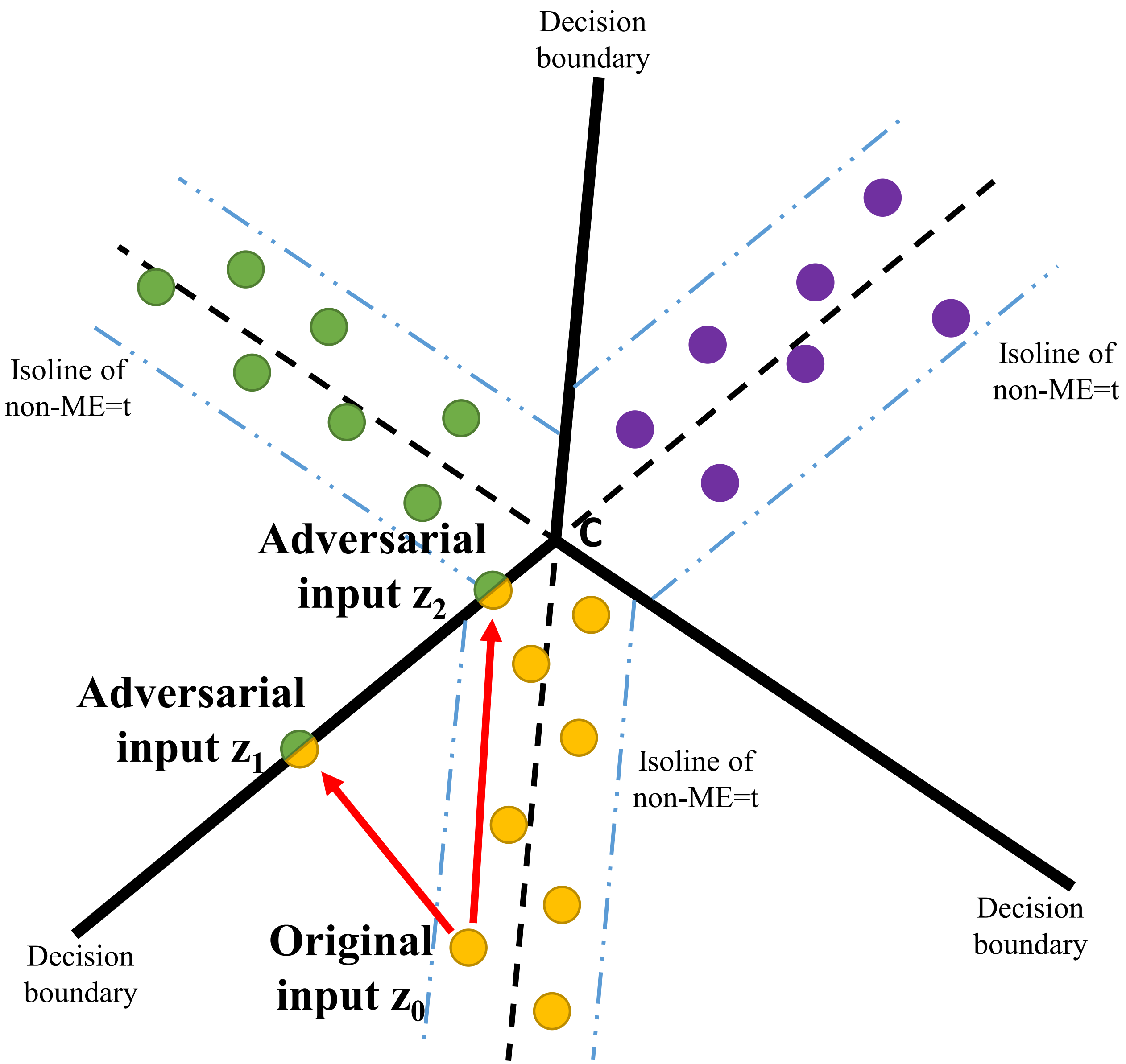
The Insights of RCE Training



The left plot is the decision domain in 2-d feature space for 3 classes (each class with one color)

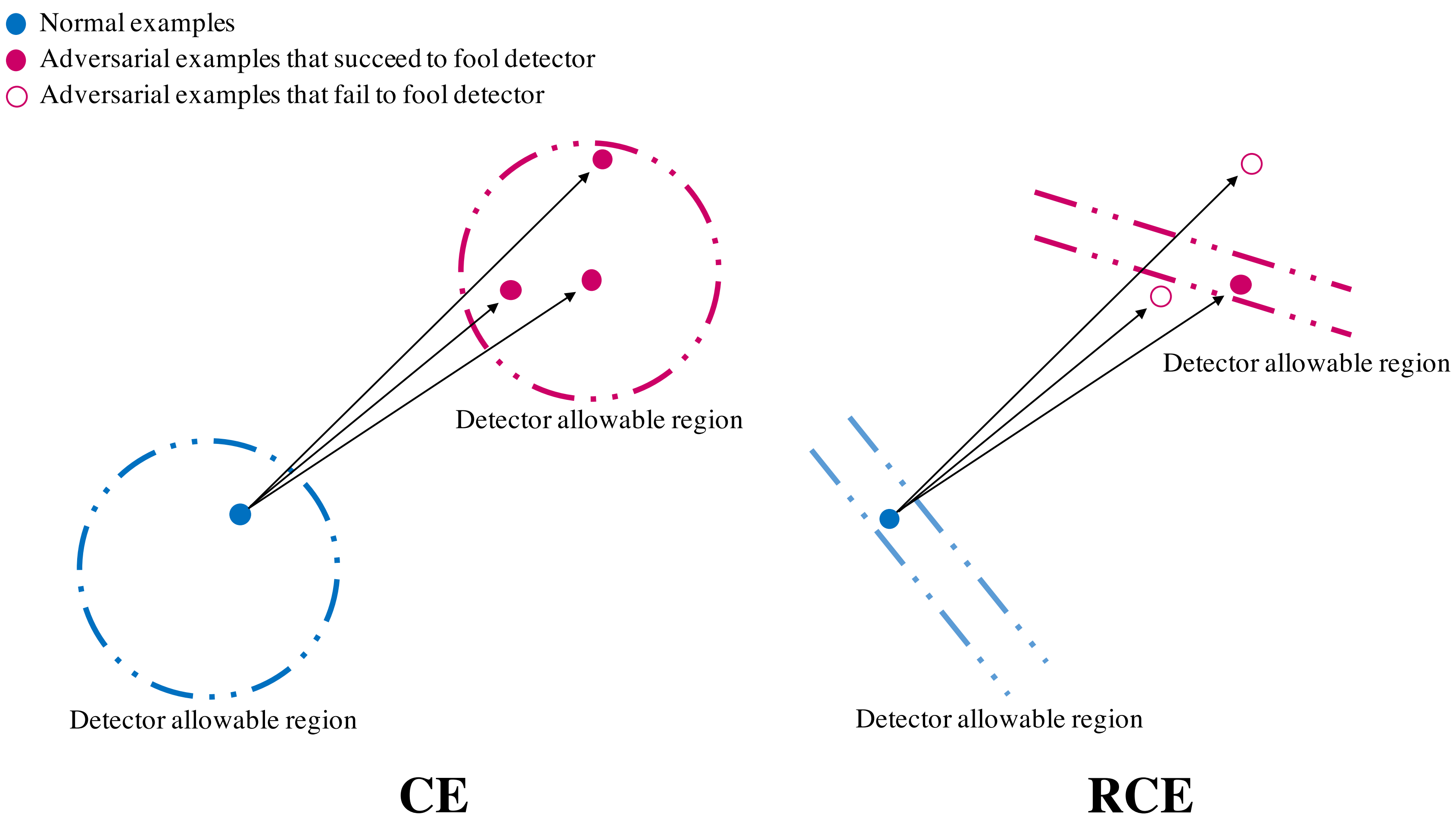
When the non-ME of the returned predictions are maximized, the learned features for each class with tend to locate near the black dash lines, where the points on the dash lines have the maximal non-ME.

The Insights of RCE Training



Then if an adversary want to craft an adversarial example based on z_0, he has to move further to z_2 rather than z_1 to obtain a normal value of non-ME.

The Insights of RCE Training



In practice, the learned low-dimensional feature distributions by RCE make it more difficult to craft an adversarial examples with normal values of non-ME.

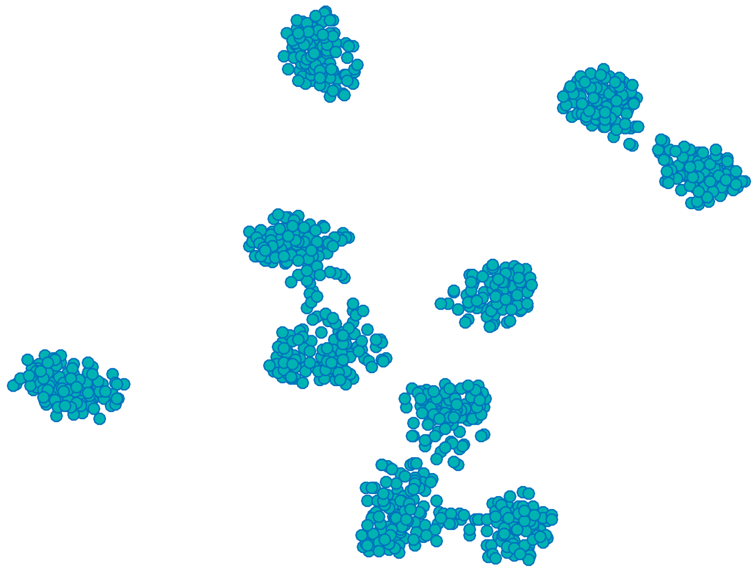
Experiments

Table 1: Classification error rates (%) on test sets.

Method	MNIST	CIFAR-10
Resnet-32 (CE)	0.38	7.13
Resnet-32 (RCE)	**0.29**	**7.02**
Resnet-56 (CE)	0.36	**6.49**
Resnet-56 (RCE)	**0.32**	6.60

Classification error rates (%) on normal test examples

Experiments

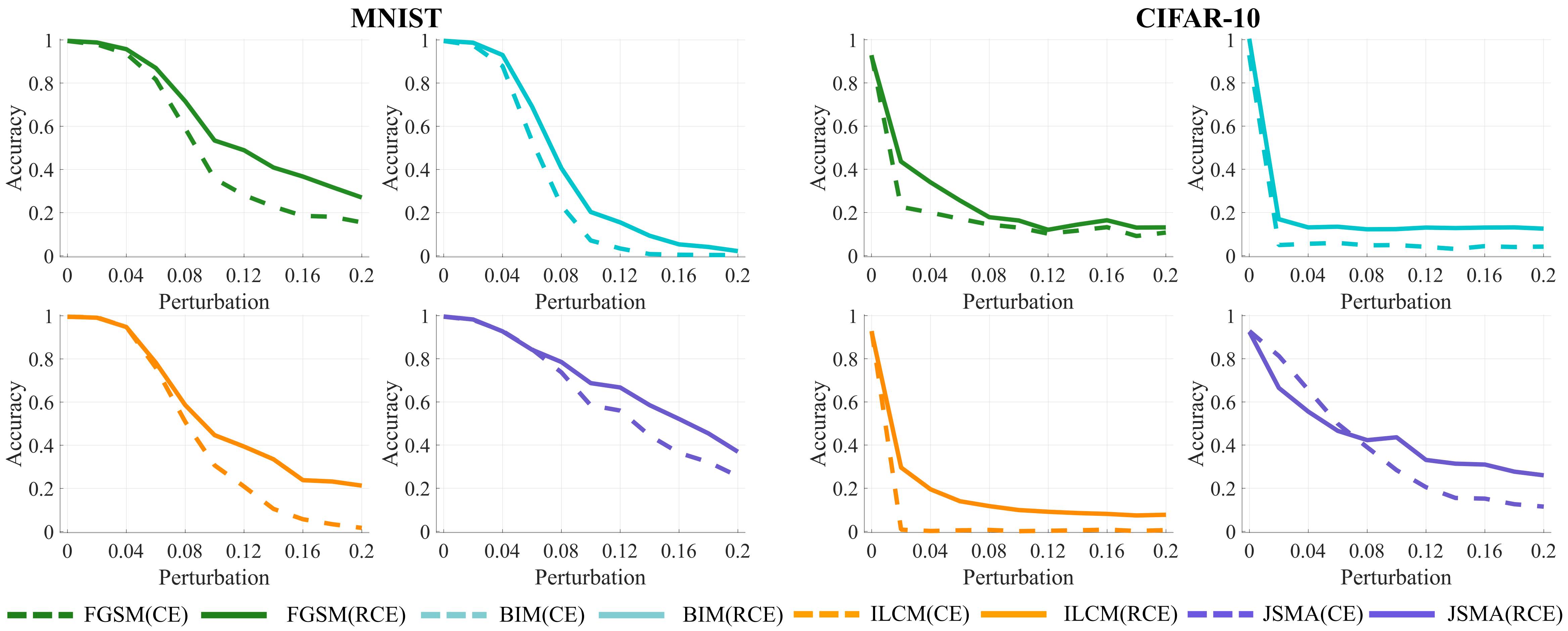


CE

RCE

t-SNE visualization of learned features on CIFAR-10

Experiments



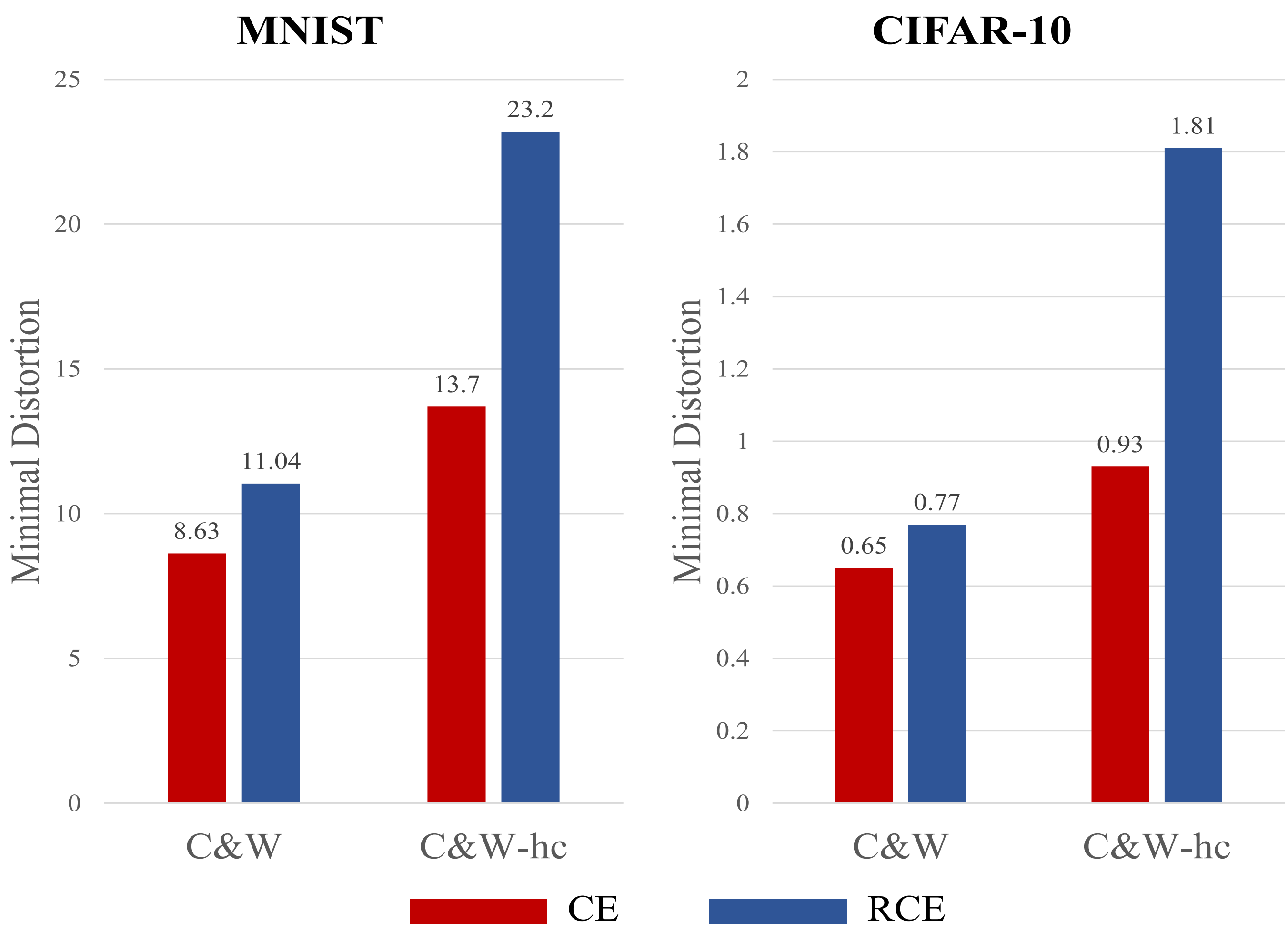
Classification accuracy under different attack methods

Experiments

Attack	Obj.	MNIST Confidence	MNIST non-ME	MNIST K-density	CIFAR-10 Confidence	CIFAR-10 non-ME	CIFAR-10 K-density
FGSM	CE	79.7	66.8	98.8 (-)	71.5	66.9	**99.7** (-)
	RCE	98.8	98.6	**99.4** (*)	92.6	91.4	98.0 (*)
BIM	CE	88.9	70.5	90.0 (-)	0.0	64.6	**100.0** (-)
	RCE	91.7	90.6	**91.8** (*)	0.7	70.2	**100.0** (*)
ILCM	CE	98.4	50.4	96.2 (-)	16.4	37.1	84.2 (-)
	RCE	100.0	97.0	**98.6** (*)	64.1	77.8	**93.9** (*)
JSMA	CE	98.6	60.1	97.7 (-)	99.2	27.3	85.8 (-)
	RCE	100.0	99.4	**99.0** (*)	99.5	91.9	**95.4** (*)
C&W	CE	98.6	64.1	99.4 (-)	99.5	50.2	95.3 (-)
	RCE	100.0	99.5	**99.8** (*)	99.6	94.7	**98.2** (*)
C&W-hc	CE	0.0	40.0	91.1 (-)	0.0	28.8	75.4 (-)
	RCE	0.1	93.4	**99.6** (*)	0.2	53.6	**91.8** (*)

AUC-scores (10^{-2}) when detecting adversarial examples

Experiments



Minimal distortion when applying the C&W attack under oblivious threat model, i.e., the adversary knows the classifier but does not know the detector

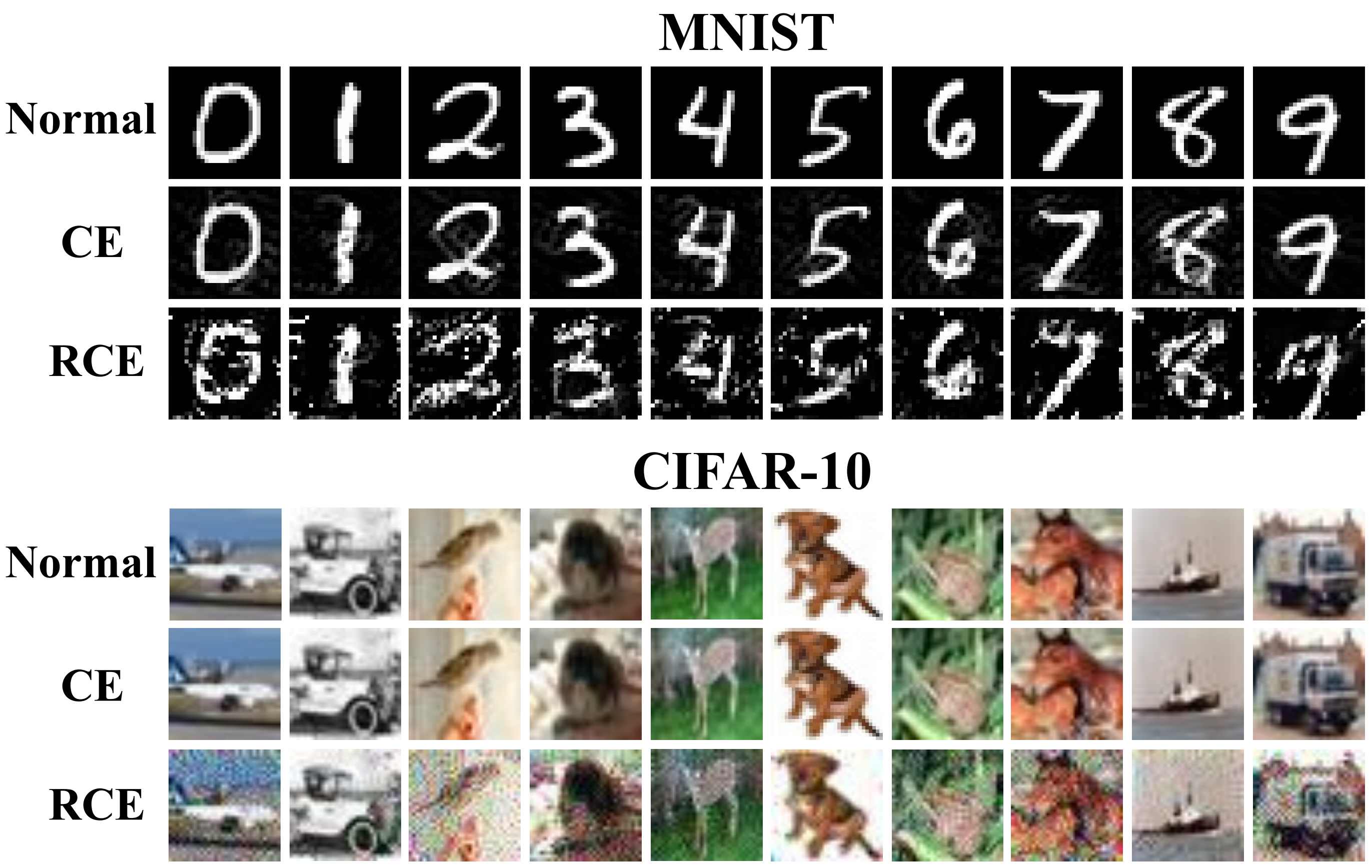
Experiments

Obj.	MNIST		CIFAR-10	
	Ratio	Distortion	Ratio	Distortion
CE	1	17.12	0	1.26
RCE	**77**	**31.59**	**12**	**3.89**

Table 3: The ratios (%) of $f_2(x^*) > 0$ and minimal distortions of the adversarial examples crafted by C&W-wb. Model is Resnet-32.

The results when apply the C&W attack under white-box threat model, i.e., the adversary also know the detector. The 'Ratio': the ratio of adversarial examples that induce higher values of detection metric than a threshold.

Experiments



The visualization of the adversarial examples crafted by white-box C&W attack

Experiments

	Res.-32 (CE)	Res.-32 (RCE)
Res.-56 (CE)	75.0	90.8
Res.-56 (RCE)	89.1	84.9

Table 4: AUC-scores (10^{-2}) on CIFAR-10. Resnet-32 is the substitute model and Resnet-56 is the target model.

AUC-scores (10^{-2}) under the black-box threat model. We use the adversarial examples crafted on Resnet-32 to feed to Resnet-56