

## A RANDOMIZED SMOOTHING

Randomized smoothing technique (Duchi et al., 2012) was originally proposed for solving convex non-smooth optimization problems. It is based on the observations that random perturbation of the optimization variable can be used to transform the loss into a smoother one. Instead of using only  $L(\mathbf{x})$  and  $\nabla L(\mathbf{x})$  to solve

$$\min L(\mathbf{x}),$$

randomized smoothing turns to solve the following objective function, which utilizes more global information from neighboring areas:

$$\min \mathbb{E}_{\xi \sim U(-1,1)} L(\mathbf{x} + u\xi), \quad (\text{A.1})$$

where  $\xi$  is a random variable, and  $u$  is a small number. Duchi et al. (2012) showed that randomized smoothing makes the loss in (A.1) smoother than before. Hence, even if the original loss  $L$  is non-smooth, it can still be solved by stochastic gradient descent with provable guarantees.

## B ADDITIONAL ABLATION STUDIES

In this section, we conduct additional ablation studies to provide a comprehensive view to the Backward Smoothing method.

### B.1 THE EFFECT OF $\beta$

We conduct the ablation studies to figure out the effect of  $\beta$  in Backward Smoothing method by fixing  $\gamma$  and the attack step size. Table 9 shows the experimental results. Similar to what  $\beta$  does in TRADES (Zhang et al., 2019), here in Backward Smoothing,  $\beta$  still controls the trade-off between natural accuracy and robust accuracy. We observe that with a larger  $\beta$ , natural accuracy keeps decreasing and the best robustness is obtained with  $\beta = 10.0$ .

Table 9: Sensitivity analysis of  $\beta$  on CIFAR-10 and CIFAR-100 datasets using ResNet-18 model.

Dataset	CIFAR-10		CIFAR-100	
	$\beta$	Nat (%) Rob (%)	Nat (%) Rob (%)	
	2.0	84.87	46.46	62.22 24.83
	4.0	84.58	50.01	59.03 27.58
	6.0	83.96	51.65	57.46 28.66
	8.0	82.48	51.88	57.51 29.38
	10.0	82.38	<b>52.50</b>	56.96 <b>30.50</b>
	12.0	81.63	52.38	56.46 29.95

### B.2 DOES BACKWARD SMOOTHING ALONE WORKS?

To further understand the role of Backward Smoothing in robust training, we conduct experiments on using Backward Smoothing alone, i.e., only use Backward Smoothing initialization but do not perform gradient-based attack at all. Table 10 and Table 11 show the experimental results. We can observe that Backward Smoothing as an initialization itself only provides a limited level of robustness (not as good as single-step attack). This is reasonable since the loss for Backward Smoothing does not directly promote adversarial attacks. Therefore it only serves as an initialization to help single-step attacks better solve the inner maximization problems.

Table 10: Performance of using Backward Smoothing alone on CIFAR-10 dataset using ResNet-18 model.

Method	Nat (%)	Rob (%)
Fast AT	84.79	46.30
Fast TRADES	84.80	46.25
Backward Smoothing Alone	69.87	39.26

Table 11: Performance of using Backward Smoothing alone on CIFAR-100 dataset using ResNet-18 model.

Method	Nat (%)	Rob (%)
Fast AT	60.35	24.64
Fast TRADES	60.22	19.40
Backward Smoothing Alone	43.47	18.51