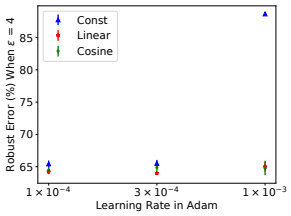
Task	ϵ Scheduler	Without Learning Rate Resets					With Periodic Learning Rate Resets				
		Clean Error (%)	Robust Error (%)				Clean Error (%)	Robust Error (%)			
			PGD100	APGD100 CE	APGD100 DLR	Square5K		PGD100	APGD100 CE	APGD100 DLR	Square5K
MNIST	Const	1.56(17)	10.86(143)	15.18(155)	14.70(136)	19.58(45)					
LeNet	Cosine	1.08(2)	8.46(82)	14.36(134)	13.46(129)	16.78(25)					
$\epsilon = 0.4$	Linear	1.06(6)	8.79(116)	13.91(150)	13.17(120)	17.05(47)					
CIFAR10	Const	28.25(47)	56.19(32)	58.18(46)	58.65(69)	54.37(29)	28.33(81)	54.16(26)	55.45(26)	56.56(4)	52.85(18)
VGG	Cosine	25.06(19)	56.00(42)	57.83(45)	58.88(16)	53.95(15)	23.91(21)	53.10(18)	54.44(16)	55.80(24)	51.41(37)
$\epsilon = 8/255$	Linear	23.56(95)	55.88(5)	57.74(16)	58.39(18)	53.66(24)	21.88(33)	52.97(17)	54.32(17)	55.63(17)	51.28(4)
CIFAR10	Const	18.62(6)	54.97(9)	57.26(13)	56.60(25)	50.59(19)	21.00(5)	48.87(25)	50.29(27)	50.98(6)	46.84(9)
ResNet18	Cosine	18.43(26)	53.85(21)	56.16(18)	55.77(24)	49.60(18)	19.90(18)	48.49(27)	49.71(22)	50.54(9)	46.19(11)
$\epsilon = 8/255$	Linear	18.55(14)	53.41(10)	55.69(17)	55.45(22)	49.66(28)	20.26(28)	48.52(13)	49.73(9)	50.68(11)	46.47(26)

Table 1: Clean and robust error on the test set under various adversarial attacks. The numbers between the brackets indicate the standard deviation across different runs. Specifically, for example, $28.25(47)$ stands for 28.25 ± 0.47.

We thank the reviewers for their constructive comments. Below, we first address the concerns raised by several reviewers regarding the experimental evaluation, and then provide point-to-point responses to each reviewer.

The table above shows that **our PAS strategy still yields better performance under stronger attacks**. As suggested by reviewer 1, we first evaluate our models using 100-iteration PGD with 10 restarts (PGD100). To solve the issue of suboptimal step size, we also evaluate our models using the state-of-the-art AutoPGD attack [Croce ICML20], which searches for the optimal step size. We run AutoPGD for 100 iterations, based on either the cross-entropy loss (APGD100-CE) or the difference of logit ratio loss (APGD100-DLR). For black-box attacks, we run the state-of-the-art SquareAttack [Andriushchenko ECCV20] for 5000 iterations (Square5K).

Furthermore, we would like to emphasize that the main focus of our work is optimization. In this regard, **one main advantage of PAS is to avoid convergence failure and make the optimization less sensitive to the learning rate**, as shown in Figure 4. The figure on the right shows the same observation with l_2 attacks. Specifically, under an l_2 adversarial budget $\epsilon = 4$ on MNIST (as in [Madry ICLR18]), adversarial training with a constant ϵ fails to converge when using a high learning rate in Adam, e.g., 10^{-3}. By contrast, a linear/cosine scheduler is more robust to the choice of learning rate and yields consistently better performance.



Reviewer #1: The requested additional experiments are presented above. *ReLU networks:* We discuss ReLU networks in detail in Appendix A.3. Corollary 1 shows that gradient scattering is more severe under larger ϵ for ReLU networks. Gradient scattering is measured as the first-order gradient difference, i.e., $\|\nabla_\theta \mathcal{L}_\epsilon(\theta_1) - \nabla_\theta \mathcal{L}_\epsilon(\theta_2)\|$, whose upper bound increases with ϵ for any activation function. *Experimental settings:* The step size of PGD follows [Ye ICCV19] and is the same as in the seminal work [Madry ICLR18]. *Comparison with challenges:* Both challenges are leaderboards for attacks, not defenses. The architecture of our MNIST models are the same as the ones in the challenge. When $\epsilon = 0.3$, our model has better robust accuracy than the standard defense model provided in this challenge: our model yields a robust accuracy of 95.08% / 93.01% under PGD100 / PGD100 with 50 restarts, whereas the corresponding accuracy for the provided model is 92.52% / 89.62%. We will release our trained model for public testing.

Reviewer #2: More experiments are provided in the general response. *Error definition:* Yes, they are 1-accuracy. *Proposition 1:* Following the definition in Equation 1, $g_\epsilon(\mathbf{x}, \mathbf{W})$ is the adversarial loss of the point $\mathbf{x}$ under the adversarial budget $\mathcal{S}_\epsilon(\mathbf{x})$. We will revise the proof to make it clearer. *$\mathcal{V}_\epsilon$ and $\mathcal{T}_\epsilon$ in binary cases:* Thanks for pointing this out. We realized that this claim is incorrect and will remove it. However, this does not affect our other claims. *Assumption on page 4:* Precisely, our assumption is $\nabla_\theta g(\mathbf{x}_1, \theta) \neq \nabla_\theta g(\mathbf{x}_2, \theta)$ when $\mathbf{x}_1 \neq \mathbf{x}_2$. This assumption is based on the *clean* loss function g and is true in general for deep neural networks. By contrast, the conclusion involves the adversarial loss, and refers to the discontinuity of the parameter gradients, not the fact of having different parameter gradients for different inputs. *Tightness of Proposition 2's bound:* We will move this claim to the main text. *Linear model experiments:* We will add this to validate the theorem. *IBP-based local Lipschitz constant:* IBP or convex relaxation calculate the upper bound U_ϵ and the lower bound I_ϵ of the adversarial loss: $I_\epsilon(\mathbf{x}, \theta) \leq g_\epsilon(\mathbf{x}, \theta) \leq U_\epsilon(\mathbf{x}, \theta)$. These bounds are computed based on the adversarial budget defined in the input space, whereas the Lipschitz constant is defined in the parameter space. Therefore, the curvature of these bounds, i.e., $\nabla^2_\theta I_\epsilon$ and $\nabla^2_\theta U_\epsilon$, does not provide a bound for $\nabla^2_\theta g_\epsilon$. We believe that calculating a tight guaranteed bound of the Hessian eigenvalues is non-trivial. *Clarity:* Thanks, we will revise as suggested. *Related work:* Note that we cite [Wong ICLR20], but their method differs from vanilla FGSM [Goodfellow ICLR14] by using a random starting point, while FGSM uses a fixed one. Regarding over-regularization, we agree that adversarially-trained models have considerably lower clean accuracy than the ones trained using the clean data. However, as observed in [Zhang ICLR20], the models trained by some provably-robust methods, such as convex relaxations, have even lower clean accuracy than those trained by PGD. We will clarify this.

Reviewer #3: We have included more results in the general response. We choose datasets commonly used in the literature to facilitate comparisons. We are not aware of any work reporting adversarial training results on ImageNet.

Reviewer #4: More experimental results are provided in the general response. We cite [Li NIPS17] and visualize the loss landscape in Figure 12 of Appendix C.2.2. In contrast to [Li NIPS17], however, we explore the neighborhood in the directions of the top two Hessian eigenvalues, which more clearly shows the sharpness of the loss landscape.