Boosting Adversarial Robustness From The Perspective of Effective Margin Regularization: Appendix

BMVC 2022 Submission # 367

A Experimental Settings

Model Architecture. Section 3 uses an MLP and CNN to demonstrate the effective margin maximization effect of EMR. The MLP has 4 hidden layers with 1024 hidden neurons and one output layer. The CNN has an architecture as in Table S1, where the parameter in the convolution layer means CONV(kernel_size, output_channel, stride, padding) and in the average pooling layer means Average_Pooling(kernel_size, stride). For both models we use ReLU as the activation function.

Layer 1	CONV(5,32,1,padding=None)
Layer 2	CONV(5,64,1,padding=None)
Layer 3	Average_Pooling(2,2,padding=None)
Layer 4	CONV(3,128,1,padding=None)
Layer 5	CONV(3,128,1,padding=None)
Layer 6	Global_Average_Pooling
Layer 6	Linear(128,10)

Table S1: The architecture of CNN in Section 3.

Hyperparameters. We searched the hyperparameters for the adversarial training baselines and our method, Table S2 and S3 show the hyperparameters used in the experiment we report. In MAIL, we use the WideResNet-34-10 and default settings in the MAIL loss [2]. AT-MAIL has a slope of 30 and bias of 0.07 . TRADES-MAIL has a slope of 5 and bias of 0.05, where β=5.0. All MAIL experiment uses a weight decay of 2e-4. In AT-MAIL-ERM, λ_{EMR}=1.0, t=1.0. In TRADES-MAIL-ERM, λ_{EMR}=3.0, t=1.0. We use PGD10 attack with a step size of 0.00784 during adversarial training of MAIL and the initial learning rate 0.1 is decayed by 10 at 75 and 90 epoch, with a maximum epoch of 100. In MART+EMR [5], we use the default setting of MART loss from the official code, where $\beta = 6.0$, the step size of PGD attack is 0.007 and step is 10. With the MART loss, the WideResNet-34-10 is trained for 90 epochs and the initial learning rate 0.1 is divided by 10 at 75 and 90 epoch, and we let $\lambda_{EMR} = t = 1.0$. In LBGAT+EMR [1], we use the default setting in the official code, where

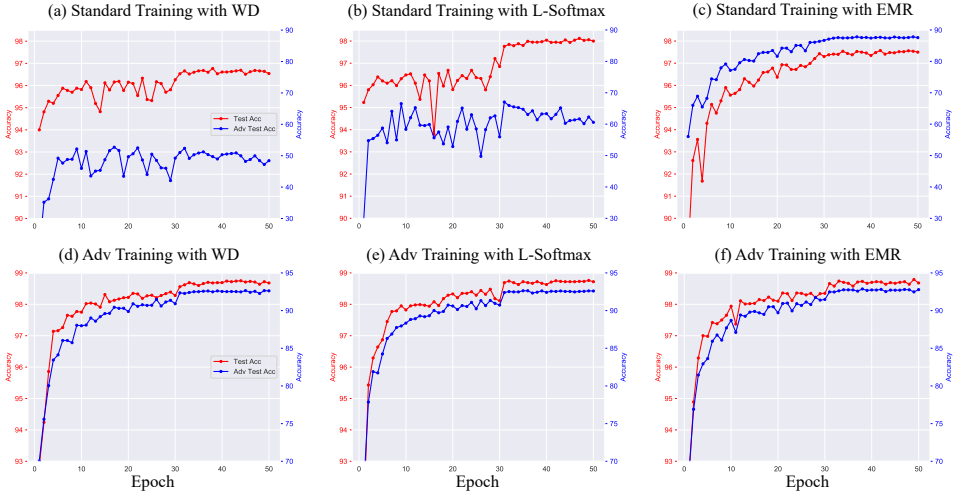

Figure S1: Training curve on MNIST with 4-hidden layer MLP. In normal training, EMR achieves significantly higher robust test accuracy than two baselines and is even comparable with that of adversarial training.

the teacher model is a randomly initialized ResNet18, the student model is a WideResNet34-10, β =0.0 (without TRADES) and step size of PGD10 is 0.003. We train the model for 100 epochs and divide the initial learning rate 0.1 by 10 at 76 and 91 epoch, and let $\lambda_{EMR} = 0.5$ and t =1. For the MART and LBGAT baseline, we evaluate the official models released by the authors. Note that we use the l_∞-bound adversarial examples with an $\varepsilon = 0.031$ in training and evaluation of all experiment. The algorithm for adversarial training with EMR is shown in Algorithm 1. We include the code to use our method in the supplemental.

B More Experiment Result

Fig. S2 and Fig. S1 show the clean and robust test accuracy curves during CNN and MLP training. For the MLP, EMR achieves a high robust accuracy at an early stage of training. For the CNN, EMR shows an oscillation of robust test accuracy at an early stage, but the accuracy becomes stable when the learning rate is decayed.

Table S4 shows the performance of the approximate EMR (Approx-EMR) on CNN and MLP with ST and AT. The approximation achieves a comparable performance in both models and even better robustness in the CNN trained with AT. Table S5 shows the result of vanilla AT, IGR and our EMR on ResNet18. Compared with AT and IGR, the adversarial robustness (as measured by AutoAttack) is substantially improved when EMR is used. For TRADES, the improvement is not as significant as with AT. Note that IGR does not improve the performance substantially for either AT or TRADES.

To evaluate the parameter sensitivity of EMR, we evaluate the robustness of ResNet18 when selecting the hyperparameters with grid search. Fig. S3 shows the robust accuracy of different combinations of temperature t and λ_{EMR}. In AT, selecting a large temperature generally improves the performance.

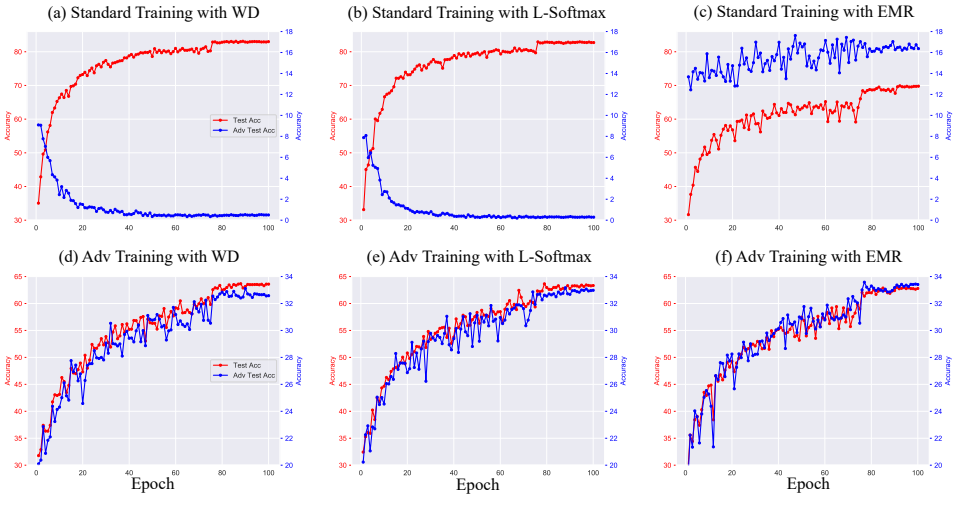

Figure S2: Training curve on CIFAR10 with 4-hidden layer CNN. In standard training, EMR has a higher robust test accuracy than training with WD or L-Softmax. In adversarial training, EMR also achieves the best robust accuracy.

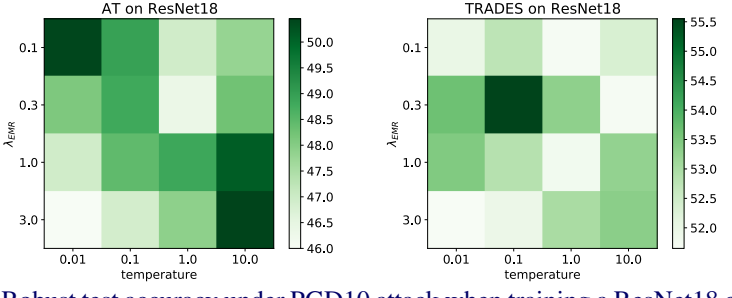

Figure S3: Robust test accuracy under PGD10 attack when training a ResNet18 on CIFAR10 with different hyperparameters.

Finally, we evaluate the computational time of EMR. Table S6 shows a comparison between training time of one SGD update of EMR, IGR, HE and the vanilla AT/TRADES when a WideResNet is used. EMR and IGR have approximately the same computational time that is longer than HE, since the loss requires an extra backpropagation. HE has a benefit in the computational time during training, but the normalization layers bring extra computation for the *inference* stage. In contrast, EMR does not need more computation during inference.

Algorithm 1: Adv. Training with Effective Margin Regularization

Input: Training data $\mathcal{D}_{tr}$, EMR parameter λ_{EMR}, EMR temperature t, learning rate η, beta of TRADES β
Output: Model parameters $\boldsymbol{\theta}$
Initialize model parameters;
for $i = 1, \ldots, N_e$ **do**
  Adjust η and λ_{EMR};
  Split $\mathcal{D}_{tr}$ into N_B =ceil(N_{tr}/B) batches;
  for $b = 1, \ldots, N_B$ **do**
    Generate adversarial examples $\{\tilde{\boldsymbol{x}}_i, y_i\}_{i=1}^{B}$;
    if *AT* **then**
      $\mathcal{L} = \frac{1}{B}\sum_{i=1}^{B} XE(f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}_i), y_i)$;
    end
    if *TRADES* **then**
      $\mathcal{L} = \frac{1}{B}\sum_i XE(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), y_i) + \beta\frac{1}{B}\sum_i \mathcal{D}_{KL}(f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}_i), f_{\boldsymbol{\theta}}(\boldsymbol{x}_i))$;
    end
    % EMR:
    $\boldsymbol{p}_i$=softmax$(f_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}_i)/t)$ and detach the gradient;
    $\mathcal{L}_{EMR} = \frac{1}{B}\sum_i^B \|\nabla_{\boldsymbol{x}} \sum_{j=1}^{K} p_{ij} l_{ij}(\tilde{\boldsymbol{x}}_i)\|_2^2$ **(eval mode)**;
    $\boldsymbol{\theta} := \boldsymbol{\theta} - \eta\nabla_{\boldsymbol{\theta}}(\mathcal{L} + \lambda_{EMR}\mathcal{L}_{EMR})$
  end
end

	Hyperparameters
AT	λ_{WD}=1e-3
AT+IGR	λ_{WD}=1e-3, λ_{IGR}=1.0
AT+EWR	λ_{WD}=5e-4, λ_{EMR}=0.1, t=40.0
TRADES	β=12.0, λ_{WD}=5e-4
TRADES+IGR	β=12.0, λ_{WD}=5e-4, λ_{IGR}=1.0
TRADES+EWR	β=12.0, λ_{WD}=5e-4, λ_{EMR}=0.3, t=0.1

Table S2: Hyperparameters of ResNet18.

	Hyperparameters
AT	λ_{WD}=1e-3
AT+IGR	λ_{WD}=1e-3, λ_{IGR}=1.0
AT+EWR	λ_{WD}=5e-4, λ_{EMR}=1.0, t=40.0
TRADES	β=12.0, λ_{WD}=5e-4
TRADES+IGR	β=12.0, λ_{WD}=5e-4, λ_{IGR}=1.0
TRADES+EWR	β=12.0, λ_{WD}=5e-4, λ_{EMR}=0.3, t=1.0

Table S3: Hyperparameters of WideResNet.

Model	Training	λ_{WD}	Clean Acc.	PGD	$\tilde{m}_{train}$	$\tilde{m}_{test}$
MLP	ST+EMR$_{0.1}$	0.001	97.50	**87.56**	4.41±1.24	2.24±0.98
	ST+Approx-EMR$_{1.0}$	0.001	**98.44**	77.09	0.85±0.71	1.73±0.73
	AT+EMR$_{0.0003}$	0.001	98.68	**92.78**	3.83±1.27	2.42±0.99
	AT+Approx-EMR$_{0.0003}$	0.001	**98.75**	92.76	4.00±1.30	2.35±0.94
CNN	ST+EMR$_{0.01}$	0.0005	69.78	**16.37**	0.66±0.48	0.70±0.50
	ST+Approx-EMR$_{30.0}$	0.0005	**71.09**	15.05	0.65±0.48	0.68±0.50
	AT+EMR$_{0.001}$	0.0005	62.79	33.41	0.74±0.64	1.08±0.83
	AT+Approx-EMR$_{0.0003}$	0.0005	**63.15**	**33.63**	0.73±0.62	1.07±0.81

Table S4: Comparison between EMR and its large-scale approximation.

	Clean Acc.	FGSM	PGD10	PGD100	AutoAttack
AT	83.39	56.95	50.88	50.07	46.90
AT+IGR[4]	**84.01**	56.97	51.03	49.66	46.52
AT+EWR (ours)	81.71	56.39	51.97	51.18	**47.94**
TRADES	**79.68**	57.62	53.67	53.00	48.56
TRADES+IGR[4]	78.61	57.34	53.70	53.08	48.48
TRADES+EWR (ours)	79.59	57.43	53.53	53.01	**48.86**

Table S5: Evaluation of adversarial robustness using ResNet18 on CIFAR10.

	Standard	IGR[4]	HE[3]	EMR(ours)
AT	2.46(.01)	3.21(.01)	2.75(.01)	3.22(.01)
TRADES	2.93(.01)	3.67(.01)	3.32(.01)	3.67(.01)

Table S6: Running time (in second) of one SGD iteration. The time is recorded with WideResNet-34-10 on a Nvidia-V100 with a batch size of 128.

References

[1] Jiequan Cui, Shu Liu, Liwei Wang, and Jiaya Jia. Learnable boundary guided adversarial training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15721–15730, 2021.

[2] Feng Liu, Bo Han, Tongliang Liu, Chen Gong, Gang Niu, Mingyuan Zhou, Masashi Sugiyama, et al. Probabilistic margins for instance reweighting in adversarial training. *Advances in Neural Information Processing Systems*, 34, 2021.

[3] Tianyu Pang, Xiao Yang, Yinpeng Dong, Kun Xu, Jun Zhu, and Hang Su. Boosting adversarial training with hypersphere embedding. *Advances in Neural Information Processing Systems*, 33:7779–7792, 2020.

[4] Andrew Ross and Finale Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[5] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*, 2019.