Regularizing Neural Networks via Adversarial Model Perturbation
Supplementary Material

Yaowei Zheng
BDBC and SKLSDE
Beihang University, China
hiyouga@buaa.edu.cn

Richong Zhang*
BDBC and SKLSDE
Beihang University, China
zhangrc@act.buaa.edu.cn

Yongyi Mao
School of EECS
University of Ottawa, Canada
ymao@uottawa.ca

A. Proof of Corollary 2

Corollary 2. *Suppose that $C_1 = C_2$. Let $A_2 = \beta A_1$ for some $\beta < 1$. Note that in this setting, $\gamma_1^* < \gamma_2^*$. Suppose that $\sigma_2^2 = r\sigma_1^2$ for some positive r. Then*

$$\gamma^*_{2,\mathrm{AMP}} < \gamma^*_{1,\mathrm{AMP}}$$

if and only if

$$\beta > \exp\left(-\frac{\epsilon^2}{2\sigma_1^2}\right) \text{ and } r > \frac{1}{1 + \frac{2\sigma_1^2}{\epsilon^2}\log\beta}$$

Proof. Since $C_1 = C_2$, we have

$$0 > \exp\left(-\frac{\epsilon^2}{2\sigma_1^2}\right) - \beta\exp\left(-\frac{\epsilon^2}{2r\sigma_1^2}\right)$$

It then follows that

$$\frac{\epsilon^2}{2r\sigma_1^2} < \frac{\epsilon^2}{2\sigma_1^2} + \log\beta \tag{1}$$

noting that $\log\beta < 0$, we have $r > 1$.

Further manipulating (1), we get

$$\frac{1}{r} < 1 + \frac{2\sigma_1^2}{\epsilon^2}\log\beta \tag{2}$$

Since $r > 0$, the right side of (2) is positive, which gives rise to

$$\beta > \exp\left(-\frac{\epsilon^2}{2\sigma_1^2}\right)$$

Continuing with (2), we arrive at

$$r > \frac{1}{1 + \frac{2\sigma_1^2}{\epsilon^2}\log\beta}$$

This proves the result. □

B. Proof of Theorem 2

Theorem 2. *Let $N = 1$. Then for a sufficiently small inner learning rate ζ, a minimization update step in AMP training for batch $\mathcal{B}$ is equivalent to a gradient-descent step on the following loss function with learning rate η:*

$$\widetilde{\mathcal{J}}_{\mathrm{ERM},\mathcal{B}}(\boldsymbol{\theta}) := \mathcal{J}_{\mathrm{ERM},\mathcal{B}}(\boldsymbol{\theta}) + \Omega(\boldsymbol{\theta})$$

where

$$\Omega(\boldsymbol{\theta}) := \begin{cases} \zeta\|\nabla_{\boldsymbol{\theta}}\mathcal{J}_{\mathrm{ERM},\mathcal{B}}(\boldsymbol{\theta})\|_2^2, & \|\zeta\nabla_{\boldsymbol{\theta}}\mathcal{J}_{\mathrm{ERM},\mathcal{B}}(\boldsymbol{\theta})\|_2 \le \epsilon \\ \epsilon\|\nabla_{\boldsymbol{\theta}}\mathcal{J}_{\mathrm{ERM},\mathcal{B}}(\boldsymbol{\theta})\|_2, & \|\zeta\nabla_{\boldsymbol{\theta}}\mathcal{J}_{\mathrm{ERM},\mathcal{B}}(\boldsymbol{\theta})\|_2 > \epsilon \end{cases}$$

Proof. At each training step of AMP, we adversarially perturb the parameter with a step size of ζ. If the norm of perturbation is larger than a preset value ϵ, it will be projected onto the L$_2$-norm ball. Denoted by $\boldsymbol{\theta}_k$ the model parameter at the k-th iteration, the perturbed parameter is:

$$\boldsymbol{\theta}_{k,\mathrm{adv}} = \begin{cases} \boldsymbol{\theta}_k + \zeta\nabla_{\boldsymbol{\theta}}\mathcal{J}_{\mathrm{ERM},\mathcal{B}}(\boldsymbol{\theta}_k), & \|\zeta\nabla_{\boldsymbol{\theta}}\mathcal{J}_{\mathrm{ERM},\mathcal{B}}(\boldsymbol{\theta}_k)\|_2 \le \epsilon \\ \boldsymbol{\theta}_k + \epsilon\frac{\nabla_{\boldsymbol{\theta}}\mathcal{J}_{\mathrm{ERM},\mathcal{B}}(\boldsymbol{\theta}_k)}{\|\nabla_{\boldsymbol{\theta}}\mathcal{J}_{\mathrm{ERM},\mathcal{B}}(\boldsymbol{\theta}_k)\|_2}, & \|\zeta\nabla_{\boldsymbol{\theta}}\mathcal{J}_{\mathrm{ERM},\mathcal{B}}(\boldsymbol{\theta}_k)\|_2 > \epsilon \end{cases}$$

Then the parameter is updated according to the gradient computed by the perturbed parameter with a step size of η:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \eta\nabla_{\boldsymbol{\theta}}\mathcal{J}_{\mathrm{ERM},\mathcal{B}}(\boldsymbol{\theta}_{k,\mathrm{adv}})$$

With a sufficient small ζ, we can utilize the first-order Taylor expansion $f(\boldsymbol{x} + \boldsymbol{\delta}) \approx f(\boldsymbol{x}) + \boldsymbol{\delta}^T\nabla_{\boldsymbol{x}}f(\boldsymbol{x})$. In the former condition (i.e. $\|\zeta\nabla_{\boldsymbol{\theta}}\mathcal{J}_{\mathrm{ERM},\mathcal{B}}(\boldsymbol{\theta})\|_2 \le \epsilon$), we have:

$$\begin{aligned} \boldsymbol{\theta}_{k+1} &= \boldsymbol{\theta}_k - \eta\nabla_{\boldsymbol{\theta}}\mathcal{J}_{\mathrm{ERM},\mathcal{B}}\left(\boldsymbol{\theta}_k + \zeta\nabla_{\boldsymbol{\theta}}\mathcal{J}_{\mathrm{ERM},\mathcal{B}}(\boldsymbol{\theta}_k)\right) \\ &\approx \boldsymbol{\theta}_k - \eta\nabla_{\boldsymbol{\theta}}\left(\mathcal{J}_{\mathrm{ERM},\mathcal{B}}(\boldsymbol{\theta}_k) + \zeta\|\nabla_{\boldsymbol{\theta}}\mathcal{J}_{\mathrm{ERM},\mathcal{B}}(\boldsymbol{\theta}_k)\|_2^2\right) \end{aligned}$$

In the latter condition (i.e. $\|\zeta\nabla_{\boldsymbol{\theta}}\mathcal{J}_{\mathrm{ERM},\mathcal{B}}(\boldsymbol{\theta})\|_2 > \epsilon$), we have:

$$\begin{aligned} \boldsymbol{\theta}_{k+1} &= \boldsymbol{\theta}_k - \eta\nabla_{\boldsymbol{\theta}}\mathcal{J}_{\mathrm{ERM},\mathcal{B}}\left(\boldsymbol{\theta}_k + \epsilon\frac{\nabla_{\boldsymbol{\theta}}\mathcal{J}_{\mathrm{ERM},\mathcal{B}}(\boldsymbol{\theta}_k)}{\|\nabla_{\boldsymbol{\theta}}\mathcal{J}_{\mathrm{ERM},\mathcal{B}}(\boldsymbol{\theta}_k)\|_2}\right) \\ &\approx \boldsymbol{\theta}_k - \eta\nabla_{\boldsymbol{\theta}}\left(\mathcal{J}_{\mathrm{ERM},\mathcal{B}}(\boldsymbol{\theta}_k) + \epsilon\|\nabla_{\boldsymbol{\theta}}\mathcal{J}_{\mathrm{ERM},\mathcal{B}}(\boldsymbol{\theta}_k)\|_2\right) \end{aligned}$$

This proves the theorem. □

[*] Corresponding author

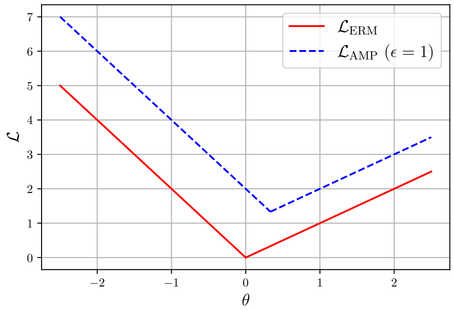


Figure 1: The losses of ERM and AMP with varying θ.

C. Why AMP is not Adversarial Training

In this section, we will further discuss the difference between AMP and adversarial training (ADV).

It is sensible that perturbing weights $\boldsymbol{\theta}$ may have an effect similar to perturbing the examples $\boldsymbol{x}$ since $\boldsymbol{\theta}$ and $\boldsymbol{x}$ usually appear together via inner product $\boldsymbol{\theta}^\top \boldsymbol{x}$. However we note that except for some peculiar cases (such as linear network with some peculiar choices of the loss function or a set of peculiarly constructed training examples), in general the solution $\boldsymbol{\theta}^*_{\rm AMP}$ to the AMP optimization problem is different from the solution $\boldsymbol{\theta}^*_{\rm ADV}$ to the ADV counterpart. The difference between $\boldsymbol{\theta}^*_{\rm AMP}$ and $\boldsymbol{\theta}^*_{\rm ADV}$ can be attributed to two sources.

First, let $\ell(\boldsymbol{x};\boldsymbol{\theta})$ denote the ERM loss for a single training example $\boldsymbol{x}$. For N examples, the overall ERM loss $\mathcal{L}_{\rm ERM}$ is the sum (or average) of $\ell(\boldsymbol{x}_i;\boldsymbol{\theta})$ over all examples $\boldsymbol{x}_i$, $i = 1, \ldots, N$. In AMP, the perturbation is to maximize the *overall* empirical loss $\mathcal{L}_{\rm ERM}$ and this perturbation is applied *globally* to weights $\boldsymbol{\theta}$. However, in ADV, the perturbation is applied *individually* to *each* training example x_i, with the objective of maximizing the *individual* ERM loss $\ell(\boldsymbol{x}_i;\boldsymbol{\theta})$.

Second, even in the case when there is only one training example $\boldsymbol{x}$ so that $\mathcal{L}_{\rm ERM} = \ell$, $\boldsymbol{\theta}^*_{\rm AMP}$ and $\boldsymbol{\theta}^*_{\rm ADV}$ may still be different. Here is an example. Let

$$g(z) := \begin{cases} z & \text{if } z \geq 0 \\ -2z & \text{if } z < 0 \end{cases}$$

Consider that there is a single scalar example $x = 1$ and the weight θ is a scalar. Define $\ell(x;\theta) = g(\theta x)$. It can be verified that $\theta^*_{\rm ADV} = \theta^*_{\rm ERM} = 0$ regardless of the perturbation radius ϵ, but $\theta^*_{\rm AMP} = \epsilon/3$ (see Figure 1, where the losses are plotted as functions of θ).

D. Definition of Expected Calibration Error

We follow the definition presented in the previous work [2]. Firstly, the predictions are grouped into M interval bins of equal sizes. Let B_m be the set of indices of samples whose prediction scores (the winning softmax score) fall into the interval $I_m = (\frac{m-1}{M}, \frac{m}{M}]$. The accuracy and confidence of B_m are defined as:

$$\text{acc}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \mathbf{1}(\hat{y}_i = y_i)$$

$$\text{conf}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \hat{p}_i$$

where $\hat{y}_i$ and y_i are the predicted label and true class labels for sample i, $\hat{p}_i$ is the confidence (the winning softmax score) of sample i. The *Expected Calibration Error* (ECE) is defined as the difference in expectation between confidence and accuracy, i.e.:

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} \Big| \text{acc}(B_m) - \text{conf}(B_m) \Big|$$

where n is the number of samples.

FGSM	SVHN	CIFAR-10	CIFAR-100
ERM	23.41±0.569	36.06±1.908	68.78±0.699
Dropout	22.36±0.591	34.13±0.844	64.70±0.549
Label Smoothing	17.74±1.674	23.24±0.427	57.30±0.410
Flooding	17.40±0.656	36.42±1.303	68.45±0.407
MixUp	19.95±0.637	25.82±0.384	65.90±0.498
Adv. Training	**14.33±0.200**	**18.58±0.304**	**48.51±0.260**
RMP	23.73±0.965	35.40±0.572	68.52±0.515
AMP	16.82±1.561	28.61±0.359	59.04±1.325
PGD	**SVHN**	**CIFAR-10**	**CIFAR-100**
ERM	45.17±1.085	58.88±2.296	85.46±0.770
Dropout	41.76±1.346	55.21±1.088	78.46±1.081
Label Smoothing	32.55±2.005	34.93±0.443	65.31±0.700
Flooding	33.50±1.707	60.32±1.393	84.66±0.285
MixUp	75.75±2.129	62.77±1.018	89.58±0.596
Adv. Training	**20.20±0.409**	**21.46±0.373**	**51.72±0.327**
RMP	44.74±0.960	58.06±0.650	84.80±0.488
AMP	25.15±1.942	49.72±0.785	73.95±2.608

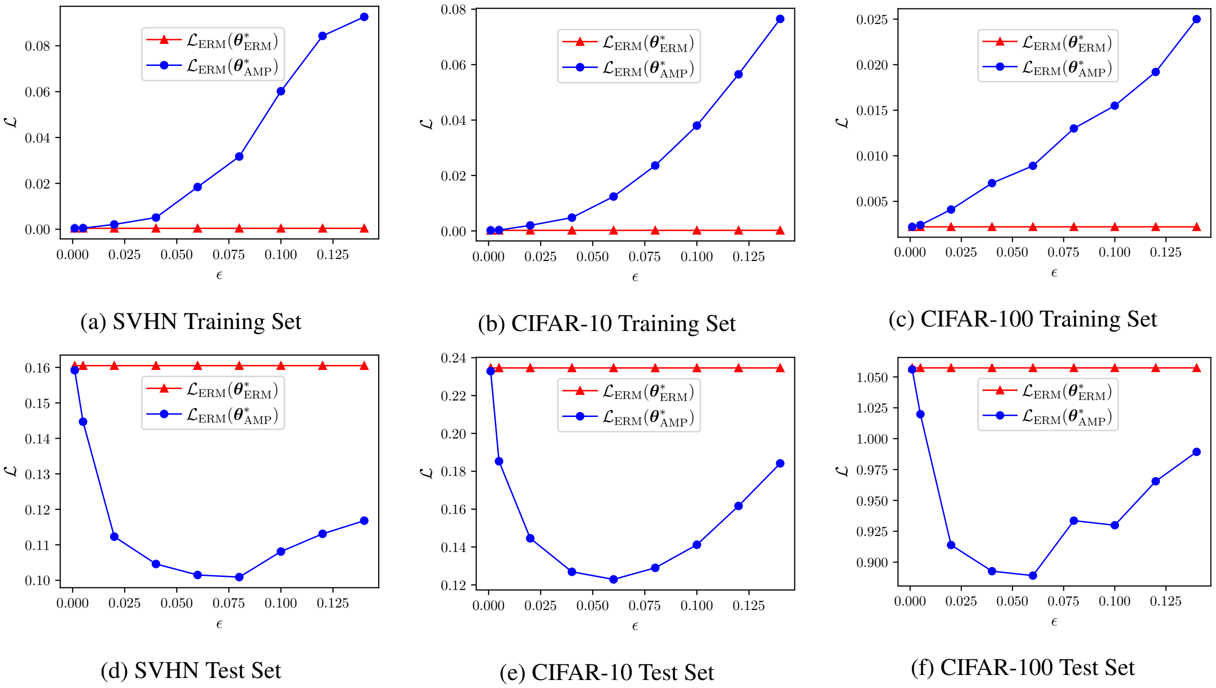
Table 1: Test errors (%) against the while-box FGSM and PGD adversarial attacks. Each experiment has been run ten times to report the mean and standard derivation of errors.

E. Influence of Perturbation

We plot the empirical risks of the pretrained PreActResNet18 models on three image datasets with varying perturbation radius in Figure 2. To clearly illustrate this, we adopt $\eta = 2$ and $N = 2$. In these experiments, the perturbation radius ϵ meets the sweet spots around 0.06 on all the three datasets, where $\mathcal{L}_{\rm ERM}(\boldsymbol{\theta}^*_{\rm AMP})$ gets the minimum value.

F. Robustness to Adversarial Attacks

The previous work [5] suggests that the flat minima make the adversarial attacks take more efforts for the input to leave the minima, so AMP is expected to improve the model's



(a) SVHN Training Set

(b) CIFAR-10 Training Set

(c) CIFAR-100 Training Set

(d) SVHN Test Set

(e) CIFAR-10 Test Set

(f) CIFAR-100 Test Set

Figure 2: The comparison of $\mathcal{L}_{\mathrm{ERM}}$ of the models trained with ERM (red) and AMP (blue) with varying perturbation radius.

adversarial robustness. To validate this, we use the models trained with different regularization schemes to evaluate their adversarial robustness against the Fast Gradient Sign Method (FGSM) [1] and Projected Gradient Decent (PGD) [4] attacks. For FGSM, we set the perturbation radius to 4 per pixel. For PGD, we set the step size to 1 and perform 10 steps to generate adversarial examples, the perturbation radius is the same as FGSM. PreActResNet18 is chosen as the model architecture. We report the top-1 classification error on the adversarial examples constructed from the test set in Table 1. From the results, adversarial training outperforms all other schemes, since it directly trains models on the adversarial examples. AMP and label smoothing also show an effect in improving the model's robustness against both single-step FGSM attack and multi-step PGD attack.

G. Loss Curve

To investigate the mechanism of different regularization schemes in the training course, we plot the evolution curves of the training loss and the test loss in Figure 3 using PreActResNet18. We select ERM and two representative analogues (Flooding and MixUp) which achieved the second-best performance in the previous experiment to compare with AMP. From Figure 3, ERM obtains the smallest training loss, and MixUp retains a high training loss since it trains models on augmented examples. AMP injects a small perturbation into the model parameter, and hence the training loss is slightly increased. It appears that the Flooding scheme affects training only when the training loss drops to a very low value, whereas MixUp and AMP take effects much earlier. For the test loss, AMP converges at a similar speed as other schemes, and reduces the test loss to a smaller value at the final stage.

H. Flatness of Selected Minima

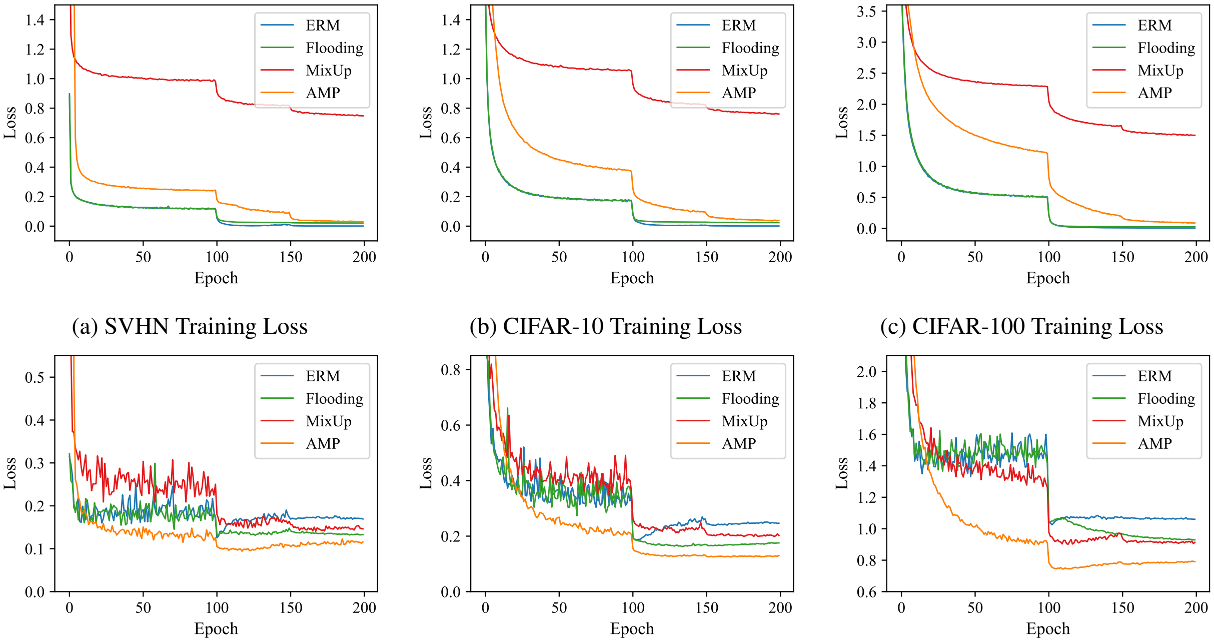
We visualize the landscapes around the minima of the empirical risk selected by ERM or AMP, the 2D views are plotted in Figure 4 and the 3D views are in Figure 5. Specifically, we compute the empirical risks of the PreActResNet18 models whose parameter is perturbed along two random directions $\boldsymbol{d}_x, \boldsymbol{d}_j$ with different step sizes δ_x, δ_y, where the direction vectors are normalized by the norm of filters suggested by [3]. Specifically, we visualize the landscapes by computing

$$\mathcal{L}_{\mathrm{ERM}}(\boldsymbol{\theta}^* + \delta_x \boldsymbol{d}_x + \delta_y \boldsymbol{d}_y)$$

The results suggest that AMP indeed selects flatter minima via adversarial perturbations.

I. Computing Environment and Resources

Our PyTorch code is executed in a CUDA environment. When evaluated on a single Tesla V100 GPU, the code takes around 2.4 hours to train a PreActResNet18 model with ERM on the CIFAR-10 dataset, and around 4.2 hours with AMP. The computation time mainly depends on the number of inner iterations, the number of epochs, and the number of GPUs. The code and datasets for reproduction can be found at https://github.com/hiyouga/AMP-Regularizer.



(a) SVHN Training Loss

(b) CIFAR-10 Training Loss

(c) CIFAR-100 Training Loss

(d) SVHN Test Loss

(e) CIFAR-10 Test Loss

(f) CIFAR-100 Test Loss

Figure 3: Loss curves for PreActResNet18 with different regularization schemes on three benchmark image datasets.

References

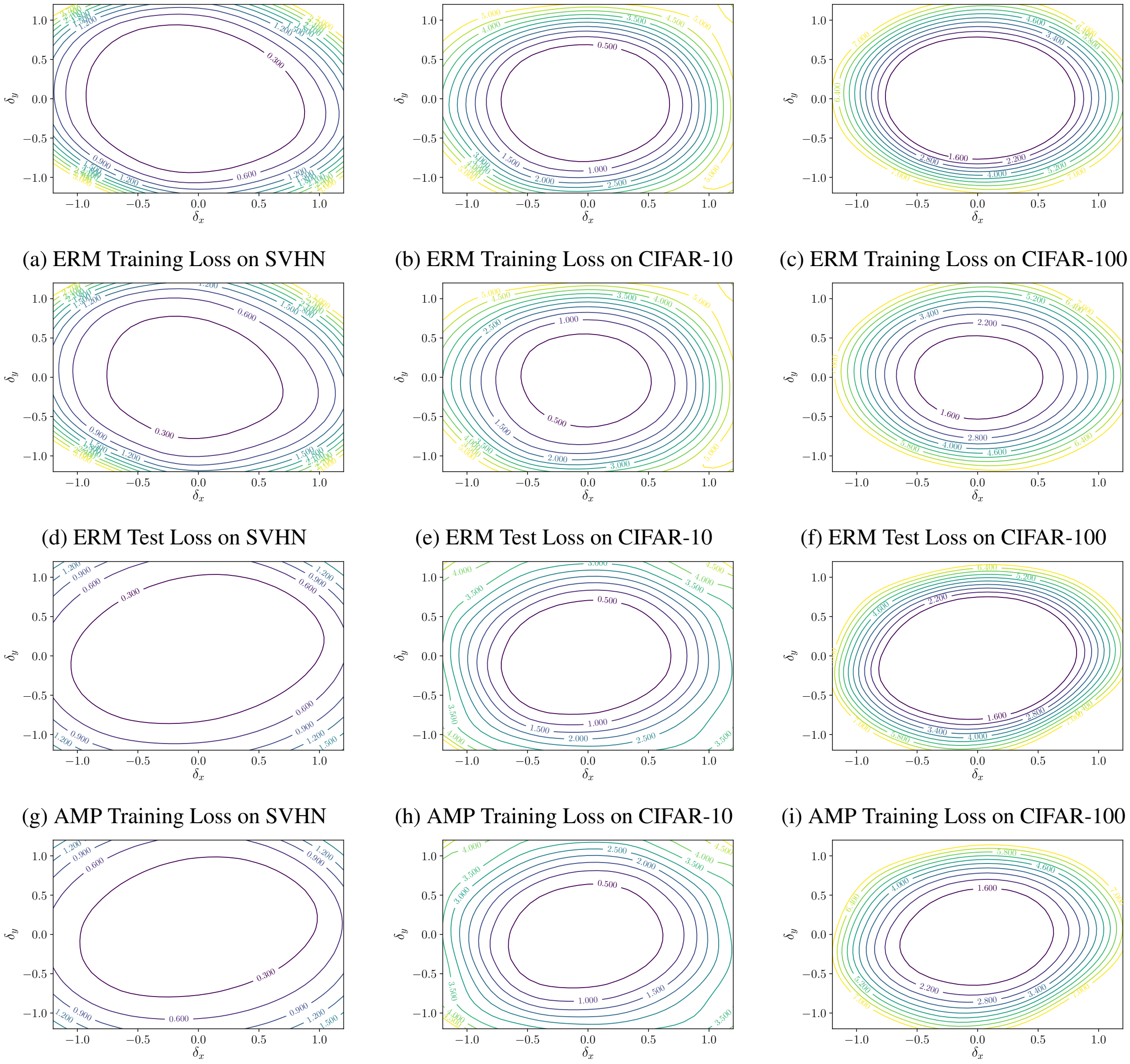
[1] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, ICLR*, 2015.

[2] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML*, pages 1321–1330, 2017.

[3] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems*, pages 6389–6399, 2018.

[4] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR*, 2018.

[5] Pu Zhao, Pin-Yu Chen, Payel Das, Karthikeyan Natesan Ramamurthy, and Xue Lin. Bridging mode connectivity in loss landscapes and adversarial robustness. In *8th International Conference on Learning Representations, ICLR*, 2020.



(a) ERM Training Loss on SVHN

(b) ERM Training Loss on CIFAR-10

(c) ERM Training Loss on CIFAR-100

(d) ERM Test Loss on SVHN

(e) ERM Test Loss on CIFAR-10

(f) ERM Test Loss on CIFAR-100

(g) AMP Training Loss on SVHN

(h) AMP Training Loss on CIFAR-10

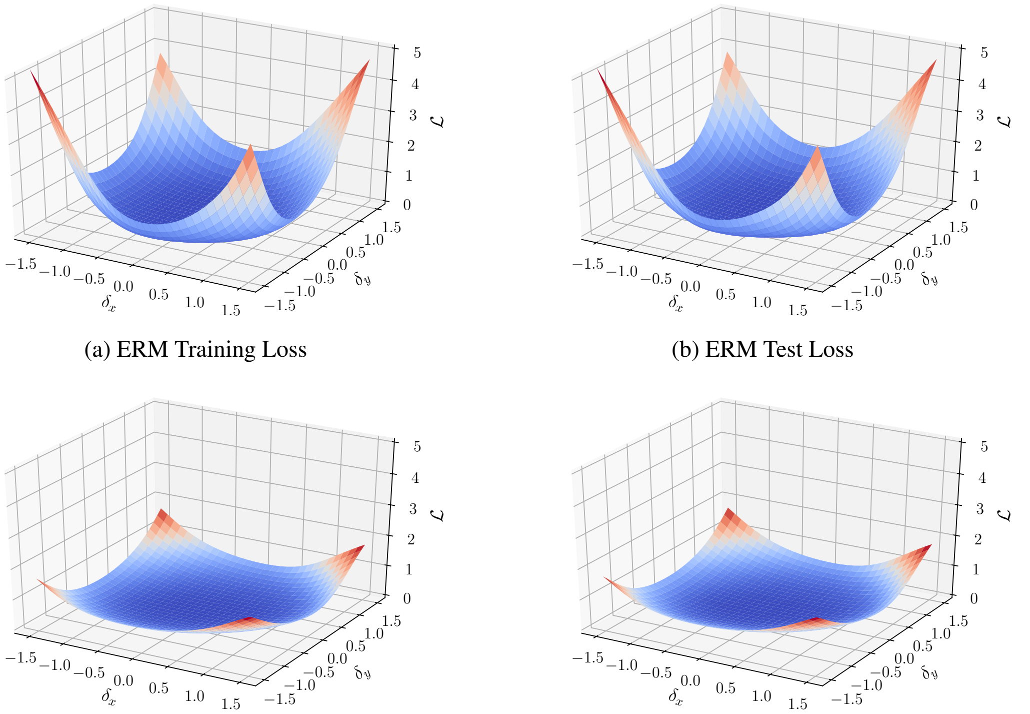
(i) AMP Training Loss on CIFAR-100

(j) AMP Test Loss on SVHN

(k) AMP Test Loss on CIFAR-10

(l) AMP Test Loss on CIFAR-100

Figure 4: 2D visualization of the minima of the empirical risk selected by ERM and AMP on three benchmark image datasets.



(a) ERM Training Loss

(b) ERM Test Loss

(c) AMP Training Loss

(d) AMP Test Loss

Figure 5: 3D visualization of the minima of the empirical risk selected by ERM and AMP on the SVHN dataset.