Improving Adversarial Robustness by Penalizing Natural Accuracy

Kshitij Chandna
New York University
kc4156@nyu.edu

Philip Kutlesa
New York University
pk2264@nyu.edu

Abstract

Current techniques in deep learning are still unable to train adversarially robust classifiers which perform as well as non-robust ones. In this work, we continue to study the space of loss functions, and show that the choice of loss can affect robustness in highly non-intuitive ways. We present a novel loss function and demonstrate that it can improve state-of-the-art adversarial robustness. Our loss function encourages accuracy on adversarial examples, and explicitly penalizes accuracy on natural examples. This is inspired by the theoretical and empirical works suggesting a fundamental trade-off between standard accuracy and adversarial robustness. Our method, NAturally Penalized (NAP) loss, achieves 61.5% robust accuracy on CIFAR-10 with ε = 8/255 perturbations in ℓ_∞ (against a PGD-60 adversary with 20 random restarts). This improves over the standard PGD defense by over 9%, and is within 1% of methods trained on 500K more (unlabeled) examples (Carmon et al., 2019). Our results thus suggest that significant robustness gains are possible by revisiting training techniques, even without additional data.[1]

1. Introduction

Modern deep learning is now mature enough to achieve high test accuracy on many image classification tasks [13, 19–21, 30]. Here, a long line of research has arrived at a certain combination of techniques that work well for image recognition, including architectures (ReLUs, Convolutions, ResNet), optimization algorithms (SGD and variants, with tuned learning-rate schedules), model size, data-augmentation, regularization, normalization, batch-size, and loss function [7, 13, 21, 41].

Many of these choices are not unique and we do not have a complete understanding of why these choices work best in practice. For example, in standard classification our true objective is small $0/1$ test loss, but we often optimize Cross Entropy train loss. We could instead optimize ℓ_2 loss (or any other surrogate loss), but in practice we find optimizing Cross Entropy often performs better. Similarly, large deep networks perform much better than smaller ones in practice, even though these networks have more than enough capacity to"overfit" the train set, and should be performing worse by classical statistical intuition [3, 25, 40]. The optimizer is also poorly understood. In practice we use SGD with learning rates much higher than optimization theory prescribes; and moreover, "accelerate" methods that optimize faster - such as Adam [18] - sometimes generalize worse. Nevertheless, despite our incomplete theoretical understanding, the research community has converged on a methodology which performs very well for standard classification.

In contrast, the field of *adversarially robust* classification has not reached the same level of maturity [4], and has not yet converged on a training methodology that preforms well. The goal of adversarial robustness is to learn classifiers which are robust to small adversarial perturbations [29] of the input. Here, it is not clear if the various design choices that we converged to for standard classification are still the best choices for robust classification.

Current advances in adversarial robustness have come through modifying the training procedure [24], loss function [17, 34, 42], architecture [37, 38], activation function [36], pre-training [15] and leveraging unlabeled data [5]. These are modern research areas, and there are still potentially large robustness gains to be realized from rethinking elements of deep learning methodology.

In this work, we focus on the choice of *loss function*, and show that an unconventional choice of loss can in fact significantly improve adversarial robustness. Concretely, our loss function includes two terms: one which encourages accuracy on adversarial examples, and one which explicitly *penalizes* accuracy on natural examples. This is inspired by the empirical and theoretical observations that there may be a trade-off between standard accuracy and adversarial accuracy for trained models [25, 32]. Intuitively, our loss function penalizes standard accuracy if it is "to good to be true" (i.e., much higher than the adversarial accuracy). This attempts to forcibly trade-off standard accuracy for improved adversarial accuracy, and in practice it yields significant

[1] Code available at: https://github.com/Kshitijc1/NAP

gains over existing methods.

The observation that choice of loss affects adversarial robustness is not novel to our work, and our loss function shares components of existing methods such as TRADES [42] and MART [34]. Many of these methods are motivated as "regularizers", which encourage the network on adversarial inputs to behave similarly to natural inputs. Our method is conceptually and fundamentally different in that it *explicitly penalizes* the classifer's correct behavior on natural inputs. Section 3 provides a detailed examination of existing methods and our approach, and discusses their distinctive characteristics.

Our Contribution Our main contribution is demonstrating that the impact of loss function on robustness is both *large* and *under-explored*. We show that an "unnatural" loss function, which explicitly penalizes natural accuracy, can in fact improve state-of-the-art adversarial robustness: achieving 61.5% robust accuracy on CIFAR-10 with $\varepsilon = 8/255$ perturbations in ℓ_∞, when evaluated against a 60-step PGD attacker with 20 random restarts. We view our work as showing that the space of reasonable loss functions is perhaps larger than expected, and that large robustness gains can still be attained in this space. We also present preliminary insights into what properties of our loss enable it to perform well.

Organization In Section 1.1, we further discuss related works. In Section 2, we present preliminaries and define notation to formally understand the training of adversarially robust classifiers. We discuss our loss function and how it compares to prior work in Section 3. In Section 4, we describe the experimental details and results. In Section 5, we discuss potential intuitions as to why the NAP loss works well compared to standard adversarial training.

1.1. Additional Related Works

Adversarial Robustness Adversarial examples were first considered by (Szegedy et al., 2013) [29]. Subsequently, defenses towards these attacks were proposed and many of these were latter broken [1]. PGD-based adversarial training [24] (related to $FGSM_k$ [12]) remains moderately robust to current attacks. In this work we consider *empirical/heuristic* robustness, as evaluated against strong heuristic attacks (PGD with random restarts). There is also a line of work on provably/certifiably robust neural networks, including randomized-smoothing and certification-based approaches [2, 27, 35, 36]. Note that there is currently a non-trivial gap between provable accuracy and empirical accuracy, and some of these approaches (in particular randomized smoothing) do not apply to ℓ_∞ robustness. There has also been recent work in *randomized smoothing* which refers to taking neural networks that are robust to random perturbations and using them to create classifiers that are adversarially robust [6, 22, 23, 28].

Unlabeled Examples Another approach to improve adversarial accuracy has been to use unlabeled examples for semi-supervised learning. This approach can be used with different loss functions and when used in combination with the TRADES loss and 500K unlabeled examples, it leads to large (around 7%) robustness improvement over TRADES [5, 33].

Robustness In this work we study adversarial robustness. This is distinct from other kinds of robustness which may be desirable, including out-of-distribution robustness, robustness to distribution shift, and robustness to non-adversarial corruptions [10, 11, 14]. Methods which improve adversarial robustness may not yield improvements in other types of robustness. In fact, there is evidence to the contrary [31].

2. Preliminaries and Notation

The goal of adversarial robustness is to learn classifiers that achieve high classification accuracy even under small worst-case perturbations. We consider ℓ_∞ robustness in this work. For a distribution $(x, y) \sim D$ on images $x \in \mathbb{R}^d$ and $y \in [k]$, our objective is to learn a classifier $f : \mathbb{R}^d \to [k]$ with small robust error:

$$RobustError_{D,\varepsilon}(f) = \mathop{\mathbb{E}}_{(x,y)\sim D}[\max_{\delta \in \ell_\infty(\varepsilon)} \mathbb{1}\{f(x+\delta) \neq y\}] \tag{1}$$

where $\ell_\infty(\varepsilon)$ is the ℓ_∞ ball of radius ε. We report robust accuracy as (1 - RobustError).

A crucial detail in adversarial training is the choice of loss function together with the choice of *which parameters we optimize through backpropagation*. This parameter choice can significantly impact robustness. To clarify this, throughout this paper we typeface in bold parameters in the loss function which are backpropogated through (i.e., $\boldsymbol{\theta}$ v.s. θ).

Existing adversarial training methods that modify the loss function, including our method, can all be seen as instances of the generic adversarial training of Algorithm 1 (Fig. 1), for different choices of loss. Formally, we denote loss functions as:

$$L(x, \hat{x}, y; \boldsymbol{\theta})$$

The (x,y) pair represents the natural example, θ is the network parameters, and $\hat{x}$ is an adversarial example for input x. Let $p(x;\theta)$ denote the softmax output of network parameterized by θ, and let $p_y(x;\theta)$ denote the softmax probability on label y. We denote Cross Entropy loss by CE and Kullback-Leibler divergence by KL.

Algorithm 1 Adversarial Training (Simplified)

Input: Neural network f_θ, Loss functions $\mathcal{L}^{\text{upd}}, \mathcal{L}^{\text{atk}}$.
Output: Adversarially-trained network f_θ.

function ADVTRAIN($\mathcal{L}$, f_θ):
  for $t = 1, 2, 3 \ldots, T$ **do**
    Sample example $(x, y) \sim S$
    Construct $\hat{x}$ as an adversarial example for x, by performing PGD on the loss $\mathcal{L}^{\text{atk}}(x, \hat{x}, y; \theta)$ starting from $\hat{x} = x$.
    $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}^{\text{upd}}(x, \hat{x}, y; \boldsymbol{\theta})$
  Output f_θ.

Figure 1. Adversarial training method

There are two losses involved in Algorithm 1: L^{upd} used to update the model parameters, and L^{atk} used to construct adversarial examples for training. In all methods except TRADES, including our method, the attack loss is Cross Entropy, $L^{atk}(x, \hat{x}, y; \theta) := CE(p(\hat{x}; \theta), y)$. For TRADES, the attack loss is $L^{atk}(x, \hat{x}, y; \theta) := KL(p(x; \theta) \| p(\hat{x}; \theta))$.

Standard adversarial training [24] is Algorithm 1 with model-update loss:

$$L^{upd}(x, \hat{x}, y; \boldsymbol{\theta}) := CE(p(\hat{x}; \boldsymbol{\theta}), y) \tag{2}$$

In this work, we introduce a new model update-loss for L^{upd}, and compare with existing losses.

3. Method: NAP Loss

Here we introduce and discuss our loss function NAturally Penalized (NAP) loss, and compare it to other loss functions for adversarial robustness. We start with the motivating observation that the following simple loss, which penalizes natural accuracy, can in fact outperform standard PGD and TRADES:

$$L(x, \hat{x}, y; \boldsymbol{\theta}) := CE(p(\hat{x}; \boldsymbol{\theta}), y) - \lambda CE(p(x, \boldsymbol{\theta}), y) \tag{3}$$

Starting with this loss, we make several modifications to yield further improvements. First, we only penalize natural accuracy on examples where it is higher than adversarial accuracy. Next, we use a "smoothed" version of Cross Entropy - Boosted Cross Entropy (BCE). The above loss is describe as "NAP (no margin, no BCE)".

NAP Loss We propose the following loss:

$$L^{NAP}(x, \hat{x}, y; \boldsymbol{\theta}) := \underbrace{BCE(p(\hat{x}; \boldsymbol{\theta}), y)}_{\text{(A): Adversarial Loss}} - \lambda \underbrace{[p_y(x; \theta) - p_y(\hat{x}; \theta)]_+}_{\text{(B): Overconfidence Margin}} \underbrace{CE(p(x; \boldsymbol{\theta}), y)}_{\text{(C): Natural Loss}} \tag{4}$$

where $\lambda > 0$ and $[\cdot]_+$ denotes ReLU$(\cdot)$. BCE is the Boosted Cross Entropy introduced in MART [34].

The first term (A) encourages adversarial accuracy. The second term (B,C) penalizes natural accuracy on examples where the network is "overconfident" and has higher natural than adversarial accuracy. We do not want to penalize natural accuracy universally, but only in cases when it is "too good to be true" - this is accounted for by the weighting term (B). Notes that term (B) is treated as a constant (i.e., not backpropagated through).

The Boosted Cross Entropy is defined as:

$$\begin{aligned} BCE(p(\hat{x}; \boldsymbol{\theta}), y) &:= CE(p(\hat{x}; \boldsymbol{\theta}), y) - log(1 - \max_{k \neq y} p_k(\hat{x}; \boldsymbol{\theta})) \\ &= -log(p_y(\hat{x}; \boldsymbol{\theta})) - log(1 - \max_{k \neq y} p_k(\hat{x}; \boldsymbol{\theta})) \end{aligned} \tag{5}$$

The BCE is the standard cross entropy, plus a term that encourages the network's softmax output to be "balanced" on incorrect classes.

In Section 4 and Table 3, we do several ablations on our loss function, and we find that:

1. The choice of Boosted Cross Entropy helps robustness but is not crucial: our method still outperforms TRADES with standard CE in place of BCE.

2. The "overconfidence margin" term (B) is not crucial: Even universally penalizing the natural loss yields improvements over TRADES and MART.

This suggest that penalizing natural loss is the crucial component of NAP's performance.

3.1. Comparison to Other Losses

Standard Loss

$$L^{standard}(x, \hat{x}, y; \boldsymbol{\theta}) := CE(p(\hat{x}; \boldsymbol{\theta}), y) \tag{6}$$

This is the Cross Entropy loss on adversarial examples used in standard adversarial training [24].

TRADES Loss

$$\begin{aligned} &L^{TRADES}(x, \hat{x}, y; \boldsymbol{\theta}) := \\ &CE(p(x; \boldsymbol{\theta}), y) + \lambda KL(p(x; \boldsymbol{\theta}) \| p(\hat{x}; \boldsymbol{\theta})) \end{aligned} \tag{7}$$

In TRADES [42], the base term optimizes for *natural accuracy*, while the KL regularizer encourages the natural and adversarial softmax outputs to be close.

In comparison, our method encourages *adversarial accuracy* in the base term, and uses a different regularizer. Our regularizer is in some sense a "softer constraint" than the KL term: we do not peranlize differences in $p(x; \theta)$ and $p(\hat{x}; \theta)$, but only the ones in $p_y(x; \theta)$ and $p_y(\hat{x}; \theta)$ (i.e., those that affect the final classification decision). Another

difference is that when $p_y(x;\theta) > p_y(\hat{x};\theta)$ the regularizer term in TRADES tries to decrease $p_y(x;\theta)$ and to increase $p_y(\hat{x};\theta)$, while the NAP regularizer term only tries to decrease $p_y(x;\theta)$.

MART Loss

$$L^{MART}(x,\hat{x},y;\boldsymbol{\theta}) := BCE(p(\hat{x};\boldsymbol{\theta}),y) + \lambda(1-p_y(x;\boldsymbol{\theta}))KL(p(x;\boldsymbol{\theta})\|p(\hat{x};\boldsymbol{\theta})) \quad (8)$$

In MART [34], the regularizer is weighted by the *natural misclassification probability*, instead of the overconfidence margin. Moreover, the KL regularizer term is the same as TRADES, and is distinct from our naturally-penalized regularizer as discussed previously.

Gradient Comparisons In addition to the aforementioned differences, our NAP loss function has the following properties that separates it from TRADES and MART. For the NAP loss, the partial derivative of the loss with respect to the natural softmax probability is always *non-positive*:

$$\frac{\partial L}{\partial p_y(x;\theta)} \leq 0 \quad (9)$$

That is, the NAP loss always penalizes natural accuracy, all else being equal. Also, the partial derivative with respect to the adversarial softmax probability is always *positive*:

$$\frac{\partial L}{\partial p_y(\hat{x};\theta)} > 0 \quad (10)$$

That is, the NAP loss always encourages adversarial accuracy, all else being equal. Note that accounting for the total derivative, the natural accuracy can increase after a gradient step, even though the partial derivative is negative.

4. Experiments

4.1. Experimental Details

Dataset Our results are reported on CIFAR-10 [20]. This dataset consists of 60K 32x32 images (50K train and 10K test). We normalize all images to be in the range $[0,1]$. We use the standard data augmentation: random horizontal flip and 4 pixel padding with 32 x 32 random crop.

Adversarial Examples The ℓ_∞ budget for adversarial examples is $\varepsilon = 8/255$. For training, adversarial examples are created with random-start and PGD-10 with step-size of $2/255$. While doing PGD to create adversarial examples, we also use the batch-mean/variance for batch normalization layers rather than the moving mean/variance. In our experiments with the loss functions this change led to small but consistent improvement in test accuracy. For testing with PGD-20 we use step-size of $0.8/255$. For evaluation with stronger attacks, we follow (Carmon et al., 2019) [5] for PGD-60: 60 PGD steps with step size $\eta = 0.01$, using 20 random restarts per example, and searching for adversarial attack success at each PGD step.

Models We use Wide ResNet 34-4 [39] to compare different loss functions and Wide ResNet 34-10 to establish the state-of-the-art model to compare with (Zhang et al., 2019; Wang et al., 2020) [34, 42]. All models were trained for 80 epochs with an initial learning rate of 0.1, a step decay of 0.1 on epoch 40 and epoch 60, batch size of 128, weight decay of 5 x 10^{-4} and SGD with momentum 0.9. All reported accuracy results are with optimal early stopping with respect to adversarial accuracy for PGD-20.

Hyperparameter Tuning For all methods involving a hyperparameter λ (TRADES, MART, and NAP), we individually hyperparameter search to set λ. The values obtained were $\lambda = 8, 8, 6$ for TRADES, MART and NAP, respectively. Reported results used these hyperparameter values unless explicitly stated otherwise.

Ablation We preform ablation on the various components and combinations of components of NAP loss to determine the significance of each.

Evaluation Metrics Natural accuracy is the percentage of correct classifications the model makes on natural examples. We report robust accuracy as (1 - RobustError) Equation (1).

4.2. Results

First, we compare our NAP loss against existing losses on a Wide ResNet 34-4. As depicted in Table 1, the NAP loss outperforms standard PGD, TRADES, and MART on robust accuracy. NAP performs 3.6 percentage points better than the second highest scoring MART against the PGD-20 adversary, and 3 percentage points better against PGD-60. The NAP loss demonstrates significant improvements for adversarial robustness compared to Standard loss and remains competitive on natural accuracy.

Loss	Natural	PGD-20	PGD-60
Standard	**82.2**	53.6	52.1
TRADES	81.8	55.3	53.9
MART	79.9	59.0	57.5
NAP	82.1	**62.6**	**61.5**

Table 1. Natural and adversarial accuracy for adversarially-trained Wide ResNet 34-4 on CIFAR-10 with $\varepsilon = 8/255$ perturbations in ℓ_∞.

To compare against current state-of-the-art models in the literature, we train the NAP loss for a Wide ResNet 34-10. This model trained with NAP loss improves on the state-of-the-art of 58.6% against the PG-20 attacked reported by MART. Moreover, it achieves 61.5% robust accuracy against the strong PGD-60 adversary. This is within 1% of the robust accuracy achieved in (Carmon et al., 2019) [5] for a model trained with 500K additional unlabeled examples. Our results thus suggest that significant robustness gains are possible by modifying elements of the training procedure, even without additional data.

Loss	Natural	PGD-20	PGD-60
Standard	**84.5**	55.5	54.0
NAP	82.1	**63.2**	**61.5**

Table 2. Natural and adversarial accuracy for adversarially-trained Wide ResNet 34-10 on CIFAR-10 with $\varepsilon = 8/255$ perturbations in ℓ_∞.

We continue to evaluate NAP Loss against novel adversaries that challenge Standard Loss and TRADES Loss [8]. These results are presented in Table 3, which depicts that NAP Loss continues to generalize well across several attacks. Notably, NAP continues to exhibit superior performance against PGD-based adversaries.

Loss	APGD_{CE}	APGD^T_{DLR}	FAB^T	Square
Standard	44.75	44.28	44.75	53.10
TRADES	55.28	**53.10**	**53.45**	**59.43**
NAP	**58.42**	50.41	50.82	57.08

Table 3. Adversarial accuracy for adversarially-trained Wide ResNet 34-10 on CIFAR-10 with $\varepsilon = 8/255$ perturbations in ℓ_∞.

4.3. Ablations

Here we perform several ablations to study the relative impact of components in the NAP loss. The Wide ResNet 34-4 model is used for all experiments presented in this section. For all modified losses, we conduct individual hyperparameter search over λ. These hyperparameter values are stated in each respective subsection.

Loss	Natural	PGD-20
NAP (no margin, no BCE)	79.1	57.3
NAP (no margin)	77.3	59.5
NAP (no BCE)	**79.5**	58.1
NAP	78.9	**62.6**

Table 4. Ablations on our NAP loss function.

NAP Loss (no BCE) We replace the Boosted Cross Entropy (BCE) term in NAP with standard Cross Entropy, and find that is still outperforms TRADES but not MART Table 4. The modified loss function is:

$$L^{NAP(no\ BCE)}(x, \hat{x}, y; \boldsymbol{\theta}) := \\ CE(p(\hat{x}; \boldsymbol{\theta}), y) - \lambda[p_y(x; \theta) - p_y(\hat{x}; \theta)]_+ CE(p(x; \boldsymbol{\theta}), y) \tag{11}$$

For the result in Table 4 λ is set to be 6.

NAP Loss (no margin) We replace the "overconfidence margin" term in NAP with a constant - that is, we always penalize the natural examples. We find that it still outperforms TRADES and Mart Table 4. The modified loss function is:

$$L^{NAP(no\ margin)}(x, \hat{x}, y; \boldsymbol{\theta}) := \\ BCE(p(\hat{x}; \boldsymbol{\theta}), y) - \lambda CE(p(x; \boldsymbol{\theta}), y) \tag{12}$$

For the result in Table 4 λ is set to be 0.3.

NAP Loss (no margin, no BCE) Here we combine both the above changes (i.e., replace the Boosted Cross Entropy term in NAP with standard Cross Entropy and replace the "overconfidence margin" term in NAP with a constant). We find that it still outperforms TRADES but not MART Table 4. The modified loss function is:

$$L^{NAP(no\ BCE,\ no\ margin)}(x, \hat{x}, y; \boldsymbol{\theta}) := \\ CE(p(\hat{x}; \boldsymbol{\theta}), y) - \lambda CE(p(x; \boldsymbol{\theta}), y) \tag{13}$$

For the result in Table 4 λ is set to 0.3.

4.4. Boosted Cross-Entropy

From Table 4 we know that replacing BCE with CE has a significant (4.5%) negative effect. Given this, we now consider whether BCE loss performs better than CE loss in other settings as well. We first compare the BCE loss and CE loss for adversarial training. That is, we use the following model update loss:

$$L^{BCE}(x, \hat{x}, y; \boldsymbol{\theta}) := BCE(p(\hat{x}; \boldsymbol{\theta}), y) \tag{14}$$

the results for adversarial training with this loss are in Table 5.

Loss	Natural	PGD-20
Standard	**82.2**	53.55
BCE	80.4	**54.7**

Table 5. Natural and adversarial accuracy for adversarially-trained Wide ResNet 34-4 on CIFAR-10 with $\varepsilon = 8/255$ perturbations in ℓ_∞. Average of 2 runs.

Next we compare the BCE loss and CE loss for natural training, and list results in Table 6. Here we train on only

the first 10K images of CIFAR-10, to approximately match the standard accuracy with that of robust models. In both cases, BCE helps but the difference is around 1% while the difference when replacing BCE with CE in NAP loss was 4.5%. Thus, BCE appears to be much more important when combined with the regularizer term. A similar observation was also made for the MART loss [34].

Loss	Natural Accuracy
CE	72.25
BCE	**73.1**

Table 6. Average natural accuracy of 4 runs for naturally-trained ResNet 34 on first 10K images from CIFAR-10.

5. Discussion

In this section we discuss potential intuitions and preliminary experiments towards understanding why our NAP loss function performs well. We stress that these are informal intuitions, and we do not have a formal understanding of the impact of loss functions on robustness.

Intuitions for Penalizing Natural Accuracy Our NAP loss is inspired by the empirical and theoretical observations that there may be a trade-off between standard accuracy and adversarial accuracy for current training procedures/models [25, 32]. In fact, our experiments (Table 1) have the property that the trends for adversarial and natural accuracy are reversed. If such a trade-off is indeed intrinsic among trained models, then it may be reasonable to forcibly sacrifice standard accuracy with the aim of improving adversarial accuracy, as the NAP loss does.

Another potential intuition about our loss function is in the "robust features" framework [9,16]. The intuition is: we want to prevent our model from learning "fragile" features about the distribution, which are very helpful for standard classification, but are not robust. Relying on such features is an easy way to improve standard accuracy, but may be harmful towards the goal of adversarial accuracy. If our model is more accurate on natural examples than adversarial ones, this suggests it could be learning these "fragile/non-robust" features. Our loss regularizes against this by penalizing natural accuracy if it is higher than adversarial accuracy.

These two intuitions are not necessarily separate. One possible explanation of the trade-off between standard accuracy and natural accuracy is that current techniques to improve adversarial accuracy attempt to force the model to only rely on robust features [16]. These features are harder to learn than non-robust features, and hence the natural accuracy suffers.

Softmax Distribution of Natural vs. Adversarial Examples To study the magnitude and influence of the "margin term" in the NAP loss, we plot the empirical distributions on the test set of $p_y(x;\theta)$,$p_y(\hat{x};\theta)$ and the "margin term" $p_y(x;\theta) - p_y(\hat{x};\theta)$ at the end of training for the models in Table 1. The corresponding histograms are present in Figure 2 and Figure 3. Interestingly, the adversarial confidence $p_y(\hat{x};\theta)$ is bimodal (with modes around 0 and 1) for standard adversarial training but not for the NAP loss. This suggests the NAP loss is indeed encouraging the network to behave similarly on natural and adversarial inputs, though the exact mechanisms for why this yields robustness improvements is unclear, and requires further study. The train data also has a similar but milder bimodal distribution for standard adversarial training.

6. Conclusion

We introduce a novel loss function, the NAturally Penalized (NAP) loss, which yields significant improvements in state-of-the-art adversarial robustness for CIFAR-10. We view our work as showing that the space of reasonable loss functions for adversarial robustness is perhaps larger than expected, and significant robustness gains are still possible in this space.

References

[1] Athalye, A., Carlini, N., and Wagner, D. A. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholm, Sweden, July 10-15, 2018, pp. 274–283, 2018.* 2

[2] Balunovic, M. and Vechev, M. Adversarial training and provable defenses: Bridging the gap. In *International Conference on Learning Representations, 2020. https://openreview.net/forum?id=SJxSDxrKDr.* 2

[3] Belkin, M., Hsu, D., Ma, S., and Mandal, S. Reconciling modern machine-learning practice and the classical bias– variance trade-off. In *Proceedings of the National Academy of Sciences, 116(32):15849–15854, 2019.* 1

[4] Carlini, N., Athalye, A., Papernot, N., Brendel, W., Rauber, J., Tsipras, D., Goodfellow, I., and Madry, A. On evaluating adversarial robustness. 2019. 1

[5] Carmon, Y., Raghunathan, A., Schmidt, L., Duchi, J. C., and Liang, P.S. Unlabeled data improves adversarial robustness. In *Advances in Neural Information Processing Systems, pp. 11190–11201, 2019.* 1, 2, 4, 5

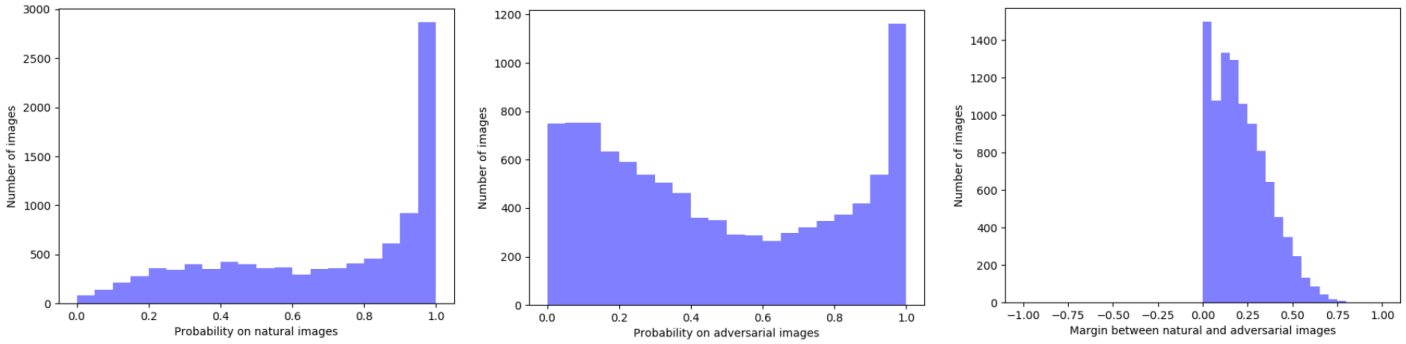

Figure 2. **Standard Loss.** Histogram of softmax probability on correct labels for the Test set at the end of standard adversarial training. Using Wide ResNet 34-10 same model as Table 2. Plotting histograms of (left to right): $p_y(x;\theta)$,$p_y(\hat{x};\theta)$, and $p_y(x;\theta) - p_y(\hat{x};\theta)$.

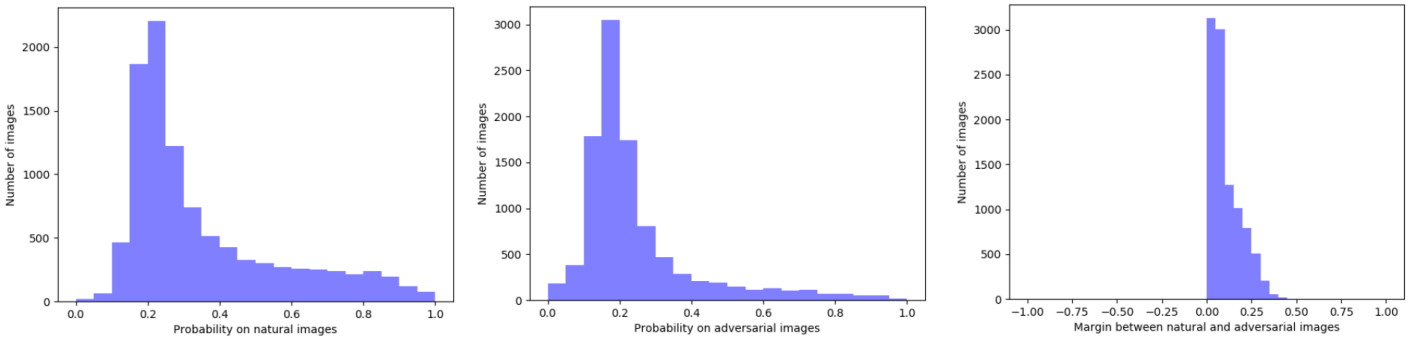

Figure 3. **NAP Loss.** Histogram of softmax probability on correct labels for the Test set at the end of training with the NAP loss. Using Wide ResNet 34-10 same model as Table 2. Plotting histograms of (left to right): $p_y(x;\theta)$,$p_y(\hat{x};\theta)$, and $p_y(x;\theta) - p_y(\hat{x};\theta)$.

[6] Cohen, J. M., Rosenfeld, E., and Kolter, J. Z. Certified adversarial robustness via randomized smoothing. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach California, USA, pp. 1310–1320, 2019.* 2

[7] Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. Autoaugment: Learning augmentation strategies from data. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, pp. 113–123, 2019.* 1

[8] Croce, F. and Hein, M., 2020, November. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning (pp. 2206-2216). PMLR.* 5

[9] Engstrom, L., Gilmer, J., Goh, G., Hendrycks, D., Ilyas, A., Madry, A., Nakano, R., Nakkiran, P., Santurkar, S., Tran, B., Tsipras, D., and Wallace, E. A discussion of 'adversarial examples are not bugs, they are features'. Distill, 2019. doi: 10.23915/distill.00019. https://distill.pub/2019/advex-bugs-discussion 6

[10] Fawzi, A. and Frossard, P. Manitest: Are classifiers really invariant? In *Proceedings of the British Machine Vision Conference 2015, BMVC 2015, Swansea, UK, September 7-10, 2015, pp. 106.1–106.13, 2015.* 2

[11] Geirhos, R., Temme, C. R. M., Rauber, J., Schutt, H. H., Bethge, M., and Wichmann, F. A. Generalisation in humans and deep neural networks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montreal, Canada, pp. 7549–7561, 2018.* 2

[12] Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representa-*

tions, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015. 2

[13] He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, pp. 770–778, 2016.* 1

[14] Hendrycks, D. and Dietterich, T. G. Benchmarking neural network robustness to common corruptions and perturbations. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019, 2019.* 2

[15] Hendrycks, D., Lee, K., and Mazeika, M. Using pre-training can improve model robustness and uncertainty. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA, pp. 2712–2721, 2019.* 1

[16] Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., and Madry, A. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems, pp. 125–136, 2019.* 6

[17] Kannan, H., Kurakin, A., and Goodfellow, I. J. Adversarial logit pairing. CoRR, abs/1803.06373, 2018. http: //arxiv.org/abs/1803.06373. 1

[18] Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014. 1

[19] Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. Commun. ACM, 60(6):84–90, 2017. doi: 10.1145/ 3065386. http://doi.acm.org/10.1145/ 3065386. 1

[20] Krizhevsky, A. et al. Learning multiple layers of features from tiny images. 2009. 1, 4

[21] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998. 1

[22] Lecuyer, M., Atlidakis, V., Geambasu, R., Hsu, D., and Jana, S. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy, SP 2019, San Francisco, CA, USA, May 19-23, 2019, pp. 656–672, 2019.* 2

[23] Li, B., Chen, C., Wang, W., and Carin, L. Second-order adversarial attack and certifiable robustness. CoRR, abs/1809.03113, 2018. URL http://arxiv.org/ abs/1809.03113. 2

[24] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083, 2017. 1, 2, 3

[25] Nakkiran, P. Adversarial robustness may be at odds with simplicity. arXiv preprint arXiv:1901.00532, 2019. 1, 6

[26] Nakkiran, P., Kaplun, G., Bansal, Y., Yang, T., Barak, B., and Sutskever, I. Deep double descent: Where bigger models and more data hurt. In *International Conference on Learning Representations, 2020. URL https:// openreview.net/forum?id=B1g5sA4twr.*

[27] Raghunathan, A., Steinhardt, J., and Liang, P. Certified defenses against adversarial examples. In 6th *International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings, 2018.* 2

[28] Salman, H., Li, J., Razenshteyn, I. P., Zhang, P., Zhang, H., Bubeck, S., and Yang, G. Provably robust deep learning via adversarially trained smoothed classifiers. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada, pp. 11289–11300, 2019.* 2

[29] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199, 2013. 1, 2

[30] Tan, M. and Le, Q. V. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA, pp. 6105–6114, 2019.* 1

[31] Taori, R., Dave, A., Shankar, V., Carlini, N., Recht, B., and Schmidt, L. When robustness doesn't promote robustness: Synthetic vs. natural distribution shifts on imagenet, 2020. URL https://openreview.net/forum? id=HyxPIyrFvH. 2

[32] Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., and Madry, A. Robustness may be at odds with accuracy. arXiv preprint arXiv:1805.12152, 2018. 1, 6

[33] Uesato, J., Alayrac, J., Huang, P., Stanforth, R., Fawzi, A., Kohli, P., et al. Are labels required for improving adversarial robustness? In Advances in Neural Information Processing Systems, pp. 12192–12202, 2019. 2

[34] Wang, Y., Zou, D., Yi, J., Bailey, J., Ma, X., and Gu, Q. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations, 2020. URL https://openreview.net/forum?id=rklOg6EFwS*. 1, 2, 3, 4, 6

[35] Wong, E., Schmidt, F. R., Metzen, J. H., and Kolter, J. Z. Scaling provable adversarial defenses. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montreal, Canada, pp. 8410–8419, 2018*. 2

[36] Xiao, C., Zhong, P., and Zheng, C. Enhancing adversarial defense by k-winners-take-all. In *International Conference on Learning Representations, 2020. URL https://openreview.net/forum?id=Skgvy64tvr*. 1, 2

[37] Xie, C. and Yuille, A. Intriguing properties of adversarial training at scale. In *International Conference on Learning Representations, 2020. URL https://openreview. net/forum?id=HyxJhCEFDS*. 1

[38] Xie, C., Wu, Y., Maaten, L. v. d., Yuille, A. L., and He, K. Feature denoising for improving adversarial robustness. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 501–509, 2019*. 1

[39] Zagoruyko, S. and Komodakis, N. Wide residual networks. In *Proceedings of the British Machine Vision Conference 2016, BMVC 2016, York, UK, September 19-22, 2016, 2016*.

[40] Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. arXiv preprint arXiv:1611.03530, 2016. 4

[41] Zhang, H., Cisse , M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond empirical risk minimization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings, 2018*. 1

[42] Zhang, H., Yu, Y., Jiao, J., Xing, E. P., Ghaoui, L. E., and Jordan, M. I. Theoretically principled trade-off between robustness and accuracy. arXiv preprint arXiv:1901.08573, 2019. 1

1, 2, 3, 4

7. Appendix

Here we share additional figures that illustrate Standard, NAP, Trades and MART losses training and testing performances for various hyperparameter values.

Figure 4. NAP loss (green) achieves best test error with λ set to 6

Figure 5. NAP loss (green) with λ set to 6 test error alternative

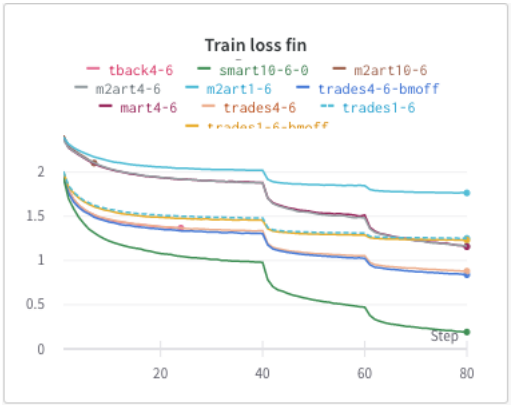

Figure 6. NAP loss (green) with λ set to 6. Training loss for various loss functions explored.