LOGIT REGULARIZATION METHODS FOR ADVERSARIAL ROBUSTNESS

Anonymous authors
Paper under double-blind review

ABSTRACT

While great progress has been made at making neural networks effective across a wide range of tasks, many are surprisingly vulnerable to small, carefully chosen perturbations of their input, known as adversarial examples. In this paper, we advocate for and experimentally investigate the use of logit regularization techniques as an adversarial defense, which can be used in conjunction with other methods for creating adversarial robustness at little to no cost. We demonstrate that much of the effectiveness of one recent adversarial defense mechanism can be attributed to logit regularization and show how to improve its defense against both white-box and black-box attacks, in the process creating a stronger black-box attacks against PGD-based models.

1 INTRODUCTION

Neural networks, despite their high performance on a variety of tasks, can be brittle. Given data intentionally chosen to trick them, many deep learning models suffer extremely low performance. This type of data, commonly referred to as *adversarial examples*, represent a security threat to any machine learning system where an attacker has the ability to choose data input to a model, potentially allowing the attacker to control a model's behavior.

Today, adversarial examples are typically created by small, but carefully chosen transformations of data that models are otherwise high-performant on. This is primarily due to the ease of experimentation with existing datasets (Gilmer et al., 2018), though the full threat of adversarial examples is only limited by the ability and creativity of an attacker's example generation process.

Even with the limited threat models considered in current research, performance on adversarially chosen examples can be dramatically worse than unperturbed data – for example, white-box accuracy on adversarially chosen examples for the CIFAR-10 image classification task (Krizhevsky & Hinton, 2009) is lower than 50%, even for the most robust defenses known today (Madry et al., 2018; Kannan et al., 2018), while unperturbed accuracy can be as high as 98.5% Cubuk et al. (2018).

Current defenses against adversarial examples generally come in one of a few flavors. Perhaps the most common approach is to generate adversarial examples as part of the training procedure and explicitly train on them ("adversarial training"). Another approach is to transform the model's input representation in a way that thwarts an attacker's adversarial example construction mechanism. While these methods can be effective, care must be taken to make sure that they are not merely obfuscating gradients (Athalye et al., 2018). Last, generative models can be built to model the original data distribution, recognizing when the input data is out of sample and potentially correcting it (Song et al., 2018; Samangouei et al., 2018). Of these, perhaps the most robust today is adversarial logit pairing (Kannan et al., 2018), which extends the adversarial training work of Madry et al. (2018) by incorporating an additional term to make the logits (pre-softmax values) of an unperturbed and adversarial example more similar.

In this work, we show that adversarial logit pairing derives a large fraction of its benefits from regularizing the model's logits toward zero, which we demonstrate through simple and easy to understand theoretical arguments in addition to empirical demonstration. Investigating this phenomenon further, we examine two alternatives for logit regularization, finding that both result in improved robustness to adversarial examples, sometimes surprisingly so – for example, using the right amount of label smoothing (Szegedy et al., 2016) can result in greater than 40% robustness to a projected gradient descent (PGD) attack (Madry et al., 2018) on CIFAR-10 while training only on the original, unperturbed training examples, and is also a compelling black-box defense. We then present an alternative formulation of adversarial logit pairing that separates the logit pairing and logit regularization effects, improving the defense. The end result of these investigations is a defense that sets a new state-of-the-art for PGD-based adversaries on CIFAR-10 for both white box and black box attacks, while requiring little to no computational overhead on top of adversarial training.

2 Overview of Adversarial Training

Before proceeding with our analysis, it is prudent to review existing work on adversarial training for context. While adversarial examples have been examined in the machine learning community in some capacity for many years (Dalvi et al., 2004), their study has drawn a sharp focus in the current renaissance of deep learning, starting with Szegedy et al. (2014) and Goodfellow et al. (2015). In Goodfellow et al. (2015), adversarial training is presented as training with a weighted loss between an original and adversarial example, *i.e.* with a loss of

$$\tilde{J}(\theta, x, y) = \frac{1}{m}\sum_{i=1}^{m} \alpha J(\theta, x^{(i)}, y^{(i)}) + (1-\alpha) J(\theta, g(x^{(i)}), y^{(i)}) \tag{1}$$

where $g(x)$ is a function representing the adversarial example generation process, originally presented as $g(x) = x + \epsilon \cdot \text{sign}(\nabla_x J(\theta, x, y))$, α is a weighting term between the original and adversarial examples typically set to 0.5, and as usual θ are the model parameters to learn, J is a cross-entropy loss, m is the dataset size, $x^{(i)}$ is the ith input example, and $y^{(i)}$ is its label. Due to the use of a single signed gradient with respect to the input example, this method was termed the "fast gradient sign method" (FGSM), requiring a single additional forward and backward pass of the network to create. Kurakin et al. (2016) extended FGSM into a multi-step attack, iteratively adjusting the perturbation applied to the input example through several rounds of FGSM. This was also the first attack that could be described as a variant of projected gradient descent (PGD). Both of these approaches primarily target an L^∞ threat model, where the L^∞ norm between the original and adversarial example is constrained to a small value.

Madry et al. (2018) built upon these works by initializing the search process for the adversarial perturbation randomly, and is the strongest attack currently available to the best of our knowledge. Through extensive experiments, they showed that even performing PGD with a single random initialization is able to approximate the strongest adversary found with current first-order methods. However, as with multi-step FGSM, performing adversarial training with this approach can be rather expensive, taking an order of magnitude longer than standard training due to requiring $N+1$ forward and backward passes of the model, where N is the number of PGD iterations.

Improving on PGD-based adversarial training, Kannan et al. (2018) introduced adversarial logit pairing (ALP), which adds a term to the adversarial training loss function that encourages the model to have similar logits for original and adversarial examples:

$$\tilde{J}(\theta, x, y) = \frac{1}{m}\sum_{i=1}^{m} \alpha J(\theta, x^{(i)}, y^{(i)}) + (1-\alpha) J(\theta, g(x^{(i)}), y^{(i)}) + \lambda L(f(x^{(i)}; \theta), f(g(x^{(i)}); \theta)). \tag{2}$$

where L was set to an L^2 loss in experiments and $f(x, \theta)$ returns the logits of the model corresponding to example x. Adversarial logit pairing has the motivation of increasing the amount of structure in the learning process, by encouraging the model to have similar prediction patterns on the original and adversarial examples, a process reminiscent of distillation (Hinton et al., 2015).

Kannan et al. (2018) also studied a baseline version of ALP, called "clean logit pairing", which paired randomly chosen unperturbed examples together. Surprisingly, this worked reasonably well, inspiring them to experiment with a similar idea they call "clean logit squeezing", regularizing the L^2 norm of the model's logits, which worked even more effectively, though this idea itself was not combined with adversarial training. It is this aspect of the work that is most related to what we study in this paper.

3 Adversarial Logit Pairing and Logit Regularization

We now show how adversarial logit pairing (Kannan et al., 2018) acts as a logit regularizer. For notational convenience, denote $\ell_c^{(i)}$ as the logit of the model for class c on example i in its original, unperturbed form, and $\tilde{\ell}_c^{(i)}$ as the logit for the corresponding adversarial example. The logit pairing term in adversarial logit pairing is a simple L^2 loss:

$$L = \frac{1}{2}(\ell_c^{(i)} - \tilde{\ell}_c^{(i)})^2 \tag{3}$$

While it is obvious that minimizing this term will have the effect of making the original and adversarial logits more similar in some capacity, what precise effect does it have on the model during training? To examine this, we look at the gradient of this loss term with respect to the logits themselves:

$$\frac{\partial L}{\partial \ell_c^{(i)}} = \ell_c^{(i)} - \tilde{\ell}_c^{(i)} \tag{4}$$

$$\frac{\partial L}{\partial \tilde{\ell}_c^{(i)}} = \tilde{\ell}_c^{(i)} - \ell_c^{(i)} \tag{5}$$

Under the assumption that the adversarial example moves the model's predictions away from the correct label (as should be the case with any reasonable adversarial example, such as an untargeted PGD-based attack), we have that $\ell_c^{(i)} > \tilde{\ell}_c^{(i)}$ when $c = y^{(i)}$ is the correct category, and $\ell_c^{(i)} < \tilde{\ell}_c^{(i)}$ otherwise. Keeping in mind that model updates move in the direction opposite of the gradient, then the update to the model's weights will attempt to make the original logits smaller and the adversarial logits larger when $c = y^{(i)}$ and will otherwise attempt to make the original logits larger and the adversarial logits smaller.

However, this must be considered in the context of the adversarial training loss $\tilde{J}$ – in particular, the standard cross-entropy loss used in $\tilde{J}$ for the adversarial example $g(x^{(i)})$ already encourages the adversarial logits to be higher for the correct category and smaller for all incorrect categories, and furthermore the scale of the loss $\tilde{J}$ typically is an order of magnitude larger than the adversarial pairing loss. Thus, we argue that the main effect of adversarial logit pairing is actually in the remaining two types of updates, encouraging the logits of the original example to be smaller for the correct category and larger for all incorrect categories. These two updates have very similar effects to simply regularizing model logits, *e.g.* in a manner similar to "logit squeezing" Kannan et al. (2018) or label smoothing Szegedy et al. (2016).

We can also view this from a different perspective by explicitly incorporating the *scale* of the logits in the logit pairing term. If we factor out a shared scale factor γ from each logit, the logit pairing term becomes

$$L = \frac{1}{2}(\gamma\ell_c^{(i)} - \gamma\tilde{\ell}_c^{(i)})^2 \tag{6}$$

implying that

$$\frac{\partial L}{\partial \gamma} = \gamma(\ell_c^{(i)} - \tilde{\ell}_c^{(i)})^2, \tag{7}$$

meaning that the model would always attempt to update the scale of the logits in the opposite direction of the sign of γ, *i.e.* toward zero, so long as the logits were not identical. In practice, this affect is counterbalanced by the adversarial training term, which requires non-identical logits to minimize its cross-entropy loss.

Given this interpretation, in this work we now explore 1) whether this can be verified experimentally, 2) if other forms of logit regularization have similar effects, 3) if decoupling adversarial logit pairing

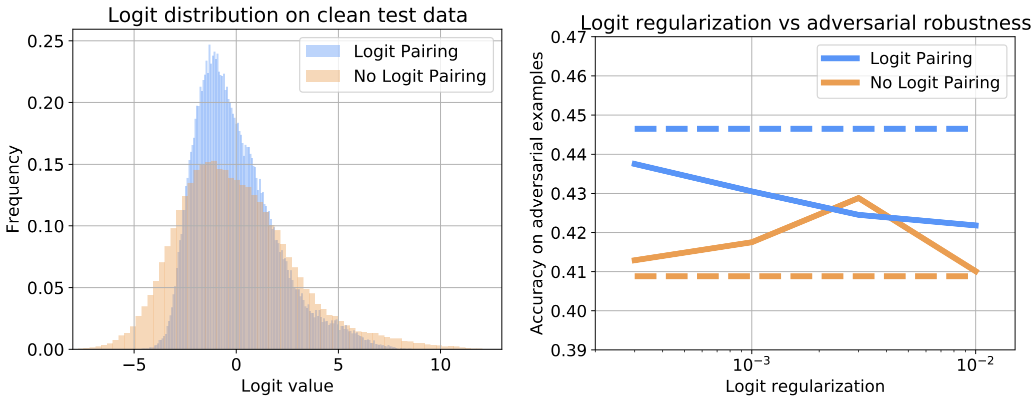

Figure 1: Left: Distribution of logits on clean test data for models trained with and without logit pairing. Right: Performance against a PGD-based attack for models trained with varying amounts of logit regularization, with and without logit pairing.

explicitly into a form where the effect of logit regularization and pairing can be disentangled, and 4) whether the above analysis can yield insights to making more adversarially robust methods.

3.1 Experimental Evidence

Perhaps the most straightforward way to test our hypothesis is to examine the logits of a model trained with ALP vs one trained with standard adversarial training. If true, the model trained with ALP will have logits that are generally smaller in magnitude. We do this in Figure 1(left), using a ResNet (He et al., 2016) classifier on CIFAR-10 (Krizhevsky & Hinton, 2009) as our testbed.

We see that it is indeed the case that the logits for a model trained with ALP are of smaller magnitude than those of a model trained with PGD, with a variance reduction of the logits from 8.31 to 4.02 on clean test data (distributions on a set of PGD adversarial examples are very similar). This provides evidence that ALP *does* have the effect of regularizing logits, though this data alone is not sufficient to determine if this is a key mechanism in improved performance.

To answer this, we can examine if standard adversarial training can be improved by explicitly regularizing the logits. If adversarial robustness can be improved, but similar improvements can *not* be made to ALP, then at least some of the benefits of ALP can be attributed to logit regularization. We present the results of this experiments in Figure 1(right), implemented using the “logit squeezing” form of regularization (L^2-regularization on the logits).

These results show that incorporating regularization on model logits is able to recover slightly more than half of the improvement from logit pairing, with too little regularization having only a small effect, and too much regularization approaching the point of being harmful. However, when added to a model already trained with ALP, regularizing the logits does not lead to any improvement, and in fact hurts performance, likely due to putting too much strength into logit regularization. This evidence makes clear that one of the key improvements from logit pairing is due to a logit regularization effect.

We would like to emphasize that these results are not meant to diminish ALP in any sense – adversarial logit pairing *works*, and our goals are to investigate the mechanism by which it works and explore if it can be generalized or improved. Given these results, it is worth examining other methods that have an effect of regularizing logits in order to tell whether this is a more general phenomenon.

4 Other forms of logit regularization

Label Smoothing. Label smoothing is the process of replacing the one-hot training distribution of labels with a softer distribution, where the probability of the correct class has been smoothed out onto the incorrect classes (Szegedy et al., 2016). Concretely, label smoothing uses the target

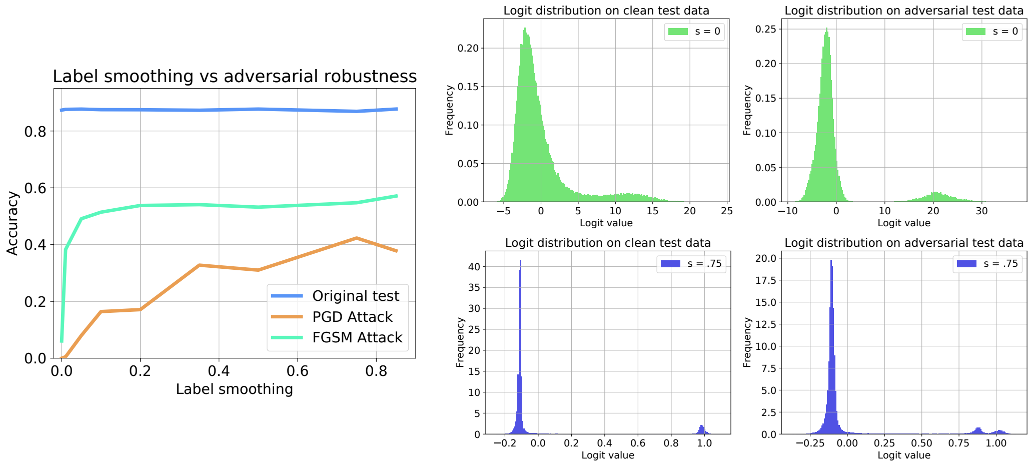

Figure 2: Left: Clean and adversarial accuracy on CIFAR-10 as a function of the label smoothing strength. Note that models for this figure were trained exclusively on the original training data – no adversarial examples were involved in the training procedure. Right: Logit distribution of model trained with no label smoothing (top) and with label smoothing of $s = .75$ (bottom), evaluated on the original test images (left) and PGD-based adversarial examples (right).

distribution:

$$p_c^{(i)} = \begin{cases} 1 - s & c = y^{(i)} \\ \frac{s}{C-1} & c \neq y^{(i)} \end{cases} \tag{8}$$

where $p_c^{(i)}$ is the target probability for class c for example i, C is the number of categories and $s \in [0, 1 - \frac{1}{C}]$ is the smoothing strength. Label smoothing was originally introduced as a form of regularization, designed to prevent models from being too confident about training examples, with the goal of improved generalization. It can be easily implemented as a preprocessing step on the labels, and does not affect model training time in any significant way. Interestingly, Kurakin et al. (2016) found that disabling the small amount of label smoothing present in a model trained on ImageNet actually *improved* adversarial robustness roughly by 1%. Here we find a different effect, with the caveat of relatively different experimental setups from Kurakin et al. (2016).

In Figure 2(left) we show the effect label smoothing has on the performance of a model trained only on clean (*i.e.* non-adversarial) training data. Very surprisingly, using only label smoothing can result in a model that is nearly as robust as models trained with PGD-based adversarial training or adversarial logit pairing and take an order of magnitude less time to train – though we note that when PGD and ALP-based models are trained only on adversarial examples rather than a mixture of clean and adversarial data, their robustness exceeds this performance by around 5%. Furthermore, this benefit of label smoothing comes at no significant loss in accuracy on unperturbed test data, while generally, adversarial training tends to trade off original vs adversarial performance. Another curiosity is that adding in any label smoothing at all dramatically improves robustness to FGSM-based adversaries (adding label smoothing of $s = .01$ brought accuracy up from 6.1% to 38.3%), while PGD-based attacks saw much more gradual improvement.

Examining the logits (Figure 2, right), we see a striking difference between the models – the model trained with label smoothing both has a dramatically smaller dynamic range of logits and also presents a much more bimodal logit distribution than the model trained without label smoothing. In other words, it has learned to predict extremely consistent values for logits, which is what we attribute its adversarial robustness to. Anecdotally, we observed that this behavior held for all positive values of s, with a stronger effect the higher s was. Additional experiments with label smoothing are given in Section 5.3.

Paired-Example Data Augmentation Recently, a new form of data augmentation was found that, in contrast to standard label-preserving data augmentation, combined different training examples together, dramatically altering both the appearance of the training examples and their labels. Introduced concurrently by Zhang et al. (2018); Inoue (2018); Tokozume et al. (2018), these types of data augmentation typically have the form of element-wise weighted averaging of two input examples (typically images), with the training label also determined as a weighted average of the original two training labels (represented as one-hot vectors). Besides making target labels soft (*i.e.* not 1-of-K) during training time, these methods also encourage models to behave linearly between examples, which may improve robustness to out of sample data. Interestingly, Zhang et al. (2018) found that this type of data augmentation improved robustness to FGSM attacks on ImageNet (Russakovsky et al., 2015), but Kannan et al. (2018) found that the method did not improve robustness against a targeted attack with a stronger PGD-based adversary.

In our experiments we found evidence agreeing with both conclusions – when applying *mixup* (Zhang et al., 2018), we found a sizeable increase in robustness to FGSM adversaries, going from 6.1% on CIFAR-10 by training without *mixup* to 30.8% with *mixup*, but did not observe a significant change when evaluated against a PGD-based adversary. While robustness to a PGD adversary with only 5 steps increased by a tiny amount (from 0.0% to 0.5%), robustness to a 10-step PGD adversary remained at 0%. In our experiments, we use *VH-mixup*, the slightly improved version of *mixup* from Summers & Dinneen (2018).

4.1 Decoupling Adversarial Logit Pairing

While we have now considered alternate methods by which logits can be regularized, at this point it is still not clear exactly how they might be used with or interact with the logit regularization effect of adversarial logit pairing. Doing so requires separating out the logit pairing and logit regularization effects of ALP.

In adversarial logit pairing Kannan et al. (2018), the logit pairing term is implemented as an L^2 loss:

$$L(f(x^{(i)};\theta), f(g(x^{(i)});\theta)) = \|f(x^{(i)}) - f(g(x^{(i)})\|^2, \tag{9}$$

though other losses such as an L^1 or Huber loss are also possible. We would like to break L into separate pairing and regularization terms:

$$L(f(x^{(i)};\theta), f(g(x^{(i)});\theta)) = h(f(x^{(i)}), f(g(x^{(i)}))) + \beta(\|f(x^{(i)})\|^2 + \|f(g(x^{(i)})\|^2) \tag{10}$$

where the purpose of the first term is explicitly for making the logits more similar (with as little logit regularization as possible), and the second term is explicitly for regularizing the logits toward zero. There are several natural choices for h, such as the the Jensen-Shannon divergence, a cosine similarity, or any similarity metric that does not have a significant regularization effect. We have found that simply taking the cross entropy between the distributions induced by the logits was effective – depending on the actual values of the logits, this can either still have a mild squeezing effect (if the logits are very different), a mild expanding effect (if the logits are very similar), or something in between.

One implementation detail worth noting is that it can be difficult to reason about and set the relative strengths of the pairing loss and adversarial training loss. To that end, we set the strength of the pairing loss h as a constant fraction of the adversarial loss, implemented by setting the coefficient of the loss as a constant multiplied by a non-differentiable version of the ratio between the losses.

By decomposing adversarial logit pairing explicitly into logit pairing and logit regularization terms in this way, adversarial robustness to a 10-step PGD attack improves by an absolute 1.9% over ALP, or 5.6% over standard PGD-based adversarial training .

5 Additional experiments

5.1 Implementation Details

In the experiments on CIFAR-10 throughout this paper, we used a ResNet (He et al., 2016), equivalent to the "simple" model of Madry et al. (2018), with a weight decay of $2 \cdot 10^{-4}$ and a momentum

Table 1: White-box accuracy of models on CIFAR-10.

	Adversary				
Training Method	Natural	FGSM	PGD (5 steps)	PGD (10 steps)	PGD (20 steps)
Regular Training	87.4%	6.1%	0.0%	0.0%	0.0%
Label Smoothing	86.9%	54.7%	49.4%	41.7%	34.4%
PGD	75.8%	50.5%	51.0%	46.1%	45.3%
ALP	74.0%	52.6%	53.6%	49.1%	48.5%
LRM (ours)	68.5%	52.8%	53.8%	51.4%	51.0%

optimizer with strength of 0.9. Standard data augmentation of random crops and horizontal flips was used. After a warm up period of 5 epochs, the learning rate peaked at 0.1 and decayed by a factor of 10 at 100 and 150 epochs, training for a total of 200 epochs for models not trained on adversarial examples and 101 epochs for models using adversarial training – adversarial accuracy tends to increase for a brief period of time after a learning rate decay, then quickly drop by a small amount, an empirical finding also echoed by Schmidt et al. (2018). The minibatch size was 128.

Adversarial examples were constrained to a maximum L^{∞} norm of $.03$, and all PGD-based attacks used a step size of 0.0078. Adversarial attacks were constructed using the CleverHans library (Papernot et al., 2018), implemented in TensorFlow (Abadi et al., 2016). All experiments were done on two Nvidia Geforce GTX 1080 Ti GPUs.

5.2 Towards a more robust model

Given these forms of logit regularization, perhaps the most natural question is whether they can be combined to create an even more robust model. Thus, in this section we focus exclusively on making a model (and comparable baselines) as robust as possible to PGD-based attacks. In particular, for baseline methods (PGD-based adversarial training (Madry et al., 2018) and adversarial logit pairing (Kannan et al., 2018)), we opt to train exclusively on adversarial examples, effectively setting $\alpha = 0$ in Equation 1, which roughly trades off accuracy of $4 - 5\%$ for clean test examples for a similar gain in adversarial performance.

To combine the logit regularization methods together, we use a small amount of label smoothing ($s = 0.1$), use VH-mixup (Summers & Dinneen, 2018) on the input examples, use the logit pairing formulation of Section 4.1 with $\beta = 10^{-3}$, and set the ratio between the adversarial training loss and the pairing loss to 0.125, which focuses the loss on keeping adversarial and original examples similar. These parameters were not tuned much due to resource constraints. We refer to this combination simply as LRM ("Logit Regularization Methods").

White-box performance is shown in Table 1. LRM achieves the highest level of adversarial robustness of the methods considered for all PGD-based attacks, and to the best of our knowledge represents the most robust method on CIFAR-10 to date. However, like other adversarial defenses, this comes at the cost of performance on the original test set, which makes sense – from the perspective of adversarial training, a clean test image is simply the center of the set of feasible adversarial examples. Nonetheless, it is interesting that the tradeoff between adversarial and non-adversarial performance can continue to be pushed further, with the optimal value of that tradeoff dependent on application (*e.g.* whether worst-case performance is more important than performance on unperturbed examples).

Next, black-box performance is shown in Table 2. As is standard in most black-box evaluations of adversarial defenses, this is performed by generating adversarial examples with one model (the "Source") and evaluating them on a separate independently trained model (the "Target"). As found in other works (*e.g.* Madry et al. (2018)), the success of a black-box attack depends both on how similar the training procedure was between the source and target models and on the strength of the source model – for example, LRM uniformly results in a stronger black-box attack than ALP (Kannan et al., 2018), which itself is a uniformly stronger black-box attack than adversarial training with PGD Madry et al. (2018). As such, using LRM as the source mildly damages the black-box defenses of PGD and ALP.

Table 2: Black-box accuracy of models on CIFAR-10 with a 10-step PGD adversary.

	Source				
Target	Regular Training	Label Smoothing	PGD	ALP	LRM (ours)
Regular Training	26.8%	32.0%	75.1%	73.9%	67.3%
Label Smoothing	66.7%	67.3%	75.6%	74.2%	67.6%
PGD	69.8%	69.1%	58.0%	57.9%	55.7%
ALP	69.4%	68.8%	60.8%	59.3%	56.4%
LRM (ours)	71.5%	70.9%	60.5%	59.4%	54.3%

5.3 Label Smoothing Investigations

In both the white-box and black-box analyses, we found that label smoothing was surprisingly effective given its near-zero cost. For black-box attacks in particular (Table 2), label smoothing was generally among the most robust models across all different sources, with the label smoothing target network having the highest minimum performance across sources, a fact which was not shared even by any of the adversarially-trained models. Inspired by this, we conducted further investigation where we have found similarly surprising mixed behavior.

In this experiment, we investigated the robustness of a model trained with label smoothing to stronger white-box attacks, noting that performance on PGD-based attacks from Table 1 dropped considerably when moving from 5 iterations of PGD to 10 and 20 iterations. We found that this trend continued, with accuracy dropping all the way to 13.6% when performing a 200-iteration PGD attack, a trend that was not observed with any of the three models trained on PGD attacks. This suggests that label smoothing, while providing only a mild amount of worst-case adversarial robustness, can actually make the adversarial optimization problem much more challenging, which we believe is also the underlying reason for its effectiveness against black-box attacks. The exact mechanism by which it does this, however, remains elusive, which we think is an interesting avenue for further research in black-box defenses.

6 Discussion

In this work, we have shown the usefulness of logit regularization for improving the robustness of neural networks to adversarial examples. We first presented an analysis of adversarial logit pairing, the current state-of-the-art in adversarial defense, showing that roughly half of its improvement over adversarial training can be attributed to a non-obvious logit regularization effect. Based on this, we investigated two other forms of logit regularization, demonstrating the benefits of both, and then presented an alternative method for adversarial logit pairing that more cleanly decouples the logit pairing and logit regularization effects while also improving performance.

By combining these logit regularization techniques together, we were able to create both a stronger defense against white-box PGD-based attacks and also a stronger attack against PGD-based defenses, both of which come at almost no additional cost to PGD-based adversarial training. We also showed the surprising strength of label smoothing as a black-box defense and its corresponding weakness to only highly-optimized white-box attacks.

We anticipate that future work will push the limits of logit regularization even further to improve defenses against adversarial examples, possibly using more techniques originally devised for other purposes (Pereyra et al., 2017). We also hope that these investigations will yield insights into training adversarially-robust models without the overhead of multi-step adversarial training, an obstacle that has made it challenge to scale up adversarial defenses to larger datasets without a sizable computational budget.

REFERENCES

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pp. 265–283, 2016.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning*, 2018.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

Nilesh Dalvi, Pedro Domingos, Sumit Sanghai, Deepak Verma, et al. Adversarial classification. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 99–108. ACM, 2004.

Justin Gilmer, Ryan P Adams, Ian Goodfellow, David Andersen, and George E Dahl. Motivating the rules of the game for adversarial example research. *arXiv preprint arXiv:1807.06732*, 2018.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Hiroshi Inoue. Data augmentation by pairing samples for images classification. *arXiv preprint arXiv:1801.02929*, 2018.

Harini Kannan, Alexey Kurakin, and Ian Goodfellow. Adversarial logit pairing. In *Advances in Neural Information Processing Systems*, 2018.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

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

Nicolas Papernot, Fartash Faghri, Nicholas Carlini, Ian Goodfellow, Reuben Feinman, Alexey Kurakin, Cihang Xie, Yash Sharma, Tom Brown, Aurko Roy, Alexander Matyasko, Vahid Behzadan, Karen Hambardzumyan, Zhishuai Zhang, Yi-Lin Juang, Zhi Li, Ryan Sheatsley, Abhibhav Garg, Jonathan Uesato, Willi Gierke, Yinpeng Dong, David Berthelot, Paul Hendricks, Jonas Rauber, and Rujun Long. Technical report on the cleverhans v2.1.0 adversarial examples library. *arXiv preprint arXiv:1610.00768*, 2018.

Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. Regularizing neural networks by penalizing confident output distributions. In *International Conference on Learning Representations*, 2017.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. In *International Conference on Learning Representations*, 2018.

Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Mądry. Adversarially robust generalization requires more data. In *Advances in Neural Information Processing Systems*, 2018.

Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. In *International Conference on Learning Representations*, 2018.

Cecilia Summers and Michael J Dinneen. Improved mixed-example data augmentation. *arXiv preprint arXiv:1805.11272*, 2018.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Yuji Tokozume, Yoshitaka Ushiku, and Tatsuya Harada. Between-class learning for image classification. In *Computer Vision and Pattern Recognition*, 2018.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.