Adversarial Training with Bi-directional Likelihood Regularization for Visual Classification

Weitao Wan[1], Jiansheng Chen[1]⋆, and Ming-Hsuan Yang[2,3]

[1] Department of Electronic Engineering, Tsinghua University
[2] Department of EECS, UC Merced [3] Google Research

Abstract. Neural networks are vulnerable to adversarial attacks. Practically, adversarial training is by far the most effective approach for enhancing the robustness of neural networks against adversarial examples. The current adversarial training approach aims to maximize the posterior probability for adversarially perturbed training data. However, such a training strategy ignores the fact that the clean data and adversarial examples should have intrinsically different feature distributions despite that they are assigned with the same class label under adversarial training. We propose that this problem can be solved by explicitly modeling the deep feature distribution, for example as a Gaussian Mixture, and then properly introducing the likelihood regularization into the loss function. Specifically, by maximizing the likelihood of features of clean data and minimizing that of adversarial examples simultaneously, the neural network learns a more reasonable feature distribution in which the intrinsic difference between clean data and adversarial examples can be explicitly preserved. We call such a new robust training strategy the adversarial training with bi-directional likelihood regularization (ATBLR) method. Extensive experiments on various datasets demonstrate that the ATBLR method facilitates robust classification of both clean data and adversarial examples, and performs favorably against previous state-of-the-art methods for robust visual classification.

Keywords: Adversarial training, feature distribution, optimization.

1 Introduction

A key challenge for utilizing neural networks in visual classification is their vulnerability to adversarial examples, which has attracted increasing concerns in recent years [4,18,16,13]. Visual adversarial examples are crafted by adding small perturbations that are imperceptible to human eyes onto the clean data, causing the neural networks to produce wrong predictions. In addition, researches have demonstrated that adversarial examples can be transferable across different models [11,20], *i.e.* adversarial examples generated based on one model can successfully attack other models. As such, the existence of adversarial examples has become a serious threat to the safety of neural networks.

⋆ Corresponding author.

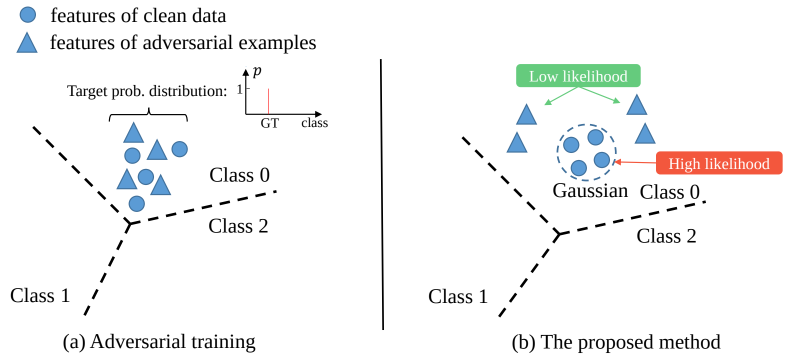

Fig. 1. Illustration of the expected feature space of (a) adversarial training and (b) the proposed ATBLR method. Adversarial examples are generated to resemble other classes. But existing adversarial training methods ignore their intrinsically different feature distribution and treat them equally with the clean data. The proposed method addresses this issue by optimizing not only the class probability distribution but also the likelihood of the feature distribution.

Improving the robustness of neural networks has become a critical issue in addition to increasing the classification accuracy. Numerous algorithms are proposed to address this issue, among which the most effective approaches are based on adversarial training [4,12]. The basic idea of adversarial training is to generate adversarial examples based on the latest model weights during training and feed them into the model for training. The adversarial examples are assigned the same class label as their source images. Madry *et al.* [12] propose a more generic form of adversarial training, which is formulated as a saddle-point optimization problem. However, the adversarial training only aims to optimize the posterior probability, without considering the feature distribution. The feature space of adversarial training is illustrated in Fig. 1(a). This paper focuses on the deepest features of neural networks, *e.g.* the output of the global average pooling layer after the last convolutional layer in ResNet [5]. Fig. 1(a) shows the expected feature space of adversarial training but it is difficult to achieve in practice because existing adversarial training methods ignore the intrinsic difference between the feature distributions of the clean data and adversarial examples. For instance, a clean sample from class 0 is adversarially perturbed into class 1. Previous research [6] justifies that such an adversarial example contains highly predictive but non-robust features for class 1. As such, its features should follow a different distribution compared to the features of the clean data from class 0. However, the adversarial training scheme ignores its similarity to class 1 and forces the neural network to treat it the same way as the clean data from class 0 by assigning them with the same target class distribution, which is typically a one-hot distribution of the ground-truth (GT) class. This unreasonable underlying constraint in existing adversarial training methods leads to sub-optimal classification performance.

To address this issue, we propose to optimize the neural networks so that not only the clean data and its corresponding adversarial examples can be classified into the same class but also the their feature distributions are explicitly encouraged to be different. The proposed method is illustrated in Fig. 1(b). To achieve this, we explicitly learn the feature distribution of the clean data by incorporating the Gaussian Mixture Model into the network. More specifically, for the visual classification task, features belonging to each class correspond to one Gaussian component, of which the Gaussian mean is the trainable parameter updated by stochastic gradient descent and the Gaussian covariance matrix is reduced to identity matrix for simplicity. As such, the entire network can be trained end-to-end. Then we adopt the likelihood regularization term introduced in [21] to encourage the extracted features of clean data to follow the learned Gaussian Mixture distribution. We note that the likelihood regularization in this paper is intrinsically different from that in [21] because our method takes two different types of inputs, *i.e.* the clean data and adversarial examples, and optimizes the likelihood term towards different directions for these two inputs. For the clean data, the objective is to maximize the likelihood since we aim to learn its feature distribution through training. For the adversarial examples, since they should follow a distribution that is different from the one of clean data, the objective is to minimize the likelihood. The common objective for both the clean data and adversarial examples is the cross-entropy loss for the posterior probability and the target class. We refer to the proposed method as Adversarial Training with Bi-directional Likelihood Regularization (ATBLR). We present a comparison study in Fig. 3, Section 4.3 to demonstrate that the proposed bi-directional likelihood regularization leads to different feature distributions for the clean data and adversarial examples.

Our method can be implemented efficiently, without increasing the number of trainable parameters. The classification layer in a neural network is typically a fully-connected layer with $K \times C$ trainable parameters, in which K is the number of object classes and C is the dimension of the features. It outputs the class distribution based on the features. Our method replaces it with a Gaussian Mixture Model without adding extra trainable parameters. Since this paper is focused on the visual classification task, the deepest features belonging to each class can be assigned with one Gaussian component. As such, the GMM also requires $K \times C$ trainable parameters in total for the K Gaussian components when the covariance is reduced to identity matrix as aforementioned. The likelihood regularization, which is essentially the l_2 distance between features and the corresponding Gaussian mean, brings about very little computational overhead to the neural networks.

The main contributions of this paper are summarized as follows:

- We propose the bi-directional likelihood regularization on the conventional adversarial training method based on the learned feature distribution. Features of the clean data and adversarial examples are explicitly encouraged to follow different distributions.

- We improve both the robustness of neural networks and the classification performance on clean data without adding extra trainable parameters.
- We evaluate the proposed method on various datasets including MNIST [10], CIFAR-10 and CIFAR-100 [8] for different adversarial attacks. Experimental results show that the proposed algorithm performs favorably against the state-of-the-art methods.

2 Related Work

2.1 Adversarial Attacks

Adversarial examples are crated data with small perturbations that cause misclassification in neural networks [18]. Plenty of algorithms have been developed to generate adversarial examples.

Fast Gradient Sign Method (FGSM). Goodfellow *et al.* [4] propose the Fast Gradient Sign Method (FGSM) which uses a single-step perturbation along the gradient of the loss function $\mathcal{L}$ with respect to the input image x. The adversarial example x_{adv} is computed by $x_{adv} = x + \epsilon \cdot sign(\nabla_x \mathcal{L}(x, y))$. To perform a targeted attack, we replace the true label y with a wrong target label t and reverse the sign of the gradient by $x_{adv} = x - \epsilon \cdot sign(\nabla_x \mathcal{L}(x, t))$.

Basic Iterative Method (BIM). Kurakin *et al.* [9] extends the single-step approach to an iterative attack which updates the adversarial example at each iteration by the formulation of FGSM method and clips the resulting image to constrain it within the ϵ-ball from the original input x. The adversarial example is computed by $x^i_{adv} = clip_{x,\epsilon}(x^{i-1}_{adv} + \alpha \cdot sign(\nabla_x \mathcal{L}(x^{i-1}_{adv}, y))$, where α is the step size for each iteration.

Projected Gradient Descent (PGD). Madry *et al.* [12] discover that stronger attacks can be generated by starting the iterative search of the BIM method from a random initialization point within the allowed norm ball centered at the clean data. This method is called the Projected Gradient Descent (PGD) method.

Carlini & Wagner (C&W). Nicholas *et al.* [3] propose the C&W algorithm which is an optimization-based attack method. An auxiliary variable ω is introduced to reparameterize the variable for adversarial example by $x_{adv} = \frac{1}{2}(\tanh(\omega) + 1)$ and solve $\min_\omega \|\frac{1}{2}(\tanh(\omega) + 1) - x\|^2_2 + c \cdot f(\frac{1}{2}(\tanh(\omega) + 1))$. The loss weight c is adjusted by binary search. And $f(x) = \max(\max\{Z(x)_i : i \neq t\} - Z(x)_t, -\kappa)$, in which $Z(x)_t$ is the logit for the target class t and the non-negative parameter κ controls the confidence for the adversarial example.

2.2 Defensive Methods

With the development of adversarial attack methods, the defense methods against them have attracted greater concerns in recent years. The defensive distillation approach [14] aims to train a substitute model with smoother gradients to increase the difficulty of generating adversarial examples. Nevertheless, it is not effective for the optimization-based attack methods such as [3]. Song *et al.* [17] propose to model the image distribution in the pixel space and restore an adversarial example to be clean by maximizing its likelihood. But it is difficult to effectively model the distribution in the pixel space where there is much noise and the dimension is much larger than that in the feature space. The adversarial training method [4,18] generates adversarial examples during training and use them as training data to improve the robustness against adversarial examples. However, this method is shown to be vulnerable to iterative attacks [9]. Tramer *et al.* [19] propose to improve the performance of adversarial training by generating adversarial examples using an ensemble of neural networks. Madry *et al.* [12] propose a more general framework for adversarial training and use random initialization before searching for adversarial examples to deal with iterative attack methods. Wong *et al.* [23] incorporate the linear programming into the training to minimize the loss for the worse case within the allowed perturbation around the clean data. However, the test accuracy on the clean data is severely compromised. Xie *et al.* [24] develop a network architecture which uses the non-local mean filter to remove the noises in the feature maps of adversarial examples. Song *et al.* [15] adopt the domain adaptation algorithms in adversarial training to learn domain-invariant representations across the clean domain and the adversarial domain. However, these methods are all based on the original adversarial training method and they do not address the issues concerning the feature distributions in the adversarial training which we discussed above.

3 Proposed Algorithm

3.1 Preliminaries

Adversarial Training. This paper focuses on the visual classification task. Suppose the number of object classes in the dataset is K. Denote the set of training samples as $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, in which $x_i \in \mathbb{R}^{H\times W\times 3}$ is the image, $y_i \in \{1, 2, ..., K\}$ is the class label and N is the number of training samples. Denote the one-hot label vector corresponding to label y_i as $\boldsymbol{y}_i$. Let $f_\theta(x) : \mathbb{R}^{H\times W\times 3} \to \mathbb{R}^K$ denote a neural network parameterized by θ. The network outputs the class probability distribution given an input image. Then the classification loss function for the training pair (x_i,y_i) is

$$\mathcal{L}_{cls}(x_i, y_i; \theta) = -\boldsymbol{y}_i \log f_\theta(x_i). \tag{1}$$

The adversarial training method [12] is formulated as a min-max optimization problem, which is expressed as

$$\min_\theta \max_{\|\delta_i\|_\infty \le \epsilon} \frac{1}{N} \sum_{(x_i,y_i)\sim\mathcal{D}} \mathcal{L}_{cls}(x_i + \delta_i, y_i; \theta). \tag{2}$$

The maximizer of the inner problem can be approximately found by using k steps of the PGD attack or a single-step FGSM attack. The adversarial examples are crafted by adding the inner maximizer to the clean data. The min-max problem is solved by stochastic gradient descent by feeding the adversarial examples as inputs to the neural network.

3.2 Modeling the Feature Distribution

As discusses in Section 1, our motivation is to consider the difference between feature distributions of clean data and adversarial examples. We adopt an effective and tractable distribution to model the feature distribution, *i.e.* the Gaussian Mixture Model. For simplicity, the covariance matrix is reduced to the identity matrix. This is not only efficient but also beneficial for reducing the redundancy across different feature dimensions. Besides, we assume the prior distribution for each class is the constant $1/K$. For the visual classification task, features belonging to each class are assigned with one Gaussian component. Formally, denote the features at the deepest layer of the neural network by

$$\tilde{x}_i = h_\theta(x_i), \tag{3}$$

in which $h_\theta(\cdot)$ represents the feature extraction process in the neural network. As such, the posterior probability of the ground-truth class y_i is expressed by

$$p(y_i|\tilde{x}_i) = \frac{\mathcal{N}(\tilde{x}_i; \mu_{y_i})}{\sum_{k=1}^{K} \mathcal{N}(\tilde{x}_i; \mu_k)}, \tag{4}$$

in which μ_k is the Gaussian mean of class k and $\mathcal{N}(\cdot)$ is the density function of Gaussian distribution.

The computation in Eq. 4 can be implemented with a layer in the neural network, with the Gaussian means as its trainable parameters. This layer is deployed immediately after the deepest features of the neural network and outputs the class distribution. The entire network can be trained end-to-end and the Gaussian means are updated by gradient descent through back-propagation. Equipped with such a layer, the neural network can learn to not only predict class probabilities but also model the feature distribution.

3.3 Likelihood Regularization for Features

The adversarial training scheme in Eq. 2 adopts only the adversarial examples for training, without using the clean data. In this paper, we leverage both the clean data and the adversarial examples generated by the inner problem in Eq. 2 for training, with equal proportion. We train the neural networks equipped with the layer introduced in Section 3.2 to learn the feature distribution of clean data. In addition, we adopt the likelihood regularization [21] to maximize the likelihood of features of clean data. Formally, the likelihood regularization is defined as the negative log likelihood, which is given by

$$\mathcal{L}_{lkd} = -\frac{1}{N}\sum_{i=1}^{N} \log \mathcal{N}(h_\theta(x_i); \mu_{y_i}). \tag{5}$$

By ignoring the constant term and constant coefficient, we derive

$$\mathcal{L}_{lkd} = \frac{1}{N}\sum_{i} \|h_\theta(x_i) - \mu_{y_i}\|^2. \tag{6}$$

The likelihood regularization is weighted by a hyperparameter $\lambda > 0$ and added to the cross-entropy loss during training. Hence, the final objective function for the clean data is given by

$$\mathcal{L} = \frac{1}{N}\sum_{(x_i,y_i)\sim\mathcal{D}} (-\boldsymbol{y}_i \log f_\theta(x_i) + \lambda\|h_\theta(x_i) - \mu_{y_i}\|^2). \tag{7}$$

We note that this formulation is essentially different from the center loss [22] because the center loss does not consider modeling the feature distribution. However, the mapping function $f_\theta(\cdot)$ here contains the Gaussian Mixture Model and the posterior probability is generated based on the learned feature distribution.

By minimizing Eq. 7, the neural network not only learns to make classifications but also learns to model the feature distribution of clean data. For the adversarial training, the clean data and adversarial examples are assigned the same class label but their feature distributions should be different since adversarial examples are crafted to resemble the class other than the ground-truth one and researches [6] reveal that they contain highly predictive but non-robust features of other classes. As such, a more reasonable training approach should encourage the features of clean data and adversarial examples to be different. This can be achieved by introducing the regularization term. We propose to maximize the likelihood value for adversarial examples during training. Denote the adversarial examples generated by solving the inner maximization problem in Eq. 2 as $\{a_i\}_{i=1}^N$, in which $a_i = x_i + \arg\max_{\delta_i} \mathcal{L}_{cls}(x_i + \delta_i, y_i; \theta)$. We minimize the following loss for adversarial examples.

$$\mathcal{L}_{adv} = \frac{1}{N}\sum_{i}(-\boldsymbol{y}_i \log f_\theta(a_i) - \lambda\|h_\theta(a_i) - \mu_{y_i}\|^2). \tag{8}$$

The training scheme is illustrated in Fig. 2. It demonstrates two important modifications we make to the original adversarial training scheme. First, the original adversarial training is conducted on a discriminative model, which only considers the output probability distribution and maximizes the target probability. In contrast, our method explicitly models the feature distribution through end-to-end training. Second, we explicitly encourage different feature distributions by optimizing the likelihood regularization towards opposite directions for the clean data and adversarial examples. Our method facilitates a more reasonable feature distribution in the scope of robust classification and improves the classification accuracy of both clean data and various adversarial examples.

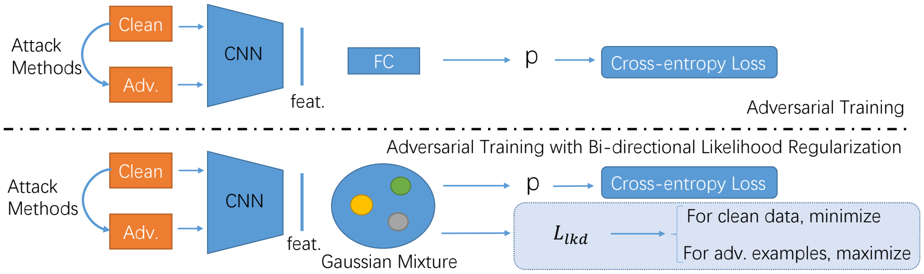

Fig. 2. Training schemes comparison. Top: the original adversarial training method [12]. Bottom: the proposed ATBLR method.

4 Experiments

To evaluate the robustness and generalization ability of the proposed method, we present experimental results on datasets including MNIST [10], CIFAR-10 and CIFAR-100 [8]. We report the natural accuracy, *i.e.* the accuracy of clean data, and that of adversarial examples. Following the widely adopted protocal [24,12,2], we consider the adversarial attack methods including FGSM [4], PGD [12] and C&W [3]. We evaluate the robustness of our method under two different threat models.

- **White-box attack**: the attacker has access to all the information of the target classification model including the model architecture and model weights. The adversarial examples for testing are generated using gradient information of the target model.
- **Black-box attack**: the attacker has knowledge of the model architecture but has no access to its model weights. The adversarial examples for testing are generated using the gradient information of a substitute model, which is independently trained using the same architecture and training hyper-parameters as the target model.

Experiments are conducted with the Tensorflow [1] using the Nvidia TITAN X GPU. All the codes and trained models will be made available to the public.

4.1 MNIST

We apply the proposed ATBLR method to train robust models for image classification to compare with the baseline method, *i.e.* the adversarial training [12]. The MNIST dataset [10] is a handwritten digit dataset consisting of 10 classes including 60,000 images for training and 10,000 images for testing. We use the data augmentation method including mirroring and 28×28 random cropping after 2-pixel zero paddings on each side. The models are tested on different types of adversarial examples, including the FGSM [4], PGD [12] with varying steps and restarts, C&W [3] with $\kappa = 0$ and C&W with a high confidence parameter $\kappa = 50$ (denoted as C&W-hc method).

Table 1. Classification accuracy (%) on the MNIST dataset for clean data and adversarial attacks. The evaluation is conducted for both white-box and black-box attacks.

Testing Input	Steps	Restarts	Accuracy (%)	
			Adv. Training [12]	ATBLR (ours)
Clean	-	-	98.8	**99.3**
White-box Attack				
FGSM	-	-	95.6	**97.2**
PGD	40	1	93.2	**94.8**
PGD	100	1	91.8	**94.1**
PGD	40	20	90.4	**93.5**
PGD	100	20	89.3	**92.7**
C&W	40	1	94.0	**95.8**
C&W-hf	40	1	93.9	**96.3**
Black-box Attack				
FGSM	-	-	96.8	**98.4**
PGD	40	1	96.0	**97.7**
PGD	100	20	95.7	**97.6**
C&W	40	1	97.0	**98.8**
C&W-hf	40	1	96.4	**98.5**

Implementation Details. Following the practice in [12], we generate PGD attacks of 40 steps during training and use a network consisting of two convolutional layers with 32 and 64 filters respectively, followed by a fully connected layer of size 1024. The input images are divided by 255 so that the pixel range is $[0, 1]$. The l_∞ norm constraint of $\epsilon = 0.3$ is imposed on the adversarial perturbations and the step size for PGD attack is 0.01. The models are trained for 50 epochs using ADAM [7] optimizer with a learning rate of 0.001. The parameter λ in Eq. 7 and 8 which balances the trade-off between the classification loss and bi-directional likelihood regularization is set to 0.1. For the evaluation of the black-box attacks, we generate adversarial examples on an independently initialized and trained copy of the target network according to [12].

The experimental results on the MNIST dataset are presented in Table 1. The results show that the strongest attack is the PGD attack with multiple restarts. It can be observed that the proposed method not only improves the robustness against adversarial attacks but also improves the accuracy of the clean data. Moreover, the performance gain is achieved without introducing any extra trainable parameters, which validates the effectiveness of addressing the difference of feature distributions between clean data and adversarial examples.

4.2 CIFAR

We apply the proposed ATBLR method to train robust classification models on the CIFAR-10 and CIFAR-100 datasets and make comparisons with previous

Table 2. Classification accuracy (%) on the CIFAR-10 dataset for clean data and adversarial attacks. The PGD attacks for testing are generated with l_∞ norm constraint $\epsilon = 8$ and a step size of 2. †We re-run the code of PATDA with $\epsilon = 8$ since it reports the result for $\epsilon = 4$ which generates weaker adversarial attacks.

Method	Clean	White-box				Black-box
		FGSM	PGD-10	PGD-100	PGD-1000	PGD-1000
Network: ResNet-32						
Natural Training	92.73	27.54	0.32	0.11	0.00	3.03
Adv. Training [12]	79.37	51.72	44.92	43.44	43.36	60.22
IAAT [2]	83.26	52.05	44.26	42.13	42.51	60.26
PATDA† [15]	83.40	53.81	46.59	45.27	44.01	61.79
FD [24]	84.24	52.81	45.64	44.60	44.21	62.84
ATBLR (ours)	86.32	**58.60**	**50.18**	**48.56**	**47.88**	**64.38**
Network: WideResNet-32						
Natural Training	95.20	32.73	2.17	0.35	0.00	4.29
Adv. Training [12]	87.30	56.13	46.26	45.14	44.87	61.07
IAAT [2]	91.34	57.08	48.53	46.50	46.54	58.20
PATDA† [15]	84.63	57.79	49.85	48.73	48.04	58.53
FD [24]	86.28	57.54	49.26	46.97	46.75	59.31
ATBLR (ours)	92.12	**59.69**	**52.11**	**51.17**	**50.63**	**62.89**

state-of-the-art methods. The CIFAR-10 dataset [8] consists of 32×32 pixel color images from 10 classes, with 50,000 training images and 10,000 testing images. The CIFAR-100 dataset [8] has 100 classes containing 50,000 training images and 10,000 testing images. We use the typical data augmentation method including mirroring and 32×32 random cropping after 4-pixel reflection paddings on each side. We use the network architectures of ResNet-32 [5] and WideRenset-32 [25] following Madry *et al.* [12] and Zhang *et al.* [26]. Our method is compared with the natural training and previous state-of-the-art training approaches designed to improve the robustness of classification models:

- Natural Training: Training with cross-entropy loss on the clean training data.
- Adversarial Training (Adv. Training) [12]: Training on the clean training data and the adversarial examples generated during training.
- Instance Adaptive Adversarial Training (IAAT) [2]: Training that enforces the sample-specific perturbation margin around every training sample.
- PGD-Adversairal Training with Domain Adaptation (PATDA) [15]: Adversarial training combined with domain adaptation algorithms.
- Feature Denoising (FD) [24]: Training that combines Adversarial Training and a network architecture with the non-local filters to remove the noise caused by the adversarial examples in feature space.

Implementation Details. During adversarial training, the adversarial examples are generated by PGD-10 attacks, *i.e.* 10 steps of PGD attack are conducted on the clean data for each training iteration. The step size for the PGD attack is set to 2 out of 255. A l_∞ norm constraint of $\epsilon = 8$ is imposed on the adversarial perturbations. The models are trained for 200 epochs using ADAM [7] optimizer with the learning rate of 0.001 and the batch size of 128. The parameter λ which balances the trade-off between the classification loss and bi-directional likelihood regularization is set to 0.02. We present more quantitative results in Section 4.4 to study the influence of the parameter λ. For evaluation in the white-box setting, the models are tested on (1) PGD-10 attacks with 5 random restarts, (2) PGD-100 attacks with 5 random restarts and (3) PGD-1000 attacks with 2 random restarts. For evaluation in the black-box setting, following the experimental setting of [2], the PGD-1000 attack with 2 random restarts is adopted.

The experimental results on the CIFAR-10 dataset are presented in Table 2. We observe that the proposed method improves the classification accuracy of both the clean data and adversarial examples. Compared with the original adversarial training, the results demonstrate that our method achieves large accuracy gain by considering the feature distribution differences and introducing the bi-directional likelihood regularization during training. Moreover, our method performs favorably against the Feature Denoising (FD) method [24], which is the previous state-of-the-art method. By switching from the network of ResNet-32 to its 10× wider variant, the classification performance is increased due to larger model capacity. Our method can increase the model's robustness for both the simple and complex models.

We present the experimental results on the CIFAR-100 dataset in Table 3. As shown by the results, the CIFAR-100 dataset is more challenging than the CIFAR-10 dataset. Nevertheless, we observe that the proposed ATBLR method consistently increases the robustness against adversarial examples and performs favorably against previous state-of-the-art methods.

4.3 Evolution of the Likelihood Regularization

During training, we propose to optimize different objective functions for clean data and adversarial examples, which are given by Eq. 7 and 8, respectively. Here we investigate the evolution of the values of $\mathcal{L}_{lkd}$ in the training progress to verify that the likelihood of clean data and adversarial examples is optimized to be different. We conduct the experiments on the CIFAR-10 dataset with the same network and training schemes as in Section 4.2. We evaluate and record the value of the likelihood regularization according to Eq. 6 for the clean data and adversarial examples, respectively, in each input batch. We compare two models. The first one is trained with the proposed ATBLR method and the second one is trained without optimizing the $\mathcal{L}_{lkd}$ during training.

The curves for the values of $\mathcal{L}_{lkd}$ are plotted in Fig. 3. We note that larger $\mathcal{L}_{lkd}$ indicates smaller likelihood value since $\mathcal{L}_{lkd}$ is essentially the negative logarithm of likelihood. In the left figure, as the training converges, the $\mathcal{L}_{lkd}$ of clean data (blue) is low, which means the network learns to model the feature distribution of the clean data. The $\mathcal{L}_{lkd}$ of adversarial examples (orange) is large, which is nearly twice that of the clean data. In contrast, the right figure shows that the $\mathcal{L}_{lkd}$ values of the clean data and adversarial examples are almost the

Table 3. Classification accuracy (%) on the CIFAR-100 dataset for clean data and adversarial attacks. The PGD attacks for testing are generated with l_∞ norm constraint $\epsilon = 8$ and a step size of 2.

Method	Clean	White-box				Black-box
		FGSM	PGD-10	PGD-100	PGD-1000	PGD-1000
Network: ResNet-32						
Natural Training	74.88	4.61	0.02	0.01	0.00	1.81
Adv. Training [12]	55.11	26.25	20.69	19.68	19.91	35.57
IAAT [2]	63.90	27.13	18.50	17.17	17.12	35.74
PATDA [15]	59.40	27.33	20.25	19.45	19.08	35.81
FD [24]	65.13	26.96	21.14	20.39	20.06	36.28
ATBLR (ours)	67.34	**28.55**	**21.80**	**21.54**	**20.96**	**37.79**
Network: WideResNet-32						
Natural Training	79.91	5.29	0.01	0.00	0.00	3.22
Adv. Training [12]	59.58	28.98	26.24	25.47	25.49	38.10
IAAT [2]	68.80	29.30	26.17	24.22	24.36	35.18
PATDA [15]	64.24	28.35	24.51	23.45	23.08	35.81
FD [24]	67.13	29.54	27.15	25.69	25.14	37.95
ATBLR (ours)	70.39	**30.85**	**29.49**	**27.53**	**27.15**	**39.24**

same during training. The comparison verifies that the proposed method effectively encourages the features of clean data and adversarial examples to follow different distributions. The quantitative evaluation in Section 4.2 validates that introducing such a regularization during training is effective in improving the accuracy of both clean data and adversarial examples.

In addition, we observe in the left figure that the two curves do not separate until about 700 iterations. This phenomenon can be explained as below. The model parameters are randomly initialized and the likelihood of both types of inputs is low at the start since the features are far from their corresponding Gaussian mean. At the early training stage, the cross-entropy loss in Eq. 7 and 8 is dominating because the λ is small. The cross-entropy loss is driving the features to move closer to the corresponding Gaussian mean, thus decreasing the $\mathcal{L}_{lkd}$ for both types of inputs. As the cross-entropy loss becomes smaller, the likelihood regularization is having a larger impact. After a certain point of equilibrium, which is about 700 iterations in this experiment, the likelihood regularization term in Eq. 8 is slowing $\mathcal{L}_{lkd}$ from decreasing for adversarial examples. Finally, the training converges and the $\mathcal{L}_{lkd}$ value of adversarial examples is larger than that of clean data, which means that the clean data follows the learned GM distribution better and the adversarial examples follow a different distribution.

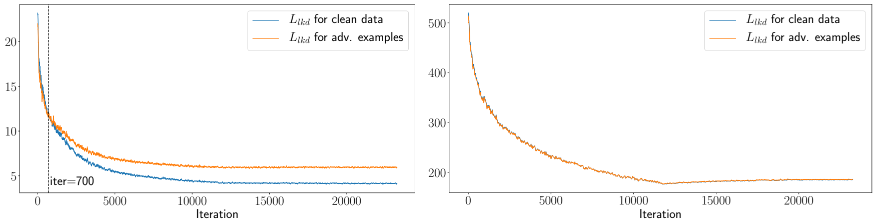

Fig. 3. Curves of the likelihood regularization for clean data and adversarial examples. **Left**: the model is trained using the proposed ATBLR method. **Right**: the model is trained without optimizing $\mathcal{L}_{lkd}$ but we record its value during training. The experiment is conducted on the CIFAR-10 dataset with ResNet-32. The first 60 epochs are shown here since the changes in the rest 140 epochs are not obvious.

4.4 Hyper-parameter Analysis

We study the effect of choosing different values of λ in the proposed ATBLR method and compare the performance. We conduct the experiments on the CIFAR-10 and CIFAR-100 datasets using the ResNet-32 network.

The experimental results are presented in Fig. 4. The PGD attack is stronger than FGSM. Nevertheless, our method improves the classification performance for different types of attacks and the clean data. Comparing results of $\lambda = 0$ and the others, we conclude that the ATBLR method can improve the classification performance consistently for different values of the hyper-parameter λ. The results also demonstrate that it is disadvantageous to set a λ that is too large. This is reasonable considering that λ is a balancing coefficient for the classification loss and the likelihood regularization. Too large a λ will lead to the dominance of the likelihood regularization term. This damages the classification performance because the features of all the clean data tend to collapse into one point under this objective function. Nevertheless, our method makes steady improvements for different λ in a wide range. We choose $\lambda = 0.02$ for experiments in Section 4.2 based on this hyper-parameter study.

4.5 Adversaries for Training

In the previous experiments, following other works, we select PGD attacks with 10 steps as the adversarial examples for training. We investigate the effect of adopting other alternatives and present the results on the CIFAR-10 dataset in Table 4. The results show that the performance gain that our method achieves becomes larger when stronger attacks are used for training. For example, if the training adversaries are switched from PGD-10 to the stronger PGD-100, the performance gain on clean data is increased from $86.32 - 79.37 = 6.95$ to $86.49 - 77.45 = 9.04$, likewise in other columns. This is expected because stronger adversarial examples have greater similarity to another class. As such, it is more favorable if they are encouraged to follow a feature distribution which is different from that of clean data of the original class.

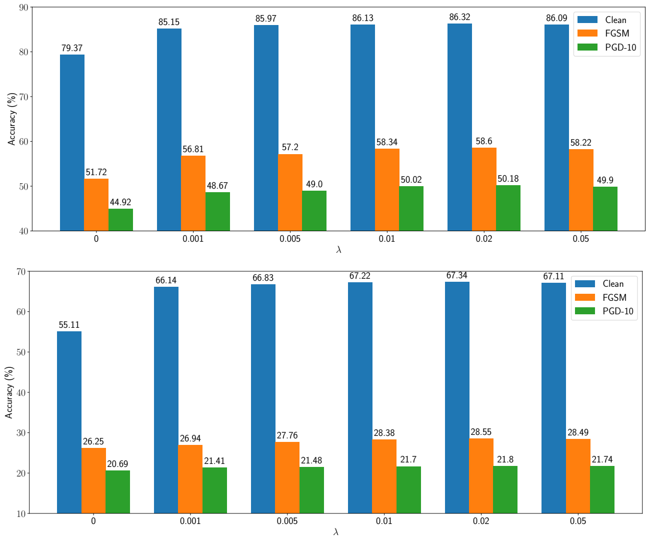

Fig. 4. Hyper-parameter study for λ on the CIFAR-10 dataset (top) and the CIFAR-100 dataset (bottom). $\lambda = 0$ denotes the original adversarial training.

Table 4. Classification accuracy (%) of the proposed ATBLR method / the original adversarial training when trained with different adversaries.

Model	Clean	FGSM	PGD-10	PGD-100
Training w/ FGSM	89.83/87.40	91.87/90.93	1.03/0.00	0.14/0.00
Training w/ PGD-10	86.32/79.37	58.60/51.72	50.18/44.92	48.56/43.44
Training w/ PGD-100	86.49/77.45	58.74/51.58	51.25/45.06	52.37/45.71

5 Conclusion

In this paper, we propose a novel method for training robust classification models against adversarial attacks. In contrast to the previous adversarial training method which optimizes only the posterior class distribution, our method learns the feature distribution of clean data through end-to-end training. Furthermore, the intrinsic difference between feature distributions for clean data and adversarial examples is preserved by optimizing the likelihood regularization in opposite directions for these two types of inputs. Moreover, our method introduces no extra trainable parameters. Extensive experiments demonstrate that our method performs favorably against previous state-of-the-art methods in terms of the classification accuracy of both the clean data and adversarial examples.

Acknowledgements. This work was supported by the National Natural Science Foundation of China under Grant 61673234 and the program of China Scholarships Council (No. 201906210354). M.-H. Yang is supported in part by NSF CAREER Grant 1149783.

References

1. Abadi, M., et al.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015), https://www.tensorflow.org/, software available from tensorflow.org 8
2. Balaji, Y., Goldstein, T., Hoffman, J.: Instance adaptive adversarial training: Improved accuracy tradeoffs in neural nets. arXiv preprint arXiv:1910.08051 (2019) 8, 10, 11, 12
3. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: IEEE Symposium on Security and Privacy (SP) (2017) 4, 5, 8
4. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014) 1, 2, 4, 5, 8
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016) 2, 10
6. Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., Madry, A.: Adversarial examples are not bugs, they are features. In: NeurIPS (2019) 2, 7
7. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 9, 11
8. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Tech. rep., University of Toronto (2009) 4, 8, 10
9. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial machine learning at scale. arXiv preprint arXiv:1611.01236 (2016) 4, 5
10. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE (1998) 4, 8
11. Liu, Y., Chen, X., Liu, C., Song, D.: Delving into transferable adversarial examples and black-box attacks. arXiv preprint arXiv:1611.02770 (2016) 1
12. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: ICLR (2018) 2, 4, 5, 8, 9, 10, 12
13. Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z.B., Swami, A.: The limitations of deep learning in adversarial settings. In: IEEE European Symposium on Security and Privacy (EuroS&P) (2016) 1
14. Papernot, N., McDaniel, P., Wu, X., Jha, S., Swami, A.: Distillation as a defense to adversarial perturbations against deep neural networks. In: IEEE Symposium on Security and Privacy (SP) (2016) 5
15. Song, C., He, K., Wang, L., Hopcroft, J.E.: Improving the generalization of adversarial training with domain adaptation. In: ICLR (2019) 5, 10, 12
16. Song, D., Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Tramer, F., Prakash, A., Kohno, T.: Physical adversarial examples for object detectors. In: 12th USENIX Workshop on Offensive Technologies (WOOT) (2018) 1
17. Song, Y., Kim, T., Nowozin, S., Ermon, S., Kushman, N.: Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. arXiv preprint arXiv:1710.10766 (2017) 5
18. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013) 1, 4, 5
19. Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I., Boneh, D., McDaniel, P.: Ensemble adversarial training: Attacks and defenses. arXiv preprint arXiv:1705.07204 (2017) 5
20. Tramèr, F., Papernot, N., Goodfellow, I., Boneh, D., McDaniel, P.: The space of transferable adversarial examples. arXiv preprint arXiv:1704.03453 (2017) 1

21. Wan, W., Zhong, Y., Li, T., Chen, J.: Rethinking feature distribution for loss functions in image classification. In: CVPR (2018) 3, 6
22. Wen, Y., Zhang, K., Li, Z., Qiao, Y.: A discriminative feature learning approach for deep face recognition. In: ECCV (2016) 7
23. Wong, E., Kolter, J.Z.: Provable defenses against adversarial examples via the convex outer adversarial polytope. ICML (2018) 5
24. Xie, C., Wu, Y., Maaten, L.v.d., Yuille, A.L., He, K.: Feature denoising for improving adversarial robustness. In: CVPR (2019) 5, 8, 10, 11, 12
25. Zagoruyko, S., Komodakis, N.: Wide residual networks. arXiv preprint arXiv:1605.07146 (2016) 10
26. Zhang, H., Yu, Y., Jiao, J., Xing, E.P., Ghaoui, L.E., Jordan, M.I.: Theoretically principled trade-off between robustness and accuracy. arXiv preprint arXiv:1901.08573 (2019) 10