Auxiliary Training: Towards Accurate and Robust Models

Linfeng Zhang[12], Muzhou Yu[23], Tong Chen[1], Zuoqiang Shi[1], Chenglong Bao[1]*, Kaisheng Ma[1]*
[1]Tsinghua University, [2]Institute for interdisciplinary Information Core Technology
[3]Xi'an Jiaotong University

Abstract

Training process is crucial for the deployment of the network in applications which have two strict requirements on both accuracy and robustness. However, most existing approaches are in a dilemma, i.e. model accuracy and robustness forming an embarrassing tradeoff – the improvement of one leads to the drop of the other. The challenge remains for as we try to improve the accuracy and robustness simultaneously. In this paper, we propose a novel training method via introducing the auxiliary classifiers for training on corrupted samples, while the clean samples are normally trained with the primary classifier. In the training stage, a novel distillation method named input-aware self distillation is proposed to facilitate the primary classifier to learn the robust information from auxiliary classifiers. Along with it, a new normalization method - selective batch normalization is proposed to prevent the model from the negative influence of corrupted images. At the end of the training period, a L_2-norm penalty is applied to the weights of primary and auxiliary classifiers such that their weights are asymptotically identical. In the stage of inference, only the primary classifier is used and thus no extra computation and storage are needed. Extensive experiments on CIFAR10, CIFAR100 and ImageNet show that noticeable improvements on both accuracy and robustness can be observed by the proposed auxiliary training. On average, auxiliary training achieves 2.21% accuracy and 21.64% robustness (measured by corruption error) improvements over traditional training methods on CIFAR100. Codes have been released on github.

1. Introduction

Dramatic achievements have been attained with the help of deep learning in various domains, including computer vision [17, 25, 35, 26], natural language processing [2, 40, 7] and so on. However, image corruption, which can be widely observed in real-world application scenarios like rotation, blurring, raining, and noises, leads to a severe accuracy degradation due to the vulnerability of neural networks. A simple and effective method to improve model robustness is data augmentation [21, 38]. However, directly adding corrupted images into training set always leads to unacceptable accuracy drop on clean images [47]. Moreover, model robustness for different kinds of corruptions always influences each other. For instance, Gaussian noise data augmentation leads to robustness increment on noise corruption but reduces model robustness on the images with different contrast and saturation [43]. Most recently, one research trend is to improve model robustness without scarifying accuracy on clean data [16, 27], yet, it's still challenging to develop a training approach that improves both accuracy and robustness simultaneously.

In this work, we propose a novel neural networks training framework named auxiliary training which consists of two types of training samples. One is the clean images from a dataset and the other is the corrupted images which are generated by adding corruptions to clean images. The corruptions in this paper consist of noise, blur and other formats of image corruption. In our training framework, given a network, the feature extraction layer is kept but auxiliary classifiers which are copies of the final classifier layer (denoted as primary classifier) are introduced for helping training the primary classifier. In the first stage of training, both two kinds of images are fed into the same convolutional layers to obtain representative features but each individual classifier is only trained by samples from a certain kind of corruption. In the second stage, a L_2-norm loss is applied for penalizing the weights between the primary classifier and auxiliary classifiers such that they attain the identical weights. As a result, the auxiliary classifiers can be dropped and only the primary classifier is kept. Therefore, the original network architecture does not change and extra computations and parameters are needless in the inference period. Figure 1 illustrates the flow of our approach.

Moreover, we propose the input-aware self distillation and selective batch normalization to facilitate model training. The input-aware self distillation regards the primary classifier as the teacher model, and auxiliary classifiers as

*Corresponding authors, {kaisheng,clbao}@mail.tsinghua.edu.cn

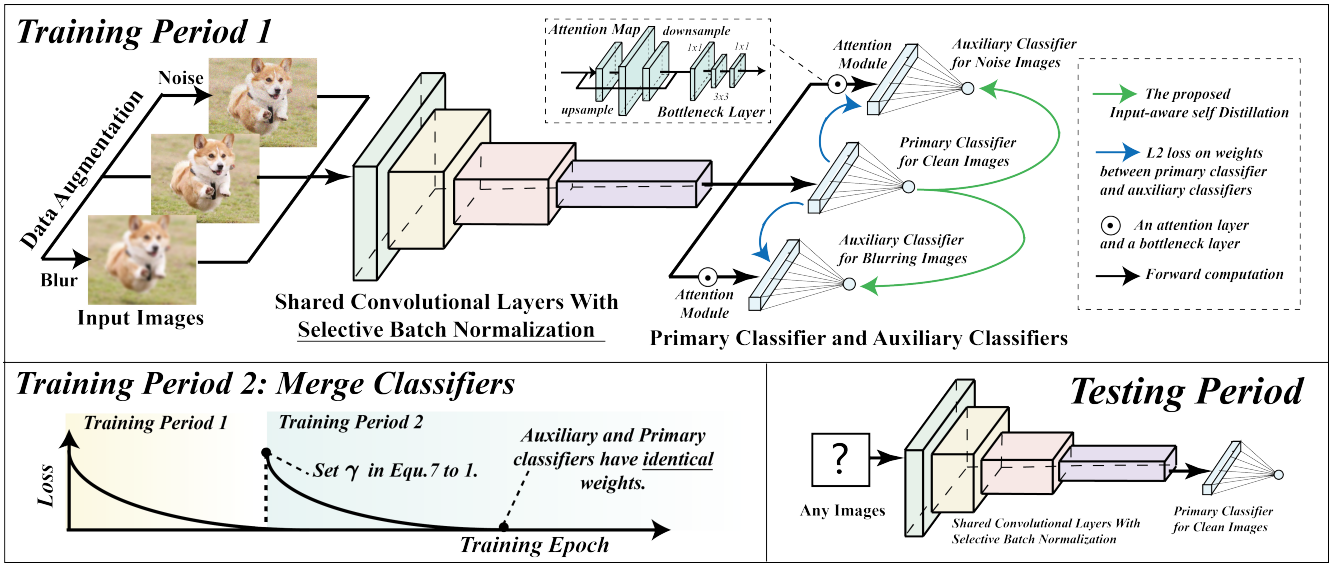

Figure 1. Details of the proposed auxiliary training. (a)Training Period 1: (i) The images involved in the training include the clean images from datasets, and corrupted (e.g. blurring, noise) images generated from data augmentation. (ii) All of the images are fed into the same convolutional layers with the proposed selective batch normalization to obtain representative features. (iii) The features of clean images are then fed into the primary classifiers, which is composed of one fully connected layer. The features of corrupted images are then fed into the auxiliary classifiers, which consist of an attention module and a fully connected layer. (b) Training Period 2: (iv) At the end of the training period, a L_2 loss is utilized to force weights of auxiliary classifiers to orientate the primary classifier until they have the exactly identical weights. (c) Testing Period: (v) In testing period, all the inputs images are classified by the primary classifier, and the auxiliary classifiers can be dropped to reduce model parameters.

students models, transferring knowledge from clean images to corrupted images and enabling the primary classifier to learn robust information from auxiliary classifiers. The selective batch normalization computes the mean and variance of clean images and corrupted images respectively and updates its parameters only by clean images, avoiding the negative influence from corrupted images. An ablation study of the aforementioned techniques is introduced in Table 9 to show their effectiveness respectively.

Besides, the formulation of the proposed auxiliary training is motivated from the connections of perturbations between the input space and parameter space. As the corrupted image can be seen as a small perturbation in the feature space, it is equivalent to a small perturbation of parameters by using first order approximation. Thus, it naturally leads to the soft constraints between primary classifier and auxiliary classifier which guarantee the mathematical rationale of our approach. A detailed analysis is given in Section 3.1. In summary, our main contributions are as follows:

- A neural network's training framework named auxiliary training is proposed, achieving noticeable accuracy and robustness improvements **with no additional computation or storage requirements during inference**. Experiments on CIFAR100 show that 2.21% accuracy and 21.64% robustness improvements can be observed compared to traditional training methods.
- Two effective techniques including input-aware self distillation and selective batch normalization are proposed to further improve the performance of the proposed auxiliary training, which provides fruitful insights in multi-exits neural network design.
- The mathematical formulation of the auxiliary training is derived from the perspective of perturbation analysis and its rationality is further explained by the context of knowledge distillation and learning with privileged information.

The rest of the paper is organized as follows. Section 2 introduces related work in model robustness and multi-task training. The formulation of the proposed training framework, input-aware self distillation and selective batch normalization are presented in Section 3. Experiments and the discussions are shown in Section 4 and 5 respectively. The conclusion is given in Section 6.

2. Related work

2.1. Model robustness

Model robustness, which indicates the stability of model performance to the perturbations in input data, is one of the most challenging topics in machine learning. Generally speaking, these perturbations can be divided into two

Algorithm 1 Auxiliary Training

Input: Training Data X, Model M with weights θ
Data Augmentation Transformation T, Labels Y
Loss function L from Equ.7, Learning Rate σ
Output: Model M with weights θ

while not converged **do**
  sample a batch of samples x, y from X, Y
  p := M.predict($\{x, T(x)\}; y; \theta$)
  $\theta := \theta - \sigma \cdot \frac{\partial L(p;\theta)}{\partial \theta}, \gamma = 0$
end while
while not converged **do**
  sample a batch of samples x, y from X, Y
  p := M.predict($\{x, T(x)\}; y; \theta$)
  $\theta := \theta - \sigma \cdot \frac{\partial L(p;\theta)}{\partial \theta}, \gamma = 1$
end while
Drop the auxiliary classifiers in M

Return M **and** θ

groups by their source: the adversarial perturbations from manual attack algorithm [11, 30] and common images corruption from nature such as noise, blurring, and rotation. As introduced by Laugros *et al.*, there is a significant difference and little connection between the two kinds of perturbations [23]. In this paper, we mainly focus on the robustness of common images corruption. Recently, more and more attention has been paid to improve the robustness of neural networks. Fruitful benchmarking datasets have been released to estimate the robustness of neural networks on image classification [15], object detection [33], semantic segmentation [19] and video processing [37]. To improve model robustness, Zheng *et al.* proposed stability training, which forces the corrupted images to have similar prediction results with the clean images [46]. Hendrycks *et al.* exploited self-supervised learning by predicting the rotation angels, improving model robustness on 19 kinds of common corruption [16]. Galloway *et al.* found that the batch normalization may be the cause of model accuracy degradation on corrupted images [10]. Gontijo Lopes *et al.* combined the "cut out" [8] methods with Gaussian noise, improving model robustness from a data augmentation perspective. Yin *et al.* explained the trade-offs among models performance of various corruption by Fourier spectrum [43], then proposed a method to improve model robustness on common corruptions based on Auto-augmentation [5]. Compared to the above works, our work is an alternative training framework by introducing corruption-specific auxiliary classifiers that help improve both accuracy and robustness for the primary classifier.

2.2. Learning with auxiliary tasks

Since the tasks in the real world are usually closely related, neural networks are always designed and expected to solve various related tasks at the same time, which is named multi-task learning [1, 29, 34, 4]. As one of the classical examples, mask RCNN [13] targets to localize, classify and segment an object with one shared backbone neural network. During the training period, the three tasks benefit from each other, leading to a dramatic accuracy increment for all. An effective method in multi-task learning is to construct additional auxiliary tasks to facilitate the training of the primary task. In this situation, the auxiliary classifiers are meaningless in the application but helpful in models training. Meyerson *et al.* proposed pseudo tasks augmentation, in which several pseudo classifiers are trained based on the same features [32]. In Self supervised GAN method, an auxiliary classifier for the discriminator was proposed to predict the rotation angles of images, facilitating the training of GAN [3]. The proposed auxiliary training can be understood from the perspective of multi-task learning when corruption samples are auxiliary tasks and it partially explains the success of our training approach. Moreover, a technique for imposing the identical weight constraints is specially designed during the training process.

3. Auxiliary training

3.1. Formulation

Let $\mathcal{X}_C = \{(x_i, y_i)\}_{i=1}^m$ be a set of clean training samples, $\mathcal{T} = \{T_1, T_2, \ldots, T_t\}$ be a set of corruption operations and $\mathcal{X}_j = \{(x_i^j, y_i) | x_i^j = T_j(x_i), (x_i, y_i) \in \mathcal{X}_c\}$ to be the corrupted training set by j-th corruption. Thus, the whole training set consists of

$$\mathcal{X} = \mathcal{X}_C \cup (\cup_{j=1}^t \mathcal{X}_j) = \cup_{j=0}^t \cup_{i=1}^m \{(x_i^j, y_i)\},$$

where $x_i^j = T_j(x_i)$ and assume T_0 is the identity map. Let $f(x; \theta_f)$ be the feature extractor which can be a convolutional neural network and $g(x; \theta_g)$ be the classifier and the feature map associated with the j-th corruption be

$$\hat{x}^j = f(T_j(x), \theta_f), \ \forall j = 0, 1, \ldots, t.$$

The traditional augmentation training method seeks for the best parameters via the following minimization:

$$\min_{\theta_f, \theta_g} \frac{1}{t+1} \sum_{j=0}^{t} \left\{ \frac{1}{m} \sum_{i=1}^{m} \ell(g(\hat{x}_i^j(\theta_f); \theta_g), y_i) \right\}, \quad (1)$$

where $\ell(\cdot, \cdot)$ is the loss function, e.g. L2-norm, cross entropy and Kullback-Leibler (KL) divergence and where $\hat{x}_i^j(\theta_f) = f(T_j(x_i), \theta_f)$ denotes the feature map of i-th sample corrupted by j-th corruption. In (1), all the corruptions are treated equally which might not be consistent with the true distribution. Thus, assume the probability of the j-th corruption is $\alpha_j = p(T_j)$ and introduce the auxiliary

classifier $g(x;\theta_g^j)$ for each corruption, the minimization (1) can be formulated as

$$\min_{\theta_f,\{\theta_g^j\}_{j=0}^t} \sum_{j=0}^{t} \alpha_j \left\{ \frac{1}{m} \sum_{i=1}^{m} \ell(g(\hat{x}_i^j(\theta_f);\theta_g^j), y_i) \right\} \triangleq L_1, \quad \text{s.t.} \quad \theta_g^0 = \theta_g^1 = \cdots = \theta_g^t. \tag{2}$$

Due to the existence of nonconvexity in (2), finding a stationary point of (2) with high accuracy and robustness is difficult. In order to facilitate training for obtaining a desired classifier, we train the primary classifier by only clean samples and propose to introduce auxiliary classifiers such that each one is only trained by specific corruption samples. Finally, we merge the information from the auxiliary classifiers by the regularization. More concretely, assume $g(\cdot;\theta_g^0)$ is our desired primary classifier and the classifier $g(\cdot;\theta_g^j)$ is well trained for j-th corruption, then it implies

$$g(\hat{x}^0;\theta_g^0) = g(\hat{x}^j;\theta_g^j). \tag{3}$$

Let j-th corruption be parameterized by ξ, then in feature space, we have $\delta x^j = \hat{x}^j - \hat{x}^0$ in a neighborhood of $\hat{x}^0$. When the capacity of the feature extraction network is large enough such that it can learn certain invariant features for the corruptions, i.e. δx^j is small, the first order Taylor expansion implies that

$$g(\hat{x}^j;\theta_g^j) \approx g(\hat{x}^0;\theta_g^j) + \frac{\partial g}{\partial x}\delta x^j. \tag{4}$$

If there is a small perturbation $\delta\theta_g^j$ such that

$$\frac{\partial g}{\partial x}\delta x^j \approx \frac{\partial g}{\partial \theta}\delta\theta_g^j. \tag{5}$$

Together with (3), (4), (5), we arrive at a necessary condition for robust primary classifier $g(\cdot;\theta_f)$:

$$g(\hat{x}^j,\theta_g^0) \approx g(\hat{x}^j,\theta_g^0 + \delta\theta_g^j),$$

by the first-order approximation, i.e. our auxiliary classifiers are $\theta_g^j = \theta_g^0 + \delta\theta_g^j$ and the perturbation is implicitly given by the corruption for the input images. Therefore, the trajectory of the corruptions for clean samples corresponds to a trajectory of the robust classifier. Therefore, to achieve the robustness of the primary classifier, it could be better to choose it smooth along the tangent direction of the trajectory of corruption. However, it is hard to analyze the tangent direction of the perturbations in feature space. Instead, we impose the smoothness of the primary classifier around θ_g^0, i.e.

$$\theta_g^0 \approx \theta_g^j, \quad g(\hat{x}^0,\theta_g^0) \approx g(\hat{x}^j,\theta_g^j).$$

This motivates us to relax the equality constraints by the penalty function Ω as:

$$\Omega(\theta_g^0,\theta_g^j) = \ell_{KL}(g(\hat{x}^0;\theta_g^0), g(\hat{x}^j;\theta_g^j)) + \gamma\|\theta_1 - \theta_2\|_2^2. \tag{6}$$

Therefore, the total loss in our auxiliary training is:

$$\min_{\theta_f,\{\theta_g^j\}_{j=0}^t} L_1 + \lambda \sum_{j=1}^{t} \Omega(\theta_g^0,\theta_g^j). \tag{7}$$

There are three hyperparameters α, λ, γ in (7) and all of them are fixed for all the experiments in this paper. Our experiments demonstrate that **the proposed auxiliary training is not sensitive to the value of hyperparameters.**

3.2. Rationality of auxiliary training

In this section, we further analyze the auxiliary training from the following two perspectives.

Input-aware self distillation. The knowledge distillation consisting of the teacher-student structure has proved to be a useful method for the accuracy improvement. However, the performance depends on how "smart" the teacher is. In practice, it is difficult to find a universal "smart" teacher. In the proposed auxiliary training, the "decentralization" idea is applied for encouraging knowledge communication among classifiers. More concretely, each classifier is only trained by the data with certain augmentation and the penalty term Ω defined in (6) imposes the knowledge transfer between primary classifier and auxiliary classifiers under the simultaneous training strategy. In other words, each classifier can be seen as a domain expert and they are learned from each other. Therefore, instead of the teacher-student structure, the auxiliary training approach is more likely to be a "student⇄student" framework which is more efficient for knowledge transfer.

Privileged information. The framework of learning using privileged information is first introduced in [39] and it is connected to the knowledge distillation in [28]. Let (x_i, x_i^*, y_i) be the i-th training sample where (x_i, y_i) be the feature-label pair and x_i^* is the additional information of x_i provided by the teacher network. In our proposed auxiliary learning framework, as both clear sample x_i^0 and corrupted samples $x_i^j, j = 1, 2, \ldots, t$ share the same label information, the privileged information can be $x_i^* = f(x_i^0;\theta_f)$ where f is a feature extractor. In the generalized distillation framework [28], the primary classifier is the teacher and the auxiliary classifiers are students. As a good feature extractor f can provide certain invariant property for the corrupted images, it is reasonable that the auxiliary classifier is relatively easy to learn in feature space which leads to better generalization error [28]. From this perspective, it motivates that the proposed architecture contains a common feature extractor but different classifiers for corruptions.

3.3. Techniques for auxiliary Training

Figure 1 and Algorithm 1 show the details of the proposed auxiliary training. Two techniques are proposed to

Model	Our approach	Baseline	Increment
AlexNet	91.43	88.28	+3.15
ResNet18	96.02	94.75	+1.27
ResNet50	96.31	95.22	+1.09
ResNet101	96.47	95.27	+1.20
WRN50	96.49	95.42	+1.07
ResNeXt50	96.34	95.59	+0.75

Table 1. Comparison of accuracy (%) between models trained by auxiliary training and standard training on CIFAR10 dataset. WRN indicates wide ResNet.

Model	Our approach	Baseline	Increment
AlexNet	70.09	68.44	+1.65
ResNet18	79.47	77.09	+2.38
ResNet50	80.16	77.42	+2.74
ResNet101	80.51	77.81	+2.70
WRN50	80.84	79.08	+1.76
ResNeXt50	81.51	79.49	+2.02

Table 2. Comparison of accuracy (%) between models trained by auxiliary training and standard training on CIFAR100 dataset. WRN indicates wide ResNet.

facilitate both the robustness and accuracy of neural networks, which are introduced as follows.

Auxiliary classifiers. Different from the primary classifier which is a single fully connected layer, the auxiliary classifiers in this paper are constructed by three components: an attention module, a bottleneck layer, and a fully connected layer, according to the shallow classifiers in SCAN [45]. The attention modules consist of one convolutional layer and one deconvolutional layer, aiming at helping the auxiliary classifiers obtain the useful features [41, 24]. A bottleneck layer [14], which is composed of 1x1, 3x3, 1x1 convolutional layers, is attached after the attention modules. Since all the auxiliary classifiers are only utilized in the training period, **they don't bring additional storage and computation in inference period.**

Selective batch normalization. Batch normalization [18] is widely utilized in all kinds of convolutional neural networks to stabilize the training of models. However, recently, Galloway *et al.* found that batch normalization reduces model robustness on both adversarial attacks and corrupted images [10]. Zhou *et al.* show that models with batch normalization may not outperform models without batch normalization, especially when data augmentation is utilized in the training period. Their experiments demonstrate that batch normalization leads to 2.9% accuracy drop on ResNet32 trained on CIFAR10 with data augmentation [47].

To alleviate the accuracy degradation from batch normalization on corrupted data, we propose the selective batch normalization (SBN), aiming at eliminating the influence from corrupted data in batch normalization. The proposed

Model	Our approach	Baseline	Increment
AlexNet	69.98	100.00	+30.02
ResNet18	57.01	85.91	+28.90
ResNet50	58.15	84.26	+26.11
ResNet101	50.03	87.08	+37.05
WRN50	59.43	87.19	+27.76
ResNeXt50	52.96	84.50	+31.54

Table 3. Comparison of robustness between models trained by auxiliary training and normal training on CIFAR10-C dataset. WRN indicates Wide ResNet. Model robustness is measured by corruption error (CE) in Equation (8). **Less is better.**

SBN is based on the observation that the statistics parameters of batch normalization are vulnerable to the shift in inputs data, i.e., the corruption in inputs images. With the proposed SBN, the mean and variance of corruption data and clean data are computed respectively in both training and inference period.

Let $\mathcal{X}^b$ be a training batch sampling from $\mathcal{X}$. The training batch is composed of clean samples $\mathcal{X}_C^b$ and corrupted samples $\mathcal{X}_j^b$, which can be formulated as $\mathcal{X}^b = \mathcal{X}_C^b \cup (\cup_{j=1}^t \mathcal{X}_j^b)$. In traditional batch normalization methods, the features of clean samples and corrupted samples are computed together, which can be formulated as

$$\widetilde{x} = \gamma \cdot \frac{\widetilde{x} - E[\mathcal{X}^b]}{\sqrt{Var[\mathcal{X}^b] + \varepsilon}} + \beta, x \in \mathcal{X}^b,$$

where γ and β are two parameters for scaling and shifting trained by back propagation. ε is a number with small value to avoid zero-division error and $\widetilde{x}$ denotes the features in convolutional layers of sample x. Comared with traditional batch normalization, the proposed SBN computes clean and corrupted samples respectively, which can be formulated as

$$\widetilde{x} = \gamma \cdot \frac{\widetilde{x} - E[\mathcal{X}_C^b]}{\sqrt{Var[\mathcal{X}_C^b] + \varepsilon}} + \beta, x \in \mathcal{X}_C^b$$

$$\widetilde{x} = \gamma \cdot \frac{\widetilde{x} - E[\mathcal{X}_j^b]}{\sqrt{Var[\mathcal{X}_j^b] + \varepsilon}} + \beta, x \in \mathcal{X}_j^b$$

In inference period, the $E[\cdot]$ and $Var[\cdot]$ are replaced by statistics means μ and variance σ^2. In the training period of traditional batch normalization, μ and σ^2 are updated by the both clean and corrupted samples in the batch, which can be formulated as

$$\mu \leftarrow \frac{1}{n}\sum_{i=1}^{n} x, \sigma^2 \leftarrow \frac{1}{n}\sum_{i=1}^{n}(x-\mu)^2, x \in \mathcal{X}^b, n = |\mathcal{X}^b|$$

In contrast, the proposed SBN updates μ and σ^2 by only the clean samples, which can be formulated as

$$\mu \leftarrow \frac{1}{n}\sum_{i=1}^{n} x, \sigma^2 \leftarrow \frac{1}{n}\sum_{i=1}^{n}(x-\mu)^2, x \in \mathcal{X}_C^b, n = |\mathcal{X}_C^b|$$

Model	Our approach	Baseline	Increment
AlexNet	80.03	100.00	+19.97
ResNet18	69.34	92.21	+22.87
ResNet50	69.13	92.28	+23.15
ResNet101	66.10	88.35	+22.25
WRNet50	68.89	87.33	+18.44
ResNeXt50	69.13	92.29	+23.16

Table 4. Comparison of robustness between models trained by auxiliary training and normal training on CIFAR100-C dataset. WRN indicates Wide ResNet. Model robustness is measured by corruption error (CE) in Equation (8). **Less is better.**

Model	Top-1	Top-5
ResNet18-Standard Training	69.21	89.01
ResNet18-Auxiliary Training	69.94	89.51
ResNet34-Standard Training	73.17	91.24
ResNet34-Auxiliary Training	74.14	91.94

Table 5. Comparison of accuracy (%) between models trained by auxiliary training and normal training on ImageNet.

Model	Training Method	Accuracy	CE
ResNet18	Baseline	94.75	85.91
ResNet18	Self-Supervised [16]	95.23	63.09
ResNet18	Gaussian Patch [27]	95.13	63.17
ResNet18	Data Augmentation	93.53	60.03
ResNet18	Auxiliary Training	**96.02**	**57.01**
Wide ResNet50	Baseline	95.42	87.19
Wide ResNet50	Self-Supervised [16]	95.47	63.77
Wide ResNet50	Gaussian Patch [27]	95.66	66.63
Wide ResNet50	Data Augmentation	93.86	60.64
Wide ResNet50	Auxiliary Training	**96.49**	**55.41**
ResNeXt50	Baseline	95.59	84.50
ResNeXt50	Self-Supervised [16]	95.67	61.86
ResNeXt50	Gaussian Patch [27]	95.52	60.13
ResNeXt50	Data Augmentation	93.72	50.52
ResNeXt50	Auxiliary Training	**96.34**	**49.37**

Table 6. Comparison between the proposed auxiliary training with other robustness training methods on CIFAR10 and CIFAR10-C. Model robustness is measured by corruption error (CE) in Equation (8).(**less is better**). Numbers in bold are the best.

4. Experiments results

4.1. Experiments settings

Experiments of the proposed auxiliary training are conducted on four kinds of convolutional neural networks, including AlexNet [21], ResNet [14], Wide ResNet [44] and ResNeXt [42] and three kinds of datasets, including CIFAR10, CIFAR100 [20] and ImageNet [6]. Moreover, robustness benchmark datasets including CIFAR-C and ImageNet-C [15] datasets are utilized to evaluate model robustness in 19 kinds of common image corruption, containing all kinds of noise, blur, weather and so on.

In the training period, normal data augmentation consisting of random cropping and horizontal flipping are utilized to improve the performance of neural networks. The SGD optimizer with weight decaying and momentum is exploited to train the models. Models on both CIFAR10 and CIFAR100 are trained by 300 epochs, with learning rate divided by 10 in the 100_{th}, 200_{th}, 290_{th} epoch. Models on ImageNet are trained by 90 epochs, with learning rate divided by 10 in the 30_{th}, 60_{th} epoch. The default hyper-parameters setting in this paper is: $\alpha_0 = 1, \alpha_{\neq 0} = 0.05, \lambda = 0.05, \gamma \in \{0, 1\}$. All the experiments are conducted by PyTorch1.2.0, running on RTX 2080 and Tesla V100 GPU devices. There are total four data augmentations in the experiments in this paper and they are Gaussian noise, Gaussian blur, rotation and images' contrast and brightness.

In this paper, the robustness of neural networks is measured by the relative value between the error rate of neural networks and AlexNet. It's named the corruption error (CE) [15], which is computed by the following formula

$$CE_{Network} = Error_{Network}/Error_{AlexNet} \quad (8)$$

where $Error$ denotes the error rates. A lower CE indicates that neural networks have more robustness.

4.2. Experiments on CIFAR and CIFAR-C

Improvements on accuracy. Table 1 and Table 2 show the accuracy of neural networks by auxiliary training on CIFAR10 and CIFAR100, respectively. It can be observed that: (i) In CIFAR10, 1.43% accuracy increment can be observed on the models trained with auxiliary training, ranging from 0.75% on ResNeXt50 as the minimum to 3.15% on AlexNet as the maximum. (ii) In CIFAR100, 2.21% accuracy increment can be detected on the models with the proposed auxiliary training, ranging from 2.74% on ResNet50 as the maximum to 2.74% on Wide ResNet50 as the minimum. (iii) Compared with the advanced models such as ResNeXt and Wide ResNet, more accuracy gain can be observed on the ResNet and AlexNet models.

Improvements on robustness. Table 3 and 4 show the experiments results of six neural networks on CIFAR10-C and CIFAR100-C. It can be observed that (i) The proposed auxiliary training leads to consistent and significant robustness improvements. On average, there are 30.15% and 21.64% CE improvements on CIFAR10-C, CIFAR100-C respectively. (ii) Although many kinds of corruption such as snow, fog, and JPEG compression are not involved in the training period, experiments show auxiliary training also improves model robustness in these corrupted images, indicating **there is a good generalization ability of the proposed auxiliary training to various corruption.**

Comparison with related work. The comparison between the proposed auxiliary training and the other three robust training methods is shown in Table 6. It's observed that

Model	Mean CE	Noise				Blur					Weather				Digital					
		Gauss.	Speckle	Shot	Impulse	Glass	Gauss.	Zoom	Motion	Defocus	Snow	Frost	Fog	Bright.	Satura.	Contra.	JPEG.	Elastic.	Spatter.	Pixelate.
ResNet18	84.11	87	85	89	91	90	85	88	87	84	84	85	79	72	70	81	91	91	80	79
+Aux. Training	78.86	79	78	81	82	89	79	84	83	85	79	80	72	66	74	75	78	89	75	70
ResNet34	85.54	87	86	89	89	93	86	88	90	91	85	87	81	73	72	81	89	93	81	81
+Aux. Training	75.58	78	76	80	81	85	77	82	79	77	75	76	70	64	62	73	77	84	71	66

Table 7. Comparison of robustness between models trained by auxiliary training and normal training on ImageNet-C dataset. Model robustness is measured by corruption error (CE) in Equation (8).(**less is better**). "+Aux. Training" indicates that the models are trained by the proposed auxiliary training.

Training Method	Clean	PGD-L_2	PGD-L_∞	BIA-L_2	BIA-L_∞	FGSM	MIA-L_2	DDN-L_2
Normal Training	94.75	23.37	4.88	24.62	6.49	18.34	24.62	1.42
Adversarial Training [31]	83.90	45.54	43.52	79.94	44.88	51.99	74.04	24.36
Auxiliary Training	85.76	49.35	46.45	82.56	47.07	54.38	76.97	26.53

Table 8. Comparison of adversarial training and the proposed auxiliary training with several adversarial attack, ResNet18 on CIFAR10. PGD Attack [30], Basic Iterative Attack [22], Fast Gradient Sign Method [12], Momentum Iterative Attack [9], Decoupled Direction and Norm Attack [36].

(i) Data augmentation can improve model robustness at the expense of model accuracy. (ii) Some robust training methods such as self supervised training and Gaussian patch can improve model robustness with almost no sacrificing of accuracy. (iii) In contrast, the proposed auxiliary training can improve both accuracy and robustness simultaneously and outperform the other three robust training methods by a large margin.

4.3. Experiments on ImageNet and ImageNet-C

Experiments on ImageNet are also conducted to show the effectiveness of auxiliary training on large scale datasets. Table 5 and Table 7 show the accuracy and robustness of four neural networks on ImageNet. On average, 0.85% top-1 and 0.60% top-5 accuracy increment on ImageNet and 7.61% CE (robustness) improvement on ImageNet-C can be observed.

4.4. Experiments on adversarial attack

Although the proposed auxiliary training is designed for the robustness to nature corruption, experiments show that it also leads to accuracy gain on adversarial attack. In this experiment, the primary classifier is trained on adversarial samples by PGD [30], and the auxiliary classifiers are still trained on nature corruption images. PGD attack, basic iterative attack [22], FGSM attack [12], momentum iterative attack [9] and the decoupled direction and norm attack [36] are utilized to evaluate model accuracy and robustness to adversarial attack.

As shown in Table 8: (i) The proposed auxiliary training outperforms the state-of-the-art defense methods - adversarial training [31] by a large margin, on both clean data accuracy and adversarial samples accuracy. (ii) 1.86% clean data accuracy improvements can be observed in the proposed auxiliary training compared with the adversarial training. (iii) 3.17% accuracy improvements on 7 kinds of adversarial attack methods can be observed in auxiliary training. The consistent and significant improvements indicate that the proposed auxiliary training method can also be utilized in the defense to the adversarial attack.

5. Discussion

5.1. Ablation study

Besides the auxiliary classifiers, there are mainly four kinds of techniques utilized in the proposed auxiliary training, i.e., selective batch normalization, input-aware self distillation, attention modules, and weights merging. To investigate their effectiveness, a series of experiments are conducted to show models' accuracy and robustness when they are trained by the auxiliary training without one of the above techniques.

Training Method	Accuracy	CE
Auxiliary Training	79.47	69.34
w/o Selective BN	76.37	69.52
w/o Self Distillation	78.44	73.67
w/o Attention	77.50	70.79
w/o Weight Merging	78.32	70.43

Table 9. An ablation study of the proposed auxiliary training with ResNet18 on accuracy (CIFAR100) and robustness (CIFAR100-C). Model robustness is measured by the corruption error in Equ.(8). **Less is better.**

As is shown in Table 9, compared with the complete auxiliary training: (i) Consistent and significant accuracy and

robustness drop can be observed on any models trained with incomplete auxiliary training. (ii) 3.1% accuracy drop and 0.18% corruption error rate increment on CIFAR100 can be observed if the selective batch normalization is not utilized in auxiliary training. The reason may come from the fact that joint training of both clean and corrupted images prevent models training on clean images from better convergence. (iii) 1.03% accuracy drop and 4.33% corruption error rate increment are observed on the auxiliary training models without input-aware self distillation, demonstrating that the primary classifiers can obtain more benefits of robustness information from the auxiliary classifiers. (iv) 1.93% accuracy drop and 1.45% corruption error rate increment can be observed on the models trained without the attention module, which might be explained by the reason that attention modules can facilitate the auxiliary classifiers to learn the corruption images better. (vi) Models trained by auxiliary training without weights merging leads to 1.15% accuracy drop and 0.79% corruption error rate increment, which may be explained by that loss on classifiers' weights that enables the primary classifier to learn from the auxiliary classifiers directly. In brief, all the techniques in the proposed auxiliary training are effective and indispensable.

5.2. Sensitivity study in frequency domain

To further prove the robustness gain by auxiliary training, a frequency perturbation experiment is conducted [43]. As shown in Figure 2, the frequency perturbation consists of three steps: At first, a discrete Fourier transformation (DFT) is applied to the input images and one point in the frequency domain is perturbed by some constant value. Finally, we obtain the perturbed image by applying the inverse Fourier transformation (IDFT).

As a result of the above perturbation in frequency domain, the relationship between model robustness and frequency information can be visualized. In Figure 3, two ResNet18 models are trained with and without the proposed auxiliary training on CIFAR100 and then evaluated on testing set with frequency perturbation on different frequency. In Figure 3, the value on the pixel in the i_{th} row and the j_{th} column of each sub-figure indicates model accuracy on images with frequency perturbation on the pixel in the i_{th} row and the j_{th} column. It's observed that: (i) The ResNet model trained by auxiliary training outperforms the model trained by standard training methods by a large margin on frequency perturbation in all the pixels, indicating that consistent and significant robustness can be obtained by auxiliary training. (ii) With both standard training methods and the proposed auxiliary training, models show more robustness on low frequency perturbation and less robustness on high frequency perturbation, indicating the models are sensitive to the high frequency perturbation such as noise.

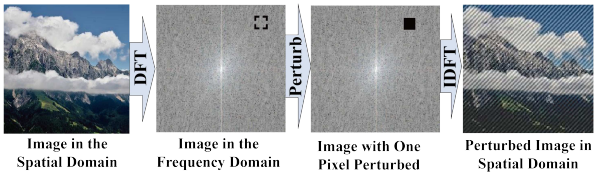

Figure 2. The process of frequency perturbation with 2D discrete Fourier transformation (DFT). Images are first transformed into the frequency domain from the spatial domain and then perturbed by a constant value on one pixel. Finally they're transformed back to the spatial domain. The perturbed pixel in the figure is marked by the black square.

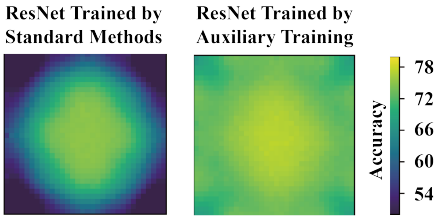

Figure 3. Accuracy heat maps of two ResNet18 models in the frequency perturbation sensitivity study. The value of the pixel in the i^{th} row and j^{th} column indicates model accuracy on CIFAR100 testing sets with frequency perturbation on the pixel in the i^{th} row and the j_{th} column.

6. Conclusion

In this paper, we propose an auxiliary training framework, which can improve both model accuracy and robustness with no additional computation and parameters in inference period. In auxiliary training, both clean images and corrupted images are fed into the neural networks, computed by the shared convolutional layers but with different classifiers. At the end of training, all the classifiers are converged to an identical one due to the L_2 loss on their weights. The proposed auxiliary training is also mathematically grounded, which can be formulated as a method which applies the penalty function methods to solve the optimization problem of neural networks training.

Moreover, further improvements on model accuracy and robustness can be achieved by the proposed selective batch normalization and input-aware self distillation. An ablation study is conducted to verify the effectiveness of each technique and a frequency perturbation sensitivity study shows that the auxiliary training can promote model robustness to image corruption in all frequency. Substantial experiments on CIFAR, CIFAR-C, ImageNet, ImageNet-C, and 7 kinds of adversarial attack methods demonstrate that the significance and generality of the proposed auxiliary training.

Acknowledgement. This work was partially supported by IIISCT (Institute for interdisciplinary Information Core Technology), National Natural Sciences Foundation of China (No.31970972 and 11901338), and Tsinghua University Initiative Scientific Research Program.

References

[1] Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Multi-task feature learning. In *NeurIPS*, pages 41–48, 2007. 3

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015. 1

[3] Ting Chen, Xiaohua Zhai, Marvin Ritter, Mario Lucic, and Neil Houlsby. Self-supervised gans via auxiliary rotation loss. In *CVPR 2018*, 2018. 3

[4] Koby Crammer and Yishay Mansour. Learning multiple tasks using shared hypotheses. In *Advances in Neural Information Processing Systems*, pages 1475–1483, 2012. 3

[5] Ekin Dogus Cubuk, Barret Zoph, Dandelion Mané, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation policies from data. *ArXiv*, abs/1805.09501, 2018. 3

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 6

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2018. 1

[8] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 3

[9] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018. 7

[10] Angus Galloway, Anna Golubeva, Thomas Tanay, Medhat Moussa, and Graham W. Taylor. Batch normalization is a cause of adversarial vulnerability. *ArXiv*, abs/1905.02161, 2019. 3, 5

[11] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572, 2014. 3

[12] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 7

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask r-cnn. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017. 3

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 5, 6

[15] Dan Hendrycks and Thomas G. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *ArXiv*, abs/1903.12261, 2019. 3, 6

[16] Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Xiaodong Song. Using self-supervised learning can improve model robustness and uncertainty. *ArXiv*, abs/1906.12340, 2019. 1, 3, 6

[17] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, pages 4700–4708, 2017. 1

[18] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *ArXiv*, abs/1502.03167, 2015. 5

[19] Christoph Kamann and Carsten Rother. Benchmarking the robustness of semantic segmentation models. *ArXiv*, abs/1908.05005, 2019. 3

[20] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009. 6

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, pages 1097–1105, 2012. 1, 6

[22] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016. 7

[23] Alfred Laugros, Alice Caplier, and Matthieu Ospici. Are adversarial robustness and common perturbation robustness independant attributes ? *ArXiv*, abs/1909.02436, 2019. 3

[24] Shikun Liu, Edward Johns, and Andrew J. Davison. End-to-end multi-task learning with attention. *ArXiv*, abs/1803.10704, 2018. 5

[25] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, pages 21–37, 2016. 1

[26] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 1

[27] Raphael Gontijo Lopes, Dong Yin, Ben Poole, Justin Gilmer, and Ekin Dogus Cubuk. Improving robustness without sacrificing accuracy with patch gaussian augmentation. *ArXiv*, abs/1906.02611, 2019. 1, 6

[28] David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. Unifying distillation and privileged information. In *ICLR*, 2016. 4

[29] Karim Lounici, Massimiliano Pontil, Alexandre B Tsybakov, and Sara Van De Geer. Taking advantage of sparsity in multitask learning. *arXiv preprint arXiv:0903.1468*, 2009. 3

[30] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *ArXiv*, abs/1706.06083, 2017. 3, 7

[31] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 7

[32] Elliot Meyerson and Risto Miikkulainen. Pseudo-task augmentation: From deep multitask learning to intratask sharing—and back. In Jennifer Dy and Andreas Krause, editors, *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pages 3511–3520, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. 3

[33] Claudio Michaelis, Benjamin Mitzkus, Robert Geirhos, Evgenia Rusak, Oliver Bringmann, Alexander S. Ecker, Matthias Bethge, and Wieland Brendel. Benchmarking robustness in object detection: Autonomous driving when winter is coming. *ArXiv*, abs/1907.07484, 2019. 3

[34] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *CVPR*, pages 3994–4003, 2016. 3

[35] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 1

[36] Jérôme Rony, Luiz G Hafemann, Luiz S Oliveira, Ismail Ben Ayed, Robert Sabourin, and Eric Granger. Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4322–4330, 2019. 7

[37] Vaishaal Shankar, Achal Dave, Rebecca Roelofs, Deva Ramanan, Benjamin Recht, and Ludwig Schmidt. A systematic framework for natural perturbations from videos. *ArXiv*, abs/1906.02168, 2019. 3

[38] Patrice Y Simard, David Steinkraus, John C Platt, et al. Best practices for convolutional neural networks applied to visual document analysis. In *ICDAR*, volume 3, 2003. 1

[39] Vladimir Vapnik and Akshay Vashist. A new learning paradigm: Learning using privileged information. *Neural networks*, 22(5-6):544–557, 2009. 4

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017. 1

[41] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *CVPR*, pages 3156–3164, 2017. 5

[42] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, pages 5987–5995, 2017. 6

[43] Dong Yin, Raphael Gontijo Lopes, Jonathon Shlens, Ekin Dogus Cubuk, and Justin Gilmer. A fourier perspective on model robustness in computer vision. *ArXiv*, abs/1906.08988, 2019. 1, 3, 8

[44] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016. 6

[45] Linfeng Zhang, Zhanhong Tan, Jiebo Song, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Scan: A scalable neural networks framework towards compact and efficient models. *ArXiv*, abs/1906.03951, 2019. 5

[46] Stephan Zheng, Yang Song, Thomas Leung, and Ian Goodfellow. Improving the robustness of deep neural networks via stability training. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 4480–4488, 2016. 3

[47] Aojun Zhou, Yukun Ma, Yudian Li, Xiaohan Zhang, and Ping Luo. Towards improving generalization of deep networks via consistent normalization. *ArXiv*, abs/1909.00182, 2019. 1, 5