2024 International Joint Conference on Neural Networks (IJCNN) | 979-8-3503-5931-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/IJCNN60899.2024.10651480

LoFT: LoRA-Based Efficient and Robust Fine-Tuning Framework for Adversarial Training

Jiadong Fu^{1,2}, Jiang Fang^{1,2}, Jiyan Sun^{1,*}, Shangyuan Zhuang^{1,2}, Liru Geng^{1,2}, Yinlong Liu^{1,2}
^{1}Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
^{2} School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
*Corresponding Author: sunjiyan@iie.ac.cn

***Abstract*—Recently, Self-Supervised Learning (SSL) has achieved great success in various famous applications e.g., BERT and ChatGPT. However, when applying SSL to safety-critical downstream tasks, such as self-driving cars, potential adversarial attacks can completely change the final decisions and thus lead to serious security issues. To overcome this issue, existing methods combine adversarial training with pre-training to improve the adversarial robustness of SSL. However, combining these two computationally complex processes may largely amplify the computation cost. Moreover, whether performing adversarial training in pre-training or fine-tuning, current methods may degrade the accuracy due to the famous catastrophic forgetting problem. The computation cost of current adversarial training methods based on full parameter updating is still high even in the fine-tuning stage.**

To address the above challenges, we propose an effective robust fine-tuning framework for SSL based on Low-Rank Adaptation (LoRA), named LoFT. First, LoFT performs adversarial training in the fine-tuning stage rather than in the pre-training stage. Second, LoFT innovatively and elaborately integrates LoRA into adversarial training to avoid the catastrophic forgetting problem. Third, LoFT exploits a low-rank matrix in LoRA which enables efficient fine-tuning by updating only a small set of parameters, which contains only 1%-5% of the parameters of the pre-trained model. The whole pre-training and fine-tuning stages take only 9.44 hours, which reduces training time by 3× over the current SOTA method. Furthermore, compared with existing SOTA robust pre-training methods for SSL, LoFT improves accuracy by 5.97% (77.41%⇒83.38%) and robustness by 13% (45.04%⇒58.04%) on the CIFAR-10 dataset.

***Index Terms*—Adversarial Training, Self Supervised Learning, Robust Fine Tuning, Low-Rank Adaptation**

I. Introduction

In recent years, self-supervised learning (SSL) has become the most important technique for pre-trained models and their extended or enhanced models which are generally called large models [1]. All of the models follow the “pre-training and fine-tuning” learning paradigm. Combining with the paradigm, SSL achieves great success in famous pre-trained models like BERT [2], GPT-2 [3] and well-known large models like LLaMA [4], ChatGPT [5]. The typical processes of the “pre-training and fine-tuning” learning paradigm are shown in Fig. 1. In the pre-training stage, SSL is used to learn the intrinsic features and structure of massive unlabeled data by constructing supervised tasks that compare similarities or differences of different data pairs. This enables the pre-trained models outstanding generalization abilities after the pre-training stage [6]. Consequently, the pre-trained models can be effectively applied to a wide range of downstream tasks through the fine-tuning stage [7].

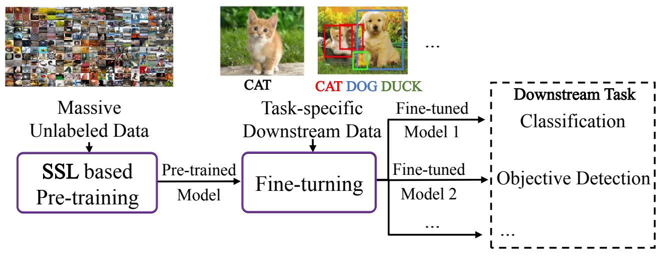

Fig. 1. Typical processes of “pre-training and fine-tuning” learning paradigm.

Despite the advantages and successes of SSL, researchers demonstrate that SSL is very vulnerable to adversarial attack as a typical deep learning technology [8]–[10]. Adversarial attacks can completely change the output of deep learning models by adding subtle but intentional perturbations to original data. For example, AutoAttack [11] only adds a perturbation of 8/255 to the image. Then it reduces the model’s accuracy by 87.41% (94.02% ⇒ 6.61%). Therefore, security is a huge issue when applying SSL-based pre-trained models to safety-critical downstream tasks such as autonomous vehicles [12] and healthcare [13].

To improve the robustness of SSL-based pre-trained models against adversarial attacks, many studies [14]–[19] introduce adversarial training in the pre-training stage to improve the adversarial robustness of the SSL based pre-trained model. Adversarial training is one of the most effective defences against adversarial attacks. It generates adversarial samples by simulating potential adversarial attacks. Then, in the pre-training stage, the adversarial samples are used for model training along with the natural samples which are collected from the original data. After the pre-training stage, the pre-trained model has learned more robust and generalized features [20], [21] to gain both high accuracy and robustness.

However, there exist three main challenges when introducing adversarial training in the pre-training stage. First, the SSL-based pre-training usually requires up to 1,000 training epochs [22]. This process combined with adversarial training may largely amplify the computation cost of pre-training. Second, these methods may degrade the accuracy of the model in downstream tasks i.e., *catastrophic forgetting problem* [23]. This is because adversarial training requires a model to learn features from both adversarial and natural samples. Previous research finds that the distributions of the natural and adversarial samples are usually mismatched, fitting these two data samples can lead to a degradation of accuracy [23]. Third, these methods may be limited in improving the adversarial robustness of downstream tasks. Unlike the fine-tuning stage, it has no clear attack target when simulating potential adversarial attacks in the pre-training stage [19]. Thus the model may not be able to adequately learn how to cope with various adversarial attacks in the specific downstream task leading to a performance gap between pre-training and fine-tuning.

To alleviate the computation cost and performance gap, some researchers introduce adversarial training in the fine-tuning stage to improve the adversarial robustness of SSL-based pre-trained models rather than in the pre-training stage [24]–[26]. For these methods, the pre-trained model is robustly fine-tuned by performing adversarial training with all parameters updated in the fine-tuning stage. However, updating all parameters still requires significant computation cost though in the fine-tuning stage [27], [28]. Moreover, updating all parameters may make the catastrophic forgetting problem more serious with clear attack target based on labeled downstream data [23].

To address the above challenges, we propose a **Lo**RA-based **F**ine-**T**uning framework **LoFT** for SSL to achieve a good trade-off between accuracy and adversarial robustness while keeping low computation costs. First, to avoid huge computation costs and performance gaps, LoFT performs adversarial training in the fine-tuning stage rather than in the pre-training stage. Second, to achieve a reasonable trade-off between accuracy and robustness, LoFT innovatively and elaborately integrates Low-Rank Adaptation (LoRA) [27] into adversarial training to avoid the catastrophic forgetting problem. Specifically, LoFT adds LoRA as a new layer on the pre-trained model. Instead of updating all parameters of the pre-trained model, LoFT retains all parameters of the pre-trained model to maximize its accuracy and only updates the parameters of LoRA during adversarial training. Third, LoFT exploits a low-rank matrix in LoRA which enables efficient fine-tuning by updating only a small set of parameters. Moreover, this contributes largely to further reducing the computation cost as the low-rank matrix is only 1%-5% of the pre-trained model in LoFT.

Extensive experiments demonstrate the outstanding performance of LoFT on accuracy, robustness and computation cost compared to current SOTA methods. Specifically, LoFT achieved an accuracy of 58.04% under AutoAttack and 83.38% on Standard test Accuracy. Compared with existing SOTA robust pre-training methods for SSL, LoFT improves accuracy by 5.97% (77.41%⇒83.38%) and robustness by 13% (45.04%⇒58.04%) on the CIFAR-10 dataset. Notably, the whole pre-training and fine-tuning stages of LoFT only require 9.44 hours. It can efficiently fine-tune the pre-trained models with a 3× time reduction (29.40h⇒9.44h) compared to current SOTA methods. Furthermore, LoFT can improve the robustness of the pre-trained model with just 3% of CIFAR-10 labeled data. Therefore, LoFT can efficiently fine-tune the pre-trained model , and ensuring model accuracy and improving model robustness in downstream tasks.

II. Related work

In this section, we first introduce the technology SSL and its corresponding adversarial attack. Then, we demonstrate some typical research of adversarial training for pre-training and fine-tuning respectively.

A. Self-Supervised Learning and Adversarial Attack

Self-Supervised Learning. Pre-training establishes the basic understanding and generation abilities capabilities for the pre-trained models. In this stage, the amount of pre-training data is critical [29]. To achieve powerful basic capabilities, SSL models generate representations that can be efficiently transferred to downstream tasks through pre-training on a large amount of unlabeled data [30]. As a very important part of SSL, contrastive learning can efficiently extract deeper features of unlabeled data based on the similarities and differences between these data [31]. More specifically, it uses contrast loss to maximize the similarity between *positive samples*, which are different augmented versions of the same data sample [32], [33], while minimizing the similarity between *negative samples* [34], [35]. For classification tasks, the negative samples are essentially different categories with different instances in them e.g., cat and dog.

Adversarial Attack. Adversarial attacks generate adversarial samples by adding small but intended perturbations to natural samples. Szegedy et al. [36] first proposed adversarial attacks against deep neural networks, which make the model make incorrect predictions through subtle perturbations [9]. After this, numerous studies have been devoted to generating, explaining and preventing adversarial attacks against deep neural networks [9]–[11]. Currently, the methods for adversarial attacks include the Fast Gradient Sign Method (FGSM) [37], Projected Gradient Descent (PGD) [38], and AutoAttack [11] and so on. These methods can also be used to assess the adversarial robustness of the target model. *As a typical deep neural network, although SSL has achieved great successes in a variety of fields, they are also vulnerable to adversarial attacks and have serious security issues when applying safety-critical downstream tasks [8]–[10].*

B. Adversarial Training in Stages of Pre-training and Fine-tuning

Adversarial training is a well-studied technology to effectively improve the robustness performance against adversarial attacks [20], [21]. Recent researches demonstrate that introducing adversarial training in the pre-training stage or fine-tuning stage can improve the model's adversarial robustness in SSL-based pre-trained models [15].

Adversarial Training in Pre-training Stage. In the pre-training stage, several recent studies [14], [17], [19], [39] introduce adversarial training in self-supervised learning (SSL)

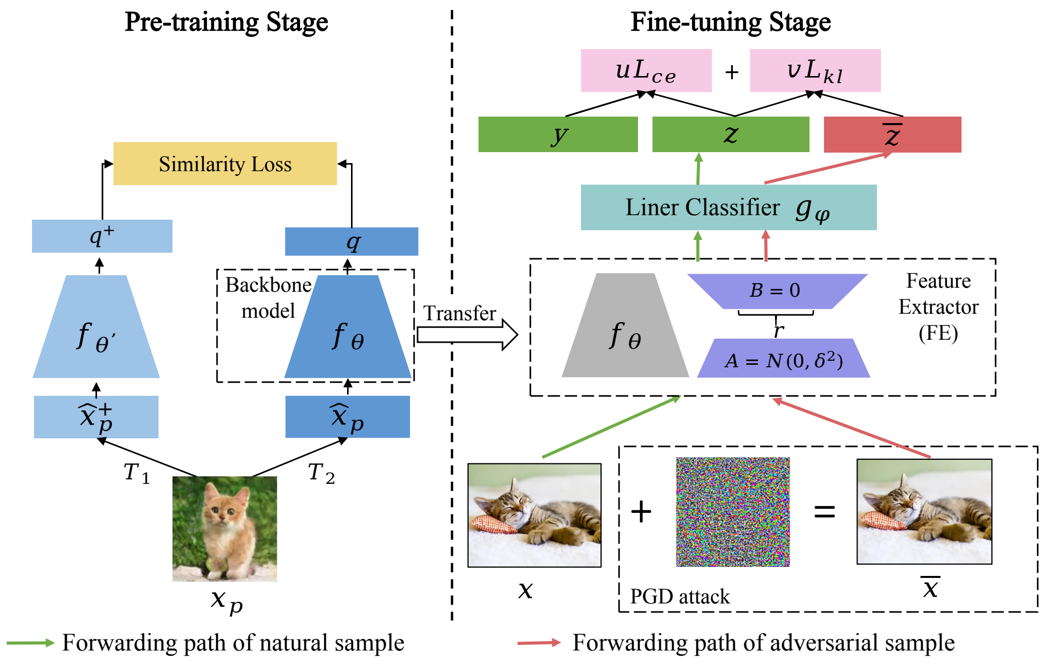

Fig. 2. The LoFT design process.

frameworks to enhance the robustness of pre-trained models and achieve inspiring successes. ACL [19] is the first to introduce adversarial attacks into a comparative learning framework and can lead to models that are both accurate and robust. Nevertheless, both strong and weak data augmentation are harmful to self-supervised adversarial training. DYNACL [14] proposes a dynamic augmentation scheme that gradually anneals from strong to weak augmentation to reduce the impact of data augmentation.

In addition, some researchers have focused on better generation of adversarial samples from unlabeled data of SSL. RoCL [39] maximizes the similarity between random augmentation of data samples and its instance-adversarial perturbations to generate effective adversarial samples for the adversarial SSL framework. BYORL [18] proposes a robust BYOL-based self-supervised learning technique that can train robust pre-trained models without any labelling. AdvCL [17] combines high-frequency components of images and pseudo-labelling mechanisms to enhance model robustness, which can enhance cross-task robustness transfer ability without loss of model accuracy and fine-tuning efficiency. *As pre-training of SSL and adversarial training are two complex computational processes, these methods combine the two processes further increasing the computational burden of pre-training [15], [22]. Besides, these methods are limited in their ability to enhance the robustness and may degrade the accuracy of the model in downstream tasks [19], [23].*

Adversarial Training in Fine-tuning Stage. Moreover, robust fine-tuning introduces adversarial training in the fine-tuning stage to obtain adversarial robust models in downstream tasks [26], [40], [41]. To further improve the generalization ability and robustness for the downstream task, some studies propose to learn the features of both natural and adversarial data together to improve the robustness of the model while ensuring the accuracy of the model [40]. To achieve the optimal balance between robustness and accuracy, Trade-off [42] is proposed to dynamically adjust the weight relationship between robustness and accuracy. In addition, RiFT [43] proposes a non-robust key module, which utilizes redundant capabilities to achieve robustness and enhances the model generalization ability without affecting the adversarial robustness. Currently, TWINS [24] proposes a dual neural network robustness fine-tuning architecture to ensure the generalization ability and improve the robustness of the model by optimizing two neural networks to obtain SOTA performance. *These methods require massive computational resources for gradient computation and fine-tuning of all the parameters of the model [27], [28]. Besides, updating all parameters may make the catastrophic forgetting problem more serious with a clear attack target based on labeled downstream data [23].*

III. Methodology

A. Overview

As shown in Fig. 2, in the pre-training stage, we use a self-supervised learning framework to pre-train the backbone model. Then, we transfer the pre-trained backbone model to the robust fine-tuning stage. In the fine-tuning stage, we integrate a Feature Extractor (FE) with the pre-trained backbone model and LoRA. Then, we add a linear classification layer to the FE and perform adversarial training on the model. By freezing the pre-trained model weights and robustly fine-tuning the LoRA layer, this model can learn the adversarial robustness and maintain accuracy.

Pre-training stage. The key to the pre-training stage is to attain powerful capabilities from as large as possible pre-training data by SSL [29]. The scale of these data is often very large. At the same time, they are generally unlabeled considering with the high cost of data labeling. We define the massive unlabelled dataset as $\mathcal{D}_p = \{x_p | x_p \in \mathbb{R}^n\}$, where x_p is the sample in $\mathcal{D}_p$. A typical SSL framework in the pre-training stage is shown in Fig. 2. Firstly, we construct a positive sample pair $(\widehat{x}_p, \widehat{x}_p^+)$ through different data augmentation ways T_1, T_2 of x_p. Then, the positive sample pairs are passed through a pair of twin backbone models $(f_\theta, f_{\theta'})$, which extract deep features (q, q^+) of the input x_p, expressed as $q = f_\theta(\widehat{x}_p)$. θ represents the parameters of the backbone model. $\theta^{'}$ is the twin backbone network parameter and in some contrastive learning methods $\theta^{'} = \theta$ [32]. We calculate the cosine similarity of the x representations (q, q^+) as a contrast loss and update the backbone model parameters θ. We pre-train the backbone model f_θ with a self-supervised learning (SSL) framework and then transfer it to the fine-tuning stage. After the pre-training of SSL, the pre-trained backbone model f_θ can learn the intrinsic features and structure of the data and efficiently transfer it to downstream tasks.

Fine-tuning stage. To improve the robustness and accuracy of pre-trained backbone models for SSL, LoFT introduces adversarial attacks in the fine-tuning stage. Specifically, we define a dataset $\mathcal{D} = \{(x, y) | x \in \mathbb{R}^n, y \in [K]\}$, where y is the label of sample x from a certain dataset $\mathcal{D}$ and K is the number of categories of all the samples in $\mathcal{D}$. First, we add the low-rank matrix layer (B, A) to the pre-trained backbone model f_θ, detailed in subsection III-B, as a Feature Extractor (FE) for our model $f_{\theta+BA}$. The feature extractor adds a linear layer g_φ that overall serves as a classification model for downstream tasks. Then, for a pair of inputs (x, y), an adversarial perturbation is generated by the PGD-attack [38], and the perturbation is added to x to obtain the adversarial sample $\overline{x}$. Finally, the forwarding path for natural sample x is shown as the green line, the natural sample x input to the feature extractor $f_{\theta+BA}$ and linear layer g_φ to get the prediction result z, expressed as $z = g_\varphi(f_{\theta+BA}(x))$. Similarly, the forwarding path for the adversarial sample is shown in red line. The adversarial sample $\overline{x}$ input to the feature extractor $f_{\theta+BA}$ and linear layer g_φ to get the prediction result $\overline{z}$, expressed as $\overline{z} = g_\varphi(f_{\theta+BA}(\overline{x}))$. Based on the prediction results of the natural samples z, the prediction results of the adversarial samples $\overline{z}$ and the true label y, we introduce trade-loss [42], as in Equation 1, which dynamically adjusts the scaling parameters to find the optimal balance between accuracy and robustness.

$$L_{LoRA} = u * L_{ce}(z, y) + v * L_{kl}(z, \overline{z}) \tag{1}$$

where L_{ce} is for loss of cross-entropy, L_{kl} is denoted KL divergence, and u, v represent scale parameters. The pre-trained model weights θ are frozen and robustly finetuned to the low-rank matrix of LoRA (B, A). We update the parameters of (B, A) by back-propagation of the gradient. After optimized by adversarial training, LoFT parameter (B, A) can learn the robustness to deal with adversarial attacks.

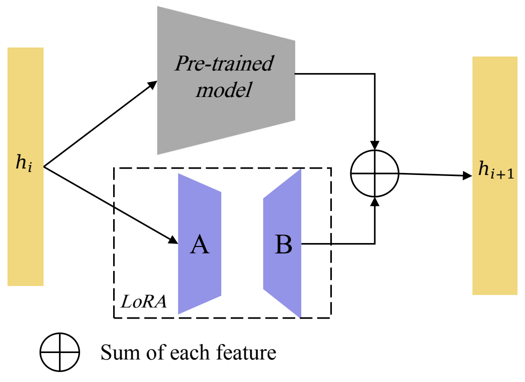

Fig. 3. LoRA-integrated pre-trained model.

The LoRA-integrated pre-training model can improve the model robustness through the LoRA layer and ensure the model accuracy through the pre-trained model. Therefore, after fine-tuning, the LoRA-integrated pre-trained model can efficiently improve the model robustness while ensuring the model's high accuracy and low cost.

B. LoRA-integrated pre-trained model

In this subsection, we describe how LoFT integrates LoRA into the pre-trained backbone model.

Unlike traditional full-parameter fine-tuning methods, LoFT focuses on keeping the weights of the pre-trained model unchanged while integrating new low-order weight matrices into the pre-trained model for adaptive tuning. Specifically, LoFT integrates the LoRA layer in the pre-trained model, and this layer can be viewed as the multiplication of two low-rank matrices BA. The parameters of the low-rank matrix are only 1%-5% of the pre-trained model. For a pre-trained model f_θ, the forward propagation result is $h = f_\theta(x)$. After integrating LoRA, the forward propagation result of the model is $h = f_\theta(x) + BAx$, where BA are two low-rank matrices. $\theta \in \mathbb{R}^{n\times d}$, where n is the dimension of the input x, and d is the dimension of the hidden features of FE. $A \in \mathbb{R}^{n\times r}$ and $B \in \mathbb{R}^{r\times d}$, where r is rank of the matrix. Therefore, the dimensions of $BA \in \mathbb{R}^{n\times d}$ are the same as the FE. The rank r of matrix BA is generally set to 2, 4, and 8.

More specifically, LoFT adds LoRA parameters to the hidden layer of the pre-trained backbone model f_θ for robust fine-tuning of the model, such as the convolution layer and the fully-connected layer. As shown in Fig.3, for the i-th hidden layer of the pre-trained backbone model $H_i \in f_\theta$, we add a pair of parameters (B_i, A_i) to the hidden layer. (B_i, A_i) is two low-rank matrices. The relationship between input h_i and output h_{i+1} is shown in Equation 2.

$$h_{i+1} = H_i(h_i) + B_i A_i(h_i) \tag{2}$$

The product of the low-rank matrices $B_i A_i$ is utilized to represent the weight matrix of the model. Through such low-rank updates, only the parameters of the LoRA part of the model are updated during fine-tuning, and the parameters of the pre-trained model do not need to be updated. We denote the parameters of the lower matrix of all hidden layers, called (B, A). Matrix A is usually initialized as a Gaussian noise matrix $A = N(0, \delta)$, and matrix B is usually initialized as a zero matrix $B = 0$. This initialization method helps to avoid large perturbations to the pre-training parameters in the early stages of fine-tuning while providing space for gradient updating for the fine-tuning process. The optimization target for the (B, A) matrix is shown in Equation 3.

$$\min_{B,A} L(\mathcal{D}, \theta, B, A) = \mathbb{E}_{(x,y)\in\mathcal{D}}[u * L_{ce}(g_\varphi(f_{\theta+BA}(x)), y) + v * L_{kl}(g_\varphi(f_{\theta+BA}(x)), g_\varphi(f_{\theta+BA}(\overline{x}))))] \tag{3}$$

Through this low-rank updating, LoFT significantly reduces the number of parameters to be optimized during the fine-tuning stage, reducing the demand for computational resources. Moreover, all the new information learned in the fine-tuning stage is included in the LoRA parameters, allowing the knowledge of the fine-tuned model to be easily accrued and combined into the pre-trained model via the LoRA parameters.

C. Robust fine-tuning of the optimization process

In this subsection, we detail the process of robust fine-tuning of the self-supervised learning model. Based on the description in the above subsection, we demonstrate our proposed algorithm for LoFT in Algorithm 1.

Algorithm 1 Self-supervised learning for robust fine-tuning via LoRA

```
Input: Data-set $\mathcal{D}$, pre-trained model $\theta$, raining epoch $E$
Output: robust model $\theta_R$
1:  Initialize LoRA parameters $A = N(0, \delta)$, $B = 0$, and linear classification layer $g_\varphi$.
2:  Initialize learning rate $\eta$ = 0.01, batch size $\zeta$ = 256
3:  for epoch e = 1,...,$E$ do
4:      for batch b = 1,...,$|\mathcal{D}|/\zeta$ do
5:          Extract a sample $(x, y)$ from $\mathcal{D}_b$
6:          Generate Adversarial Sample $(\overline{x}, y)$
7:          $z = g_\varphi(f_{\theta+BA}(x))$
8:          $\overline{z} = g_\varphi(f_{\theta+BA}(\overline{x}))$
9:          Calculation of loss $L^*$ under equation 1
10:         Update parameters $A \leftarrow A - \eta \cdot \nabla_A L^*$
11:         Update parameters $B \leftarrow B - \eta \cdot \nabla_B L^*$
12:         Update parameters $\varphi \leftarrow \varphi - \eta \cdot \nabla_\varphi L^*$
13:         $\theta_R = \theta + BA$
14:     end for
15: end for
```

First, the input f_θ is the pre-trained model obtained by training with the contrast learning framework. LoFT initialize the lower rank matrics parameters $A = N(0, \delta)$ and $B = 0$, a linear classification layer g_φ, and set the learning rate and batch size (line 1-2). Second, for each batch, it extracts data samples and generates adversarial samples through PGD-Attack (line 3-6). Then, the algorithm uses the LoRA integrated pre-training model $f_{\theta+BA}$, to compute the forward process. The outputs of the natural sample x and the adversarial sample $\overline{x}$ are z and $\overline{z}$, respectively (line 7-8). Finally, the loss L^* is computed according to equation 1 and back propagates the gradient to update the LoRA parameters A and B and the linear classification layer parameters φ (line 9-12). Through the integration of LoRA tuning BA onto the original pre-trained model parameters θ (line 13), the robust model θ_R is obtained.

IV. Experiments

In this section, we perform extensive experiments on the proposed LoFT. First, we introduced the setting of our experiment, including data, baseline, evaluation metrics, and so on. Then, based on these settings, we conducted the following experiments:

- Assessment of the LoFT model validity. Whether the robustness of the model can be improved while ensuring the accuracy of the model.
- Comparison of the LoFT model accuracy and robustness. Whether there is a performance improvement in comparing baseline.
- Comparison of the LoFT model training time. Comparison of robust self-supervised learning models, whether the training time is reduced or not.
- Evaluate the label efficiency of the LoFT model. Compare the performance of the model on different proportions of the train data set.
- Ablation experiments. We perform ablation studies on the rank r of the LoRA matrix A, B.

A. Experiment Setting

Study Data. We considered two datasets as the downstream tasks, CIFAR-10 and CIFAR-100. CIFAR-10 contains 60,000 colour images of 32x32 pixels divided into 10 categories. The dataset is divided into 50,000 training images and 10,000 test images. CIFAR-100 also contains 60,000 images, which are divided into 100 categories. Like CIFAR-10, CIFAR-100 is also divided into 50,000 training images and 10,000 test images.

Feature Extractor. In our experiments, we performed pre-training based on the BYOL [34] contrastive learning framework, choosing the ResNet-18 [44] network as the feature extractor. Specifically, we used the pre-training weights published in the official GitHub repository provided by solo-learn [45].

Baseline. We take the robust fine-tuning (RFT) methods and robust self-supervised learning (RSSL) methods as baseline methods. RFT methods include vanilla RFT [26], TWINS [24] and AutoLoRA [25]; RSSL methods include DYNACL [14], AdvCL [17] and ACL [19].

Evaluation Metrics. In our experiments, the generalization ability of the model in downstream tasks is evaluated by the Standard Test Accuracy (SA), which reflects the classification accuracy of the model on datasets without adversarial perturbation. To comprehensively evaluate the adversarial robustness of the model, we use the widely used AutoAttack [11] (AA)

TABLE I
COMPARISON OF THE VALIDITY OF OUR METHOD

Method	FT	RFT	Vanilla RFT	Our
SA(%)	94.02	84.46	81.51	83.38
AA(%)	6.61	17.16	47.52	**58.04**

FT denotes natural sample fine-tuning.
RFT denotes robust fine-tuning.

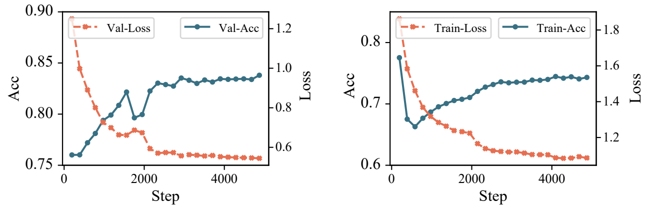

Fig. 4. LoRA-integrated pre-trained model fine-tuning step.

to accurately quantify the robustness of the model in the face of adversarial attacks.

Training Configuration. During training, we use PGD-10 (with an adversarial budget of 8/255 and a step size of 2/255) for 20 iterations to generate adversarial samples. We set the maximum training epoch E = 25. We use SGD as the optimizer, and the rank of the LoRA branch is 8.

B. *Assessment of model validity.*

We performed a LoFT validity assessment on the CIFAR-10 dataset. Fig.4 indicates that the model shows a consistent performance improvement on both the training and validation sets, with increased accuracy and decreased loss. Table I indicates that our method shows excellent robustness under adversarial attacks, significantly outperforming other compared methods with the AA of 58.04%. It indicates that our method can significantly improve the robustness of the model at the cost of little accuracy. Compared with FT, our algorithm reduces the SA metric a little but improves 51.43% (6.61⇒58.04) on the AA metric. This shows that our method can greatly improve the robustness of the model at the cost of lower accuracy.

Overall, our robust fine-tuning of the LoRA-integrated pre-trained model can effectively improve the robustness of the pre-trained model in downstream tasks.

C. *Comparison of model accuracy and robustness.*

We compare the robustness and accuracy of LoFT and baseline on the datasets CIFAR10 and CIFAR100. To fairly compare the performance of LoFT and baseline, we record the experimental results of baseline from the corresponding paper [14], [25]. Table II indicates that our method achieves 83.38% SA and 58.04% AA on CIFAR-10, and 57.38% SA and 34.16% AA on CIFAR-100. These results show that LoFT significantly improves the robustness of the model against adversarial attacks while maintaining reasonable standard accuracy. Compared to RFT, our method shows superior robustness and accuracy on both datasets. This demonstrates

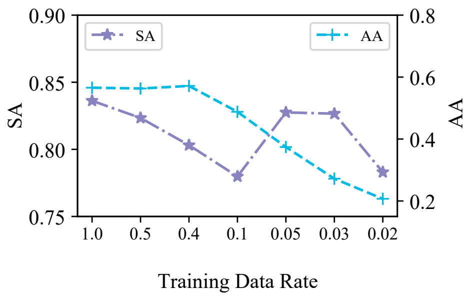

Fig. 5. Different pre-training data ratios fine-tune performance in downstream tasks.

that the robustness improvement of LoFT is consistent across datasets. Moreover, compared with RSSL methods such as DYNACL, AdvCL, and ACL, our method demonstrates stronger performance on AA despite a slight decrease in SA, especially AdvCL and ACL show relatively large improvement on AA. This further validates the effectiveness of the LoFT method in combating robustness.

In summary, LoFT not only demonstrates good standard accuracy on two commonly used datasets. More importantly, it significantly improves the robustness of the model, which is crucial for the defence against adversarial attacks in practical applications.

D. *Comparison of model training time.*

Our experiments are conducted on a server with a 2.1GHz CPU, 32G RAM and a single RTX 3090 GPU. We compare the training time of our method with robust self-supervised learning models such as ACL, AdvCL, and DYNACL. Table III demonstrates the significantly shorter adversarial training time, which can be completed in only 9.44 hours. Our method has a significant advantage in training time, which is much faster than other methods. This means that the robust fine-tuning strategy has a more efficient training strategy in enhancing the robustness of self-supervised learning models in downstream tasks. This is important for rapid deployment of the model in real applications.

Overall, LoFT can efficiently obtain a robust pre-trained feature extractor that can be transferred to downstream tasks. LoFT can efficiently fine-tune self-supervised pre-trained models with a 3× time reduction (29.40h⇒9.44h) compared to current SOTA methods.

E. *Evaluate the label efficiency of the model*

We evaluated the label efficiency of LoFT by using a subset of the original dataset. Specifically, we selectively use different proportions of the full dataset, progressively decreasing from 100% to using only 2% of the dataset. This approach evaluates the performance of LoFT for training models in environments where the number of data labels is restricted. Fig. 5 indicates that as the training data ratio decreases, the SA and AA values also show an overall decreasing trend when the data ratio decreases. The AA decreases sharply when the data

TABLE II
COMPARISON OF ROBUSTNESS AND ACCURACY WITH BASELINES

method		CIFAR10				CIFAR100			
		SA(%)	⇑SA(%)	AA(%)	⇑AA(%)	SA(%)	⇑SA(%)	AA(%)	⇑AA(%)
RFT	TWINS	86.61	-3.23	51.71	**6.33**	66.72	-9.34	28.29	**5.87**
	Vanilla RFT	81.51	**1.87**	47.52	**10.52**	58.55	-1.17	24.07	**10.09**
	AutoLoRa	84.2	-0.82	48.95	**9.09**	62.10	-4.72	27.48	**6.68**
RSSL	DYNACL	77.41	**5.97**	45.04	**13.00**	52.26	**5.12**	20.05	**14.11**
	AdvCL	73.23	**10.15**	37.46	**20.58**	37.58	**19.80**	15.54	**18.62**
	ACL	79.32	**4.06**	37.62	**20.42**	45.34	**12.04**	15.68	**18.48**
	Our	**83.38**	**-**	**58.04**	**-**	**57.38**	**-**	**34.16**	**-**

⇑ represents the performance improvement of our method over the current algorithm.

TABLE III
COMPARISON OF TRAINING TIME PERFORMANCE

Method	ACL	AdvCL	DYNACL	our
Training Time	32.7h	105.0h	29.4h	9.44h

TABLE IV
PROPORTION OF LORA TRAINING PARAMETERS FOR DIFFERENT RANK r.

Rank r	2	4	8	16
Parameter Ratio	0.63 %	1.27 %	2.50 %	4.88 %

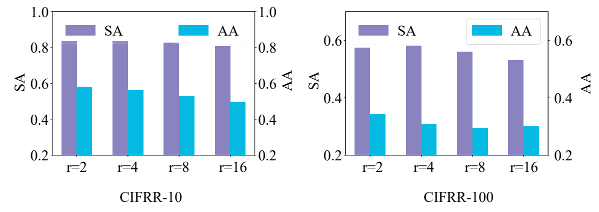

Fig. 6. Ablation study. Performance of LoRA-integrated pre-trained models with different rank r on downstream tasks.

proportion decreases to 0.03 and 0.02, which may be because the model does not have enough data to learn robustness under adversarial attacks.

In summary, LoFT maintains its good performance even at lower data sizes. In particular, LoFT still achieves relatively high SA and AA with limited data volume, which indicates that it has excellent label efficiency. This feature is valuable for real-world applications.

F. Ablation experiments.

In this section, we evaluate the key parameter of LoFT, the rank of the matrix AB, denoted as r. It affects the parameter ratios of the model in the fine-tuning stage. As shown in Table IV, the Parameter Ratio corresponds to expressing different ranks r. The Parameter Ratio represents the ratio of LoRA parameters to the parameters of the pre-trained model.

Fig. 6 shows that on both the CIFAR-10 and CIFAR-100 datasets, both the SA and AA of the model show a trend of increasing and then stabilizing as the LoRA branch rank increases. Both achieve optimum performance when the rank is raised to 8, which may be attributed that more tunable parameters help the model fit the data more accurately and generate high-quality soft labels. However, the performance benefit becomes smaller when the rank exceeds 8, so we choose rank $r = 8$ in the subsequent experiments to achieve a balance of performance and efficiency.

Overall, the rank r of the LoRA matrix has little effect on the performance of LoFT, but it affects the number of parameters for fine-tuning the model.

V. CONCLUSION

Self-supervised learning has achieved remarkable success in learning deep features using unlabeled data. However, we find that SSL models suffer from a lack of robustness in the face of complex adversarial attacks. Addressing this issue motivates us to explore new strategies to improve the robustness of these models in critical downstream tasks. In this paper, we propose a novel robust fine-tuning framework called LoFT. LoFT performs effective robust fine-tuning of pre-trained models by integrating LoRA techniques in the pre-trained models. First, LoFT introduces LoRA layers in each hidden layer of the pre-trained model. Then, LoFT achieves efficient fine-tuning of the model by updating the LoRA parameters. In addition, LoFT requires only a few computational resources to fine-tune the pre-trained model to better cope with adversarial attacks on downstream tasks. Through a series of experiments, we demonstrate that our method outperforms existing robustness enhancement algorithms both in computational efficiency and model robustness and accuracy.

VI. ACKNOWLEDGMENT

This work was supported by the National Key Research and Development Program of China (No. 2021YFB2910104) and the Climbing Program of Institute of Information Engineering, Chinese Academy of Sciences under Grant E3Z0031.

REFERENCES

[1] R. Balestriero, M. Ibrahim, V. Sobal, A. Morcos, S. Shekhar, T. Goldstein, F. Bordes, A. Bardes, G. Mialon, Y. Tian *et al.*, “A cookbook of self-supervised learning,” *arXiv preprint arXiv:2304.12210*, 2023.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[3] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[4] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[5] T. Wu, S. He, J. Liu, S. Sun, K. Liu, Q.-L. Han, and Y. Tang, "A brief overview of chatgpt: The history, status quo and potential future development," *IEEE/CAA Journal of Automatica Sinica*, vol. 10, no. 5, pp. 1122–1136, 2023.

[6] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 11, pp. 4037–4058, 2020.

[7] L. Ericsson, H. Gouk, and T. M. Hospedales, "How well do self-supervised models transfer?" in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5414–5423.

[8] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *Ieee Access*, vol. 6, pp. 14 410–14 430, 2018.

[9] C. Zhang, P. Benz, C. Lin, A. Karjauv, J. Wu, and I. S. Kweon, "A survey on universal adversarial attack," *arXiv preprint arXiv:2103.01498*, 2021.

[10] L. Sun, Y. Dou, C. Yang, K. Zhang, J. Wang, S. Y. Philip, L. He, and B. Li, "Adversarial attack and defense on graph data: A survey," *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[11] F. Croce and M. Hein, "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks," in *International conference on machine learning*. PMLR, 2020, pp. 2206–2216.

[12] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Artificial intelligence safety and security*. Chapman and Hall/CRC, 2018, pp. 99–112.

[13] S. Kaviani, K. J. Han, and I. Sohn, "Adversarial attacks and defenses on ai in medical imaging informatics: A survey," *Expert Systems with Applications*, vol. 198, p. 116815, 2022.

[14] R. Luo, Y. Wang, and Y. Wang, "Rethinking the Effect of Data Augmentation in Adversarial Contrastive Learning," Mar. 2023, arXiv:2303.01289 [cs]. [Online]. Available: http://arxiv.org/abs/2303.01289

[15] C. Zhang, K. Zhang, C. Zhang, A. Niu, J. Feng, C. D. Yoo, and I. S. Kweon, "Decoupled Adversarial Contrastive Learning for Self-supervised Adversarial Robustness," Jul. 2022, arXiv:2207.10899 [cs]. [Online]. Available: http://arxiv.org/abs/2207.10899

[16] M. Kim, H. Ha, S. Son, and S. J. Hwang, "Effective targeted attacks for adversarial self-supervised learning," 2023.

[17] L. Fan, S. Liu, P.-Y. Chen, G. Zhang, and C. Gan, "When does Contrastive Learning Preserve Adversarial Robustness from Pretraining to Finetuning?" in *Advances in Neural Information Processing Systems*, vol. 34. Curran Associates, Inc., 2021, pp. 21 480–21 492.

[18] S. Gowal, P.-S. Huang, A. van den Oord, T. Mann, and P. Kohli, "Self-supervised adversarial robustness for the low-label, high-data regime," in *International Conference on Learning Representations*, 2020.

[19] Z. Jiang, T. Chen, T. Chen, and Z. Wang, "Robust pre-training by adversarial contrastive learning," *Advances in neural information processing systems*, vol. 33, pp. 16 199–16 210, 2020.

[20] A. Shafahi, M. Najibi, M. A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein, "Adversarial training for free!" *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[21] T. Bai, J. Luo, J. Zhao, B. Wen, and Q. Wang, "Recent advances in adversarial training for adversarial robustness," *arXiv preprint arXiv:2102.01356*, 2021.

[22] G. Li and Y. Yu, "Deep contrast learning for salient object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 478–487.

[23] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural networks*, vol. 113, pp. 54–71, 2019.

[24] Z. Liu, Y. Xu, X. Ji, and A. B. Chan, "Twins: A fine-tuning framework for improved transferability of adversarial robustness and generalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 16 436–16 446.

[25] X. Xu, J. Zhang, and M. Kankanhalli, "Autolora: A parameter-free automated robust fine-tuning framework," *arXiv preprint arXiv:2310.01818*, 2023.

[26] D. Hendrycks, K. Lee, and M. Mazeika, "Using pre-training can improve model robustness and uncertainty," in *International conference on machine learning*. PMLR, 2019, pp. 2712–2721.

[27] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-Rank Adaptation of Large Language Models," Oct. 2021, arXiv:2106.09685 [cs]. [Online]. Available: http://arxiv.org/abs/2106.09685

[28] H. Zheng, L. Shen, A. Tang, Y. Luo, H. Hu, B. Du, and D. Tao, "Learn from model beyond fine-tuning: A survey," *arXiv preprint arXiv:2310.08184*, 2023.

[29] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong *et al.*, "A survey of large language models," *arXiv preprint arXiv:2303.18223*, 2023.

[30] K. Sohn, Z. Zhang, C.-L. Li, H. Zhang, C.-Y. Lee, and T. Pfister, "A simple semi-supervised learning framework for object detection," *arXiv preprint arXiv:2005.04757*, 2020.

[31] P. H. Le-Khac, G. Healy, and A. F. Smeaton, "Contrastive representation learning: A framework and review," *Ieee Access*, vol. 8, pp. 193 907–193 934, 2020.

[32] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[33] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9729–9738.

[34] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar *et al.*, "Bootstrap your own latent-a new approach to self-supervised learning," *Advances in neural information processing systems*, vol. 33, pp. 21 271–21 284, 2020.

[35] X. Chen and K. He, "Exploring simple siamese representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 15 750–15 758.

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

[37] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[38] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[39] M. Kim, J. Tack, and S. J. Hwang, "Adversarial self-supervised contrastive learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 2983–2994, 2020.

[40] A. Shafahi, P. Saadatpanah, C. Zhu, A. Ghiasi, C. Studer, D. Jacobs, and T. Goldstein, "Adversarially robust transfer learning," *arXiv preprint arXiv:1905.08232*, 2019.

[41] C. Xie, M. Tan, B. Gong, J. Wang, A. L. Yuille, and Q. V. Le, "Adversarial examples improve image recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 819–828.

[42] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan, "Theoretically principled trade-off between robustness and accuracy," in *International conference on machine learning*. PMLR, 2019, pp. 7472–7482.

[43] K. Zhu, X. Hu, J. Wang, X. Xie, and G. Yang, "Improving generalization of adversarial training via robust critical fine-tuning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4424–4434.

[44] K. Hara, H. Kataoka, and Y. Satoh, "Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?" in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 6546–6555.

[45] V. G. T. da Costa, E. Fini, M. Nabi, N. Sebe, and E. Ricci, "solo-learn: A library of self-supervised methods for visual representation learning," *Journal of Machine Learning Research*, vol. 23, no. 56, pp. 1–6, 2022. [Online]. Available: http://jmlr.org/papers/v23/21-1155.html