IEEE Open Journal of Signal Processing

Received 16 April 2024; revised 16 June 2024; accepted 18 June 2024. Date of publication 26 June 2024;
date of current version 24 July 2024. The review of this article was arranged by Associate Editor Benedetta Tondi.

Digital Object Identifier 10.1109/OJSP.2024.3419569

Efficient Key-Based Adversarial Defense for ImageNet by Using Pre-Trained Models

APRILPYONE MAUNGMAUNG [1] (Member, IEEE), ISAO ECHIZEN [1] (Senior Member, IEEE), AND HITOSHI KIYA [2] (Life Fellow, IEEE)

[1]National Institute of Informatics, Tokyo 101-8430, Japan
[2]Tokyo Metropolitan University, Tokyo 191-0065, Japan

CORRESPONDING AUTHOR: APRILPYONE MAUNGMAUNG (e-mail: pyone@nii.ac.jp).

This work was supported in part by JSPS KAKENHI under Grant JP21H04907, Grant JP21H01327, Grant 23K19983 and Grant JP24H00732, in part by JST CREST under Grant JPMJCR18A6 and Grant JPMJCR20D3 including AIP challenge program, and in part by JST AIP Acceleration under Grant JPMJCR24U3, Japan.

ABSTRACT In this paper, we propose key-based defense model proliferation by leveraging pre-trained models and utilizing recent efficient fine-tuning techniques on ImageNet-1 k classification. First, we stress that deploying key-based models on edge devices is feasible with the latest model deployment advancements, such as Apple CoreML, although the mainstream enterprise edge artificial intelligence (Edge AI) has been focused on the Cloud. Then, we point out that the previous key-based defense on on-device image classification is impractical for two reasons: (1) training many classifiers from scratch is not feasible, and (2) key-based defenses still need to be thoroughly tested on large datasets like ImageNet. To this end, we propose to leverage pre-trained models and utilize efficient fine-tuning techniques to proliferate key-based models even on limited compute resources. Experiments were carried out on the ImageNet-1 k dataset using adaptive and non-adaptive attacks. The results show that our proposed fine-tuned key-based models achieve a superior classification accuracy (more than 10% increase) compared to the previous key-based models on classifying clean and adversarial examples.

INDEX TERMS Adversarial defense, fine-tuning, image classification, image encryption, key-based defense, pre-trained models.

I. INTRODUCTION

Deep learning has brought breakthroughs in many applications [1]. Some notable examples are visual recognition [2], natural language processing [3], and speech recognition [4]. Despite the remarkable performance, machine learning (ML) generally, including deep learning, is vulnerable to various attacks. Notably, many ML algorithms, including deep neural networks, are sensitive to carefully perturbed data points known as adversarial examples intentionally designed to cause ML models make mistakes [5], [6], [7]. In many cases, perturbation added to make adversarial examples is often imperceptible to humans but still causes ML models to make erroneous predictions with high confidence. Previous works have proven that adversarial examples can be applied to real-world scenarios [8], [9]. Adversarial examples can potentially be dangerous, especially for autonomous vehicles [10], [11].

As adversarial examples are an obvious threat, researchers have proposed numerous methods to defend against adversarial examples in the literature [12]. However, most defense methods either reduce the classification accuracy or are completely broken by adaptive attacks [13], [14]. Therefore, defending against adversarial examples is still challenging and remains an open problem. Inspired by cryptography, a new line of research on adversarial defense has focused on using secret keys so that defenders have some information advantage over attackers [15], [16], [17], [18], [19]. Key-based defenses follow Kerckhoffs's second cryptographic principle, which states that a system should not require secrecy even if it is exposed to attackers, but the key should be secret [20]. By keeping a secret key, key-based defenses make adversarial attacks ineffective. The idea of making adversarial attacks expensive or ideally intractable is further supported on a theoretical basis that adversarially robust machine learning

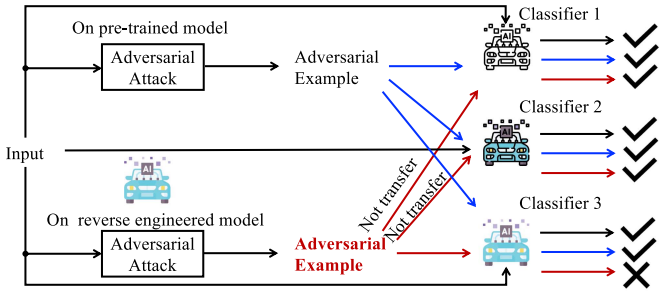


FIGURE 1. Scenario of on-device image classification under adversarial settings.

could leverage computational hardness, as in cryptography [21]. To further harden key-based defenses, researchers have also proposed to use implicit neural representation [22] and ensembles of key-based defenses [23], [24].

One of the advantages of key-based defenses is that a classifier has its own key. This feature is handy as adversarial examples are transferable within models with the same architecture [25] or different ones [26]. Therefore, in this paper, we consider an on-device image classification scenario (Fig. 1) where each classifier has its own key. If an attacker successfully reverse engineers a classifier (e.g., classifier 3 in Fig. 1), adversarial examples generated for the compromised model cannot transfer to other models.

The use case of the one-key-one-model approach is relevant and realistic, especially nowadays, because even large models can be deployed on CPUs [27] and on edge devices [28]. For example, Apple Silicon devices can run a large image generative model like Stable Diffusion [28]. In addition, even large language models (LLMs) can be deployed on edge devices by using C/C++ implementation of models with the ggml[1] library. For image classification, vit.cpp[2] (C++ inference engine for vision transformer models) is available for edge devices. Although deep learning models are widely deployed on edge devices and will be deployed more and more in the near future, the one-key-one-model approach is underexplored.

Therefore, we focus on the one-key-one-model scenario in this paper. First, we point out that key-based defenses seem promising but are not practical for two reasons: (1) training many classifiers from scratch is not feasible, and (2) key-based defenses have not yet been thoroughly tested on large datasets like ImageNet. Then, we build upon the idea of key-based defense and propose to leverage pre-trained models and use the latest fine-tuning techniques to train many defended models efficiently. In experiments, our key-based models are efficiently trained and achieve a higher classification accuracy for both clean and p-norm bounded adversarial examples for ImageNet-1 k classification compared to state-of-the-art methods. However, key-based models have information advantage over attackers for having a secret key. We make the following contributions in this paper.

- We propose to leverage pre-trained models and use the latest fine-tuning techniques to train key-based models efficiently for the first time.
- We conduct experiments for ImageNet-1 k classification and evaluate key-based models using adaptive and non-adaptive attacks.
- We apply attention visualization and plot decision boundaries on key-based defended models to gain insights into the behavior of key-based defense.

The rest of this paper is structured as follows. Section II presents related work on adversarial examples, defenses, pre-trained models, and a recent fine-tuning technique, LoRA. Section III puts forward the proposed defense. Experiments on various attacks including adaptive ones are presented in Sections IV, and V presents discussion on the proposed defense. Then, Section VI concludes this paper.

II. RELATED WORK

A. ADVERSARIAL EXAMPLES

Adversarial examples [5], [6], [7] are intentionally perturbed inputs to machine learning models that cause the model to make erroneous predictions [29]. There are two kinds of adversarial examples based on how they are generated: perturbation-based and unrestricted adversarial examples. Perturbation-based adversarial examples are generally p-norm bounded, and different matrix norms such as ℓ_∞ [30], ℓ_2 [31], ℓ_1 [32], and ℓ_0 [33] are used to restrict the perturbation. Beyond norm-bounded perturbation, adversarial examples can also be found in an unrestricted way [34] by applying spatial transformation [35] or generative models [36].

Adversarial examples can generalize to real-world applications and have the potential to be dangerous. Kurakin et al. showed that an adversarial example can be photographed with a smartphone, and the taken picture can still fool a model [8]. Researchers have also demonstrated that it is possible to construct 3D adversarial objects [9]. The threat of adversarial examples is especially alarming for autonomous vehicles. Attackers could create stickers or paint to design adversarial stop signs to cause accidents [10], [11]. In addition, adversarial examples can be deployed in many different ways to fool facial recognition or object detection systems, such as adversarial t-shirts [37], adversarial hats [38], adversarial glasses [39], adversarial make-up [40], etc. Generating real-world adversarial examples is not limited to small or imperceptible changes to the input. Adversarial examples can also be crafted by placing a small visible image-independent patch [41].

In this paper, we still deploy p-norm bounded adversarial examples since they are well-defined to evaluate the proposed defense. However, we consider a realistic attacking scenario.

B. ADVERSARIAL DEFENSES

There are two distinct strategies in designing adversarial defenses.

1) Classifiers are designed in such a way that they are robust against all adversarial examples in a specific adversarial space

[1][Online]. Available: https://github.com/ggerganov/ggml
[2][Online]. Available: https://github.com/staghado/vit.cpp

either empirically (i.e., adversarial training) or in a certified way (i.e., certified defenses). Current empirically robust classifiers utilize adversarial training, which includes adversarial examples in a training set. Madry et al. approach adversarial training as a robust optimization problem and utilize projected gradient descent (PGD) adversary under ℓ_∞-norm to approximate the worst inputs possible (i.e., adversarial examples) [30]. As PGD is iterative, the cost of computing PGD-based adversarial examples is expensive. Much progress has been made in reducing the computation cost of adversarial training, such as free adversarial training [42], fast adversarial training [43], and single-step adversarial training [44]. However, adversarially trained models (with ℓ_∞ norm-bounded perturbation) can still be attacked by ℓ_1 norm-bounded adversarial examples [45].

Another approach is to use formal verification methods in such a way that no adversarial examples exist within some bounds [46], [47], [48], [49]. Ideally, these defenses are preferred for achieving certain guarantees. Although certified defenses are attractive, they can be bypassed by generative perturbation [50] or parametric perturbation (outside of pixel norm ball) [51].

2) Another primary strategy for designing adversarial defenses is that input data to classifiers are pre-processed in such a way that adversarial examples are ineffective (i.e., input transformation defenses, key-based defenses). The idea is to find a defensive transform to reduce the impact of adversarial noise or make adversarial attacks ineffective (i.e., computing adversarial noise is either expensive or intractable). The works in this direction use various transformation methods, such as thermometer encoding [52], diverse image processing techniques [53], [54], denoising strategies [55], [56], GAN-based transformation [57], and so on. Although these input transformation-based defenses provided high accuracy at first, they can be attacked by adaptive attacks such as [13], [14]. Unlike input transformation-based defenses, key-based defenses have an information advantage over attackers. Such key-based defenses include [15], [16], [22].

Inspired by cryptography, the main idea of the key-based defense is to embed a secret key into the model structure with minimal impact on model performance. Assuming the key stays secret, an attacker will not obtain any useful information on the model, which will render adversarial attacks ineffective. Generally, key-based defenses hide a model's decision from attackers by means of training the model with encrypted images. In this paper, we adopt block-wise pixel shuffling from the key-based defense [16]. The major differences between the proposed defense and the work from [16] are as follows.

- The proposed method focuses on the key-based defense of fine-tuned models from a practical perspective, while the previous work [16] concentrates on training from scratch. Although a simple experiment of fine-tuning a ResNet50 ImageNet model was presented in the previous work, efficient fine-tuning tricks such as [58] and [59] were never considered.
- The proposed method builds on top of large-scale pre-trained models (ImageNet-21 k) that improve the overall performance, whereas the previous work utilizes an ImageNet-1 k pre-trained model on a limited case.
- The proposed method focuses on a realistic threat model where the attacker has imperfect knowledge of the defense model. The attacker carries out a transferred attack, which is more practical. In contrast, the previous work considers full knowledge of the model except for the secret key and a modified version of a black-box model where the attacker must query a defended model with a key, which is not practical in real-world applications.
- The proposed method focuses on larger models such as ViTs, whereas the previous work experiments only on a smaller model such as ResNet50.

C. PRE-TRAINED MODELS

Pre-training is a favorable paradigm for many computer vision tasks because training a large-scale model is a non-trivial task and requires a significant amount of resources. Therefore, it is not feasible for many users to train a large-scale model from scratch. Besides the training cost, pre-training also improves generalization for downstream tasks [60]. Recent works on the pre-training ImageNet-21 k dataset [61] (approximately 14 million images with about 21,000 distinct object categories) show superior performance on ImageNet-1 k classification [58], [62].

Moreover, current frontiers of AI applications such as ChatGPT are driven by pre-trained models (*a.k.a.* foundation models). The term foundation model was coined by researchers from the Stanford Institute for Human-Centered Artificial Intelligence (HAI). A foundation model is a large-scale model trained with a vast amount of data (generally using self-supervised learning), and it can be adapted to a wide range of downstream applications such as generating text and images, understanding natural language, etc. [63].

We are motivated by the success of pre-trained models, and we utilize an ImageNet-21 k pre-trained vision transformer (ViT) [64] in this paper.

D. LORA

LoRA, which stands for low-rank adaptation of large language models is an efficient fine-tuning technique initially introduced to fine-tune large language models [59]. Instead of fine-tuning the whole model, LoRA first freezes all the pretrained model weights and then injects trainable rank decomposition matrices into each layer of the transformer block. Therefore, the number of trainable parameters of a model is significantly reduced. Compared to full fine-tuning, LoRA maintains a competitive performance without increasing any additional inference latency. In this paper, we utilize LoRA to fine-tune a pre-trained ViT model to be a key-based defended ViT model.

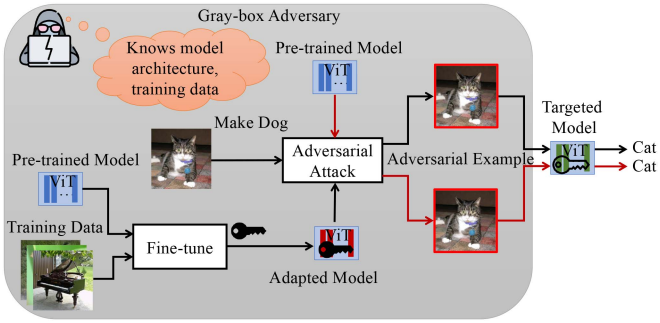


FIGURE 2. Assumed threat model. The adversary carries out attacks on both pre-trained model and adapted fine-tuned model.

III. PROPOSED DEFENSE

A. REQUIREMENTS

We aim to fulfill the following requirements in the proposed defense.

- Leverage pre-trained models and fine-tune them efficiently so that training a key-based model on an ImageNet scale is practical for the majority of users.
- Achieve high classification accuracy for both clean and adversarial examples.
- Associate one key to one model only so that a successful attack on one model does not transfer to another model with another key.

B. THREAT MODEL

A threat model includes a set of assumptions, such as an adversary's goals, knowledge, and capabilities [65].

As we focus on image classifiers, the adversary's goal is to change the predicted class from a true class either in a targeted or untargeted way. We deploy the AutoAttack (AA) strategy, which is a suite of both targeted and untargeted attacks [66] in this paper.

The adversary's knowledge can be classified as white-box, black-box, or gray-box. In white-box settings, the adversary has complete knowledge of the model, its parameters, training data, and the inner workings of the defense mechanism. In contrast, the black-blox adversary has no knowledge about the model. However, in many cases, the adversary knows something in between white-box and black-box, which is referred to as a gray-box scenario. Since the gray-box scenario is more realistic, we consider that the adversary knows the model architecture and has access to pre-trained models and training data. In addition, we assume the gray-box adversary also knows the mechanism of the key-based defense, but not the secret key.

We consider a p-norm bounded threat model for its well-defined nature. Therefore, the adversary can add small perturbation δ under some budget ϵ (i.e., $\|\delta\|_p \leq \epsilon$), where $\epsilon > 0$.

With the above assumptions, Fig. 2 shows the considered gray-box adversary that performs attacks on a pre-trained model or an adapted fine-tuned model with an assumed key to target a defended model.

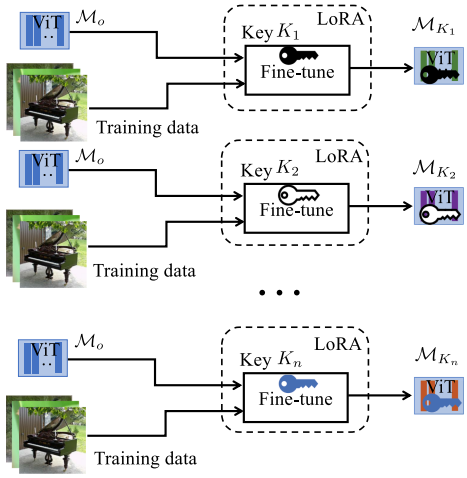


FIGURE 3. Proposed defense. A pre-trained model is fine-tuned with many keys to produce many defended models.

C. OVERVIEW

We consider an on-device image classification scenario. Such a scenario is practical because, nowadays, devices such as autonomous vehicles or smartphones are equipped with image classifiers. The basic idea of the proposed defense is to personalize a pre-trained model with a secret key efficiently so that a classifier has its own decision-making process based on a key. By leveraging the previous works, key-based defense [16], training improvements [58], and efficient fine-tuning [59], we propose to efficiently fine-tune a pre-trained model to many defended models with many different keys, as shown in Fig. 3 in this paper. Consequently, defended models can be potentially deployed on devices such that one defended model is associated with one key only.

D. TRAINING

Given a dataset $\mathcal{D}$ with pairs of examples (images and corresponding labels), $\{(x, y) \mid x \in \mathcal{X},\ y \in \mathcal{Y}\}$, a key-based defense maps the input space $\mathcal{X}$ to an encrypted space $\mathcal{H}$ by using some transformation t controlled by a secret key K (i.e., $x \mapsto t(x, K)$). A classifier $f : \mathcal{H} \to \mathcal{Y}$ is trained by using encrypted images $t(x, K) \in \mathcal{H}$. For simplicity, we define such a defended model with key K as

$$\mathcal{M}_K(\cdot) = f(t(\cdot, K)), \tag{1}$$

where $t(\cdot, K)$ is a key-based transformation and f is a deep neural network-based image classifier such as ViT [64]. Fig. 4 depicts supervised learning with a key-based defense. In this paper, we adopt block-wise pixel shuffling from [16] as $t(\cdot, K)$ and the detailed procedure is as follows.

1) Divide a three-channel (RGB) color image, x with $w \times h$ into non-overlapping blocks, each with $P \times P$ such that $\{B_1, \ldots, B_i, \ldots, B_{(\frac{h}{P} \times \frac{w}{P})}\}$.
2) Generate a random permutation vector, v with key K, such that $(v_1, \ldots, v_k, \ldots, v_{k'}, \ldots, v_{3P^2})$, where $v_k \neq v_{k'}$ if $k \neq k'$.

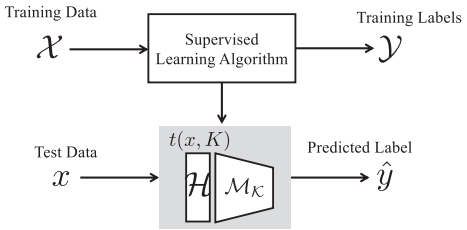
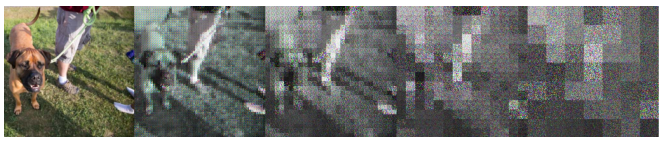


FIGURE 4. Supervised learning with key-based defense.



(a) Plain (b) $P = 4$ (c) $P = 8$ (d) $P = 16$ (e) $P = 32$

FIGURE 5. Example of block-wise transformed images in different block sizes.

3) For each block B_i,
flatten three-channel block of pixels into a vector, b_i such that $b_i = (b_i(1), \ldots, b_i(3P^2))$,
permute pixels in b_i with v such that

$$b_i'(k) = b_i(v_k), k \in \{1, \ldots, 3P^2\}, \tag{2}$$

and reshape the permuted vector b_i' back into the three-channel block B_i'.
4) Integrate all permuted blocks, B_1' to $B'_{(\frac{h}{P} \times \frac{w}{P})}$ to form a three-channel pixel shuffled image, x'.

A key-based defended model $\mathcal{M}_K$ is obtained by fine-tuning a pre-trained model $\mathcal{M}_o$ (which is trained by using plain images) with transformed images by key K. Examples of transformed images with different block size P are shown in Fig. 5. By leveraging efficient fine-tuning techniques such as LoRA [59], many defended models $(\mathcal{M}_{K_1}, \mathcal{M}_{K_2}, \ldots, \mathcal{M}_{K_n})$ can be efficiently fine-tuned given keys $(K_1, K_2, \ldots, K_n)$ as shown in Fig. 3. We are the first to consider LoRA in the key-based defense.

The previous work [67] showed that a key-based defended model can be obtained by fine-tuning only the patch embedding layer and the classifier head of an isotropic convolutional neural network. However, the performance accuracy dropped. We further extend the previous work [67] by applying LoRA to both the patch embedding layer and the qkv layer in each transformer block to improve the performance accuracy in this paper.

E. INFERENCE

During inference, a defended model $\mathcal{M}_K$ first transforms test images with key K prior to image classification procedure by a deep neural network backbone such as ViT. A predicted class label $\hat{y}$ is obtained from a defended classifier $\mathcal{M}_K$ as

$$\hat{y} = \arg\max_i \mathcal{M}_K(x)_i = \arg\max_i f(t(x, K))_i. \tag{3}$$

F. EVALUATION METRIC

To evaluate the proposed defense, we calculate accuracy for classifying both clean and adversarial examples. The accuracy is computed as

$$\text{Accuracy} = \begin{cases} \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(\mathcal{M}_K(x_i) = y_i) & \text{(clean)} \\ \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(\mathcal{M}_K(x_i + \delta i) = y_i) & \text{(attacked)}, \end{cases} \tag{4}$$

where $\mathcal{M}_K$ is a defended classifier with key K, N is the number of test images, $\mathbb{1}$(condition) is one if condition is true, otherwise zero, $\{x_i, y_i\}$ is a test image (x_i) with its corresponding label (y_i), and δ_i is its respective adversarial noise depending on a specific attack.

IV. EXPERIMENTS

A. SETUP

Datasets: We carried out ImageNet-1 k classification experiments for the proposed defense. We utilized the ImageNet-1 k dataset (with 1000 classes) consisting of 1.28 million color images for training and 50,000 color images for validation [68]. ImageNet-1 k was introduced for the ILSVRC2012 visual recognition challenge and is regarded as one of the main datasets for image classification research. It is a subset of the ImageNet-21 k dataset, which consists of approximately 14 million images with about 21,000 classes [61]. We resized all images to a dimension of 224×224. In addition, we used the CIFAR-10 [69] dataset for analysis. CIFAR-10 consists of 60,000 color images (dimension of $32 \times 32 \times 3$) with 10 classes (6000 images for each class) where 50,000 images are for training and 10,000 for testing.

Models: We utilized pre-trained ViT base models with a patch size value of 16 from [58] and [64] that were trained on ImageNet-21 k. To implement the proposed defense, we modified pytorch image models,[3] and LoRA fine-tuning.[4] We followed the training settings from [58] to fine-tune the proposed defended models. For LoRA, we used a low-rank dimension value $r = 16$ and a value of scaling factor 16. In addition, we used ResNet18 for analysis with CIFAR-10. We adopted training settings from the publicly available repository[5] to train ResNet18 from scratch.

Attacks: For evaluation, we deployed the AutoAttack (AA) strategy, which is an ensemble of strong, diverse attacks consisting of both targeted and untargeted attacks [66]. Table 1 summarizes attacks that are deployed in AA. We used AA's "standard" version with a perturbation budget value of $4/225$ under ℓ_∞ norm and a value of 0.5 under ℓ_2 norm for all attacks. Moreover, we also used projected gradient descent with 20 steps (PGD-20) [30] under ℓ_∞ for analysis and experiment purposes.

[3][Online]. Available: https://github.com/huggingface/pytorch-image-models
[4][Online]. Available: https://github.com/huggingface/peft
[5][Online]. Available: https://github.com/kuangliu/pytorch-cifar

TABLE 1. List of Attacks Deployed in AutoAttack

Attack	Knowledge	Goal
APGD-CE	white-box	untargeted
APGD-DLR	white-box	targeted
FAB	white-box	targeted
Square Attack	black-box	untargeted

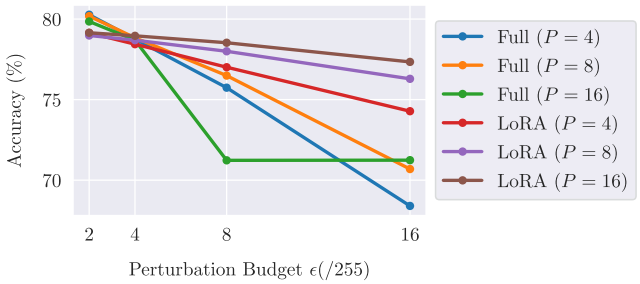


FIGURE 6. Accuracy (%) of key-based fine-tuned models under AA (ℓ_∞) adversary with various perturbation budget ϵ. The accuracy was calculated over 50,000 images (whole ImageNet-1 k validation set).

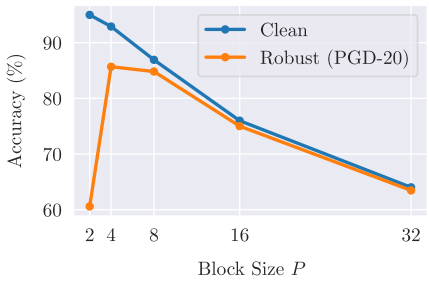


FIGURE 7. Accuracy (%) vs. block size *P*. Adversarial examples were generated by using plain model with PGD-20 (ℓ_∞, $\epsilon = 8/255$) for CIFAR-10 (test set).

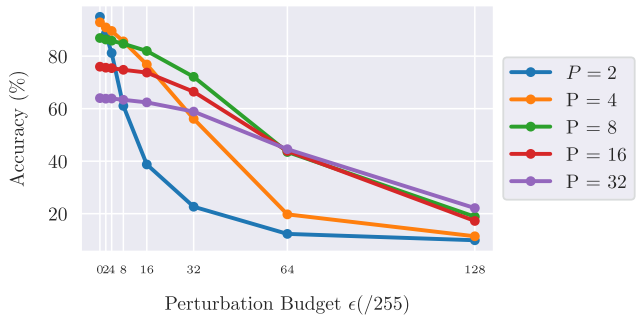


FIGURE 8. Robust accuracy (%) vs. perturbation budget ϵ. Adversarial examples were generated by using plain model with PGD-20 (ℓ_∞) for CIFAR-10 (test set).

B. CLASSIFICATION ACCURACY

We fine-tuned pre-trained ViTs [64]: ViT-B/16 (ViT base model with a patch size of 16), ViT-B/32 (ViT base model with a patch size of 32), and ConvNeXt-tiny [70] (tiny version) with key-based transformation under two fine-tuning strategies: Full and LoRA.

Full: We fine-tuned the whole network (all learnable parameters) with training parameters described in [58] for 50 epochs for ViTs and 75 epochs for ConvNeXt [70]. We used all default parameters for all models.

LoRA: In this strategy, we applied LoRA on the patch embedding and qkv layers of all transformer blocks in ViTs.

We performed ImageNet-1 k classification on both clean images and adversarial examples. Table 3 summarizes results of both clean and robust accuracy compared to state-of-the-art key-based methods [16], [67] and adversarial training (AT) methods [71], [72]. In the table, (–) denotes "there are no reported results" or "not applicable," and both clean and robust accuracies were calculated on the validation set of ImageNet-1 k (50,000) images as described in Section III-F. Although the un-defended model (plain) achieved the highest accuracy, it was most vulnerable to all attacks. In terms of clean accuracy, our best model (fully fine-tuned) with block size ($P = 4$) achieved 83.40%, which is about 1.7%-pts (percentage points) lower compared to the plain model.

Non-Adaptive Attacks: Since the proposed defense is built on top of the pre-trained model, it is natural to generate adversarial examples directly on the pre-trained model. We refer to this scenario as a non-adaptive attack. The results from Table 3 show that adversarial examples generated on the pre-trained model were not effective on either fully fine-tuned or LoRA models. In addition, we also carried out non-adaptive attacks with ℓ_∞ adversary with different perturbation budget ϵ. As shown in Fig. 6, as we increased ϵ, the accuracy gradually dropped as expected.

In addition, we carried out additional experiments on the CIFAR-10 datset. We observed block size P versus accuracy, as shown in Fig. 7. As the block size is increased, the accuracy decreases. A large block size causes higher visual distortion (see Fig. 5). Therefore, the model loses plain visual features, and the classification accuracy is dropped. Moreover, we also observed robust accuracy when the perturbation budget is increased in the CIFAR-10 dataset for defended models with different block sizes in Fig. 8. Like ImageNet, the robust accuracy on CIFAR-10 also dropped as the noise budget increased. However, the robust accuracy did not go to zero even on a extremely large perturbation budget $\epsilon = 128/255$.

Adaptive Attacks: We assume the gray-box adversary as described in Section III-B. Thus, the adversary has knowledge of the key-based defense mechanism except the key. The attacker may fine-tune the pre-trained model with a guessed key and prepare a similar substitute model as the targeted key-based model. By using the substitute key-based model, the adversary may generate adversarial examples. We refer to this scenario as an adaptive attack. From the results (Table 3), our fully fine-tuned models were robust against such attacks. Interestingly, adaptive attacks were even worse than non-adaptive attacks on fully fine-tuned models. However,

TABLE 2. Robust Accuracy (%) of EoT Attack

Block Size	Robust (EoT ℓ_∞ PGD-20, $\epsilon = 8/255$)
$P = 2$	**92.95**
$P = 4$	53.02
$P = 8$	86.62
$P = 16$	75.76

EoT was performed over 10 models with $P = 4$ and different keys for CIFAR-10.

adaptive attacks were successful on LoRA fine-tuned models, especially for the models with $P = 8$ and 16. We suspect that the LoRA applied patch embedding layer in ViT-B/16 does not fully capture the key-based transformation. We shall further investigate the reason and improve the robustness of LoRA fine-tuned models in our future work.

We consider multiple surrogate models to increase the chance of attack success further. We assume the attacker does not know the correct secret key. However, the attacker can train multiple surrogate models with different keys to get a better approximation of the correct key. In this experiment, we utilized the CIFAR-10 dataset for training efficiency. We trained 10 models with $P = 4$ for 10 random keys. Similar to estimation over transformation (EoT) [13], we consider the sum of gradients for 10 models (i.e., $\sum_{i=1}^{10} \nabla_x \mathcal{M}_{K_i}(x)$).

Table 2 shows robust accuracy of different models with different block size P against EoT attack. Notably, when the block size P is the same for surrogate and targeted models, the accuracy was reduced to $\approx 53.02\%$. This shows that the adaptive attack is effective if the approximation of the correct key with multiple keys is effective. However, training many models is expensive. The random permutation vector v generated by key K should be fully permuted (based on a random orthogonal matrix) to harden the defense model. When v is not fully permuted, adaptive attacks such as EoT will be more successful.

Comparison with State-of-the-Art Methods: We compare the proposed fine-tuned models with the previous key-based models [16], [67], and the top 3 adversarially trained (AT) models from RobustBench [73] in terms of clean and robust accuracy. Note that it is not fair to compare key-based models directly with AT models because key-based models have an information advantage (secret key) over attackers, while AT models do not. The previous key-based methods utilized different model architectures, ConvMixer-768/32 and ResNet50, which are much smaller compared to ViT-B/16. The ConvMixer one used partial fine-tuning (patch embeddings and classifier head), and the ResNet50 one employed full-finetuning. In contrast, we utilized a larger model, ViT-B/16, with the latest fine-tuning techniques for efficient training in this paper. Our models achieved a superior performance (more than a 10% increase) on both clean and robust accuracy compared to the previous key-based models and AT models.

C. TRAINING EFFICIENCY

Training a model on the ImageNet-1 k dataset from scratch requires a lot of resources and is not feasible for normal users with limited resources. However, by using the latest fine-tuning techniques like [58], [59], one can train a key-based defended model in less than a day. Table 4 shows the trainable parameters of a ViT-B/16 model on full fine-tuning and LoRA fine-tuning. LoRA is no doubt efficient in fine-tuning because it updates only a fraction of total trainable parameters. In our experiments, we applied LoRA only on the patch embedding layer and qkv layers of all transformer blocks in the ViT-B/16. We utilized 4 NVIDIA A100 GPU cards on the full fine-tuning and one A100 GPU card for LoRA fine-tuning. Although LoRA is significantly efficient, adaptive attacks defeated LoRA fine-tuned models in our experiments. We shall improve LoRA fine-tuned models to be more robust against adversarial examples in our future work.

V. ANALYSIS AND DISCUSSION

A. ANALYSIS

Block Size: The block size P directly affects key space and performance (clean and robust accuracy).

In theory, P controls the key space. For the transformation used in this paper (block-wise pixel shuffling), the key space is $(P \times P \times 3)!$. However, not all key keys are good for the key-based defense. For example, a key that permutes only 50% or less of pixels in a block will be vulnerable to attacks. Therefore, we encourage using a key that generates a full permutation vector (based on a random orthogonal matrix). The number of available keys for the key-based defense varies based on P.

The main essence of a key-based defense is to embed secret grid patterns on input images so that when training a model with such images, the model learns secret features. When using a small block size, transformed images still have a clear content outline, and the model learns much of plain features, thus having a high classification accuracy. In contrast, transformation with a large block size distorts the content of the image, and the models lose much of the plain features from plain images. Therefore, the model heavily relies on secret features and achieves a higher robust classification accuracy at the expense of clean classification accuracy. Consequently, there is a trade-off between secret and plain features for robust and non-robust accuracy. In our future work, we shall further investigate how to balance secret and plain features.

Key Diversity: To evaluate the performance of different defended models with different keys, we trained 10 ResNet18 models with 10 keys for the CIFAR-10 dataset $(\mathcal{M}_{K_1}, \ldots, \mathcal{M}_{K_{10}})$. All defended models utilized a block size value, $P = 4$. Then, we tested each model with 10 keys on the CIFAR-10 test set (10,000 images). Table 5 summarizes the clean classification accuracy of each model for test images transformed by 10 keys. When using the correct keys, models achieved a high, consistent accuracy. In contrast, accuracy significantly dropped when using incorrect keys. Moreover,

TABLE 3. Clean and Robust Accuracy (%) of Proposed Key-Based Models and State-of-the-Art Models

Model	Defense	Block Size (P)	Fine-tune	Clean	Robust ℓ_∞ ($\epsilon = 4/255$) (Non-Adaptive)	Robust ℓ_∞ ($\epsilon = 4/255$) (Adaptive)	Robust ℓ_2 ($\epsilon = 0.5$) (Non-Adaptive)	Robust ℓ_2 ($\epsilon = 0.5$) (Adaptive)
ViT-B/16	Plain	–	No	85.10	0.00	–	0.32	–
ViT-B/16 (Ours)	Key	4	Full	**83.40**	78.76	**81.45**	**81.89**	**82.17**
ViT-B/16 (Ours)	Key	8	Full	82.62	78.84	80.68	81.60	81.46
ViT-B/16 (Ours)	Key	16	Full	82.02	78.69	79.58	81.10	80.67
ViT-B/32	Plain	–	No	80.72	0.00	–	11.95	–
ViT-B/32 (Ours)	Key	4	Full	76.79	72.58	74.27	77.14	76.75
ConvNeXt-tiny	Plain	–	No	84.18	0.00	–	0.24	–
ConvNeXt-tiny (Ours)	Key	4	Full	81.68	77.93	78.69	81.10	80.75
ViT-B/16 (Ours)	Key	4	LoRA	81.12	78.44	51.56	80.04	53.26
ViT-B/16 (Ours)	Key	8	LoRA	79.94	78.69	7.62	79.94	7.63
ViT-B/16 (Ours)	Key	16	LoRA	79.71	**78.96**	0.73	79.42	0.72
ConvMixer-768/32 ([67])	Key	7	Partial	71.98	64.74	70.65	70.16	71.47
ResNet50 ([16])	Key	4	Full	75.69	66.95	–	–	–
Swin-L ([71])	AT	–	–	78.92	59.56	–	–	–
ConvNeXt-L ([71])	AT	–	–	78.02	58.48	–	–	–
ConvNeXt-L + ConvStem ([72])	AT	–	–	77.00	57.70	–	–	–

TABLE 4. Trainable Parameters of ViT/B16 With or Without LoRA

Model	# Trainable Parameters ($\times 10^6$)	
	Full	LoRA
ViT-B/16	86.57	**1.38 (1.57%)**

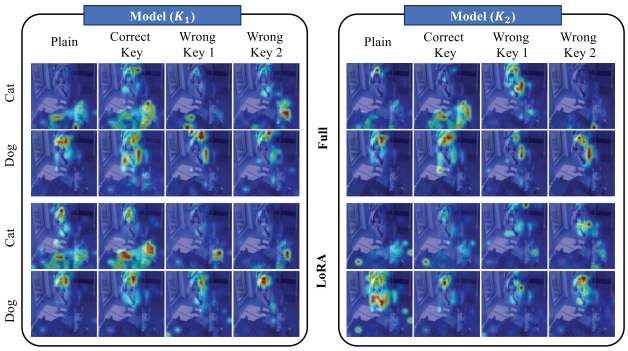


FIGURE 9. Attention visualization of two models trained with encrypted images with two keys under full fine-tuning and LoRA fine-tuning. Attention maps were computed by using different inputs: plain images, encrypted images with correct key, encrypted images with wrong key 1 and 2.

we also generated adversarial examples with PGD-20 on each model and tested the generated adversarial examples across 10 models to observe adversarial transferability. The results are captured in Table 6. When the source model of PGD-20 and the targeted model were the same, the attack was most successful. Otherwise, the attack was not effective. This experiment shows that adversarial examples are not transferable with a high success rate in the key-based defense. However, adversarial examples are transferable on LoRA fine-tuned models.

LoRA Fine-Tuned Models: Although the LoRA fine-tuned model with a larger block size achieved higher accuracy for non-adaptive attacks, our experiments show that simple adaptive attacks can defeat LoRA fine-tuned models. Informed by the previous study [25], adversarial examples have high transferability within the same model architecture. Therefore, our intuition is that if two models are highly similar in the feature space, there is a high chance of adversarial transferability from one to the other.

We conducted a simple experiment to observe the classification accuracy of plain images for full, and LoRA fine-tuned models. Table 7 summarizes the performance of fine-tuned ViT-B/16 ($P = 4$) with two keys K_1 and K_2, when giving different input images. Plain image accuracies of two LoRA models are close (1.45% gap), while fully fine-tuned models have a 4.65% plain image accuracy gap. This hints that two LoRA models may be similar in the feature space. To get further insights, we visualize attention maps of two models under both fine-tuning strategies with different inputs (Fig. 9). The figure shows that LoRA models are less attentive to the class subjects and are inconsistent. This inconsistency hints that it is easier to interfere with the attention of LoRA models than fully fine-tuned ones.

In summary, we support our conjecture that two LoRA models may be similar in some way, aiding adversarial transferability. However, further research is required to investigate how the two models are similar and how much change is needed for adversarial transferability.

TABLE 5. Clean Accuracy (%) of Different Models With Different Keys ($P = 4$) for CIFAR-10

Clean	$\mathcal{M}_{K_1}$	$\mathcal{M}_{K_2}$	$\mathcal{M}_{K_3}$	$\mathcal{M}_{K_4}$	$\mathcal{M}_{K_5}$	$\mathcal{M}_{K_6}$	$\mathcal{M}_{K_7}$	$\mathcal{M}_{K_8}$	$\mathcal{M}_{K_9}$	$\mathcal{M}_{K_{10}}$
K_1	92.45	22.43	22.17	22.66	23.59	26.71	25.16	24.17	21.36	21.93
K_2	27.72	92.81	23.07	21.73	23.95	26.41	21.05	28.05	20.54	21.79
K_3	22.44	21.88	92.86	21.70	25.58	22.45	20.95	20.28	29.49	20.75
K_4	27.11	17.72	21.44	92.74	23.79	22.59	23.61	19.28	26.72	21.43
K_5	25.72	26.62	25.37	24.16	92.49	23.76	23.59	24.54	24.75	24.36
K_6	21.41	21.05	19.98	22.65	19.89	92.33	22.01	17.84	20.37	20.49
K_7	21.38	22.62	20.69	19.80	20.02	21.46	92.59	21.11	21.49	19.92
K_8	27.53	22.05	23.97	25.09	29.49	22.85	24.55	92.85	31.01	23.43
K_9	28.22	18.49	24.59	20.18	25.20	19.92	20.99	22.02	92.48	25.52
K_{10}	22.19	18.40	22.89	18.79	21.25	19.26	18.38	21.46	19.88	92.36

TABLE 6. Robust Accuracy (%) of Different Models With Different Keys ($P = 4$) for CIFAR-10. Adversarial Examples Were Generated on One Model and Tested Across 10 Models With 10 Keys

PGD-20	$\mathcal{M}_{K_1}$	$\mathcal{M}_{K_2}$	$\mathcal{M}_{K_3}$	$\mathcal{M}_{K_4}$	$\mathcal{M}_{K_5}$	$\mathcal{M}_{K_6}$	$\mathcal{M}_{K_7}$	$\mathcal{M}_{K_8}$	$\mathcal{M}_{K_9}$	$\mathcal{M}_{K_{10}}$
$\mathcal{M}_{K_1}$	3.26	72.58	72.28	71.47	71.73	71.84	72.84	69.83	69.86	74.34
$\mathcal{M}_{K_2}$	73.37	4.86	72.07	73.82	72.50	72.63	72.33	71.92	71.55	70.61
$\mathcal{M}_{K_3}$	73.80	72.79	4.63	74.51	74.20	73.47	71.83	73.20	73.91	72.60
$\mathcal{M}_{K_4}$	69.68	71.70	71.11	3.39	70.38	71.03	71.26	69.53	68.52	69.10
$\mathcal{M}_{K_5}$	71.55	72.13	72.47	71.92	1.95	71.53	71.79	69.92	70.05	69.86
$\mathcal{M}_{K_6}$	72.52	72.42	72.50	73.62	72.26	2.18	72.41	71.56	73.79	72.74
$\mathcal{M}_{K_7}$	73.64	71.76	71.24	73.55	72.46	72.75	3.36	71.27	73.51	70.70
$\mathcal{M}_{K_8}$	69.72	71.37	70.44	71.13	69.71	70.74	70.00	3.30	69.98	70.27
$\mathcal{M}_{K_9}$	70.38	71.24	72.33	70.59	70.62	72.52	72.50	71.20	2.61	71.78
$\mathcal{M}_{K_{10}}$	72.48	70.37	71.38	71.67	70.58	72.75	70.29	71.43	71.52	3.61

TABLE 7. Classification Accuracy (%) of Fine-Tuned ViT-B/16 Models for Input Images Transformed by Different Keys With $P = 4$

	Model	Plain	Correct	K'_1	K'_2	K'_3	K'_4
(K_1)	ViT-B/16 (Full)	76.29	**83.40**	58.15	53.68	40.06	53.49
	ViT-B/16 (LoRA)	79.31	**81.12**	51.19	60.94	44.80	49.50
(K_2)	ViT-B/16 (Full)	80.94	**83.23**	51.61	62.08	54.80	55.67
	ViT-B/16 (LoRA)	80.76	**80.99**	48.53	60.43	51.45	49.35

The keys K_1 and K_2 are correct and $K'_1, \ldots, K'_4$ are incorrect.

Attention Visualization: To gain insights on ViT key-based models, we employed attention visualization methods from [74] to fully fine-tuned models and LoRA ones. We utilized an image with a cat and a dog and visualized the attention of models with different block sizes under two fine-tuning strategies. Fig. 10 shows such attention visualization. Interestingly, for this cat-dog test image, the plain model classifies the image as a cat, and fine-tuned models predict it as a dog. Although different modes have different attention, adversarial examples are still transferable for LoRA models. Next, we visualize decision boundaries.

Decision Boundary Visualization: In this analysis, we used CIFAR-10 models for convenience. Inspired by mixup regularizer [75], the previous work plots decision boundaries along the convex hull between data points and shows such visualizations capture on-manifold behavior [76]. We utilize this decision boundary visualization technique to plot along the plane spanned by three data points ("Airplane", "Bird", and "Frog") for the CIFAR-10 dataset based on our key-based defended models. The figure shows the decision boundaries of key-based models with different block sizes and the same block size with different keys. The decision boundary of the model with $P = 2$ is similar to the one of the plain model. This justifies why $P = 2$ model achieves a high clean classification accuracy. In contrast, the decision boundaries of $P = 16$ and 32 are very different. Moreover, on the same block size $P = 4$ with different keys, the decision boundaries also vary. This explains why adversarial examples generated on one key-based model are not transferable with a high success rate except for LoRA fine-tuned models.

B. DISCUSSION

Applicability: Without changing the model architecture, the proposed key-based defense can easily train many different models that yield a similar performance from a pre-trained model by varying keys. Although we adopt block-wise pixel shuffling as the key-based transformation in this paper, other key-based transformations can be applied. We demonstrate that we could train several key-based models even on ImageNet with a limited amount of resources without severely degrading the performance accuracy. With recent

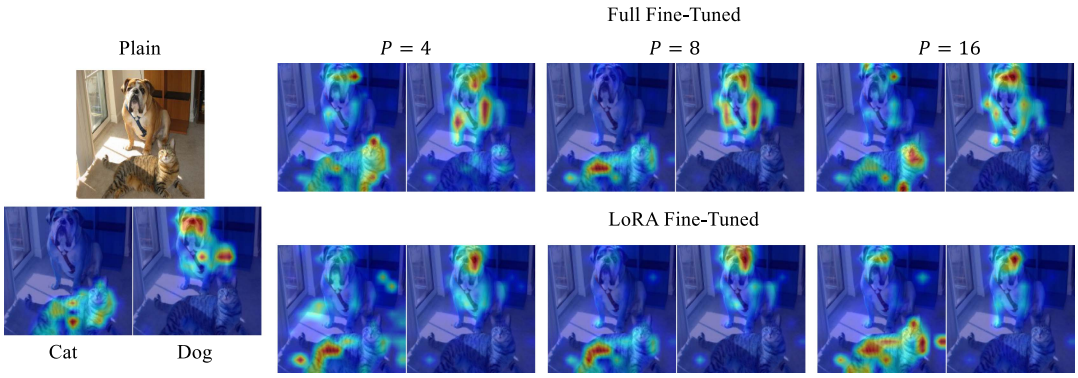


FIGURE 10. Attention visualization for fully fine-tuned models and LoRA fine-tuned models. In each pair, the left image is attention visualization for cat and the right is for dog.

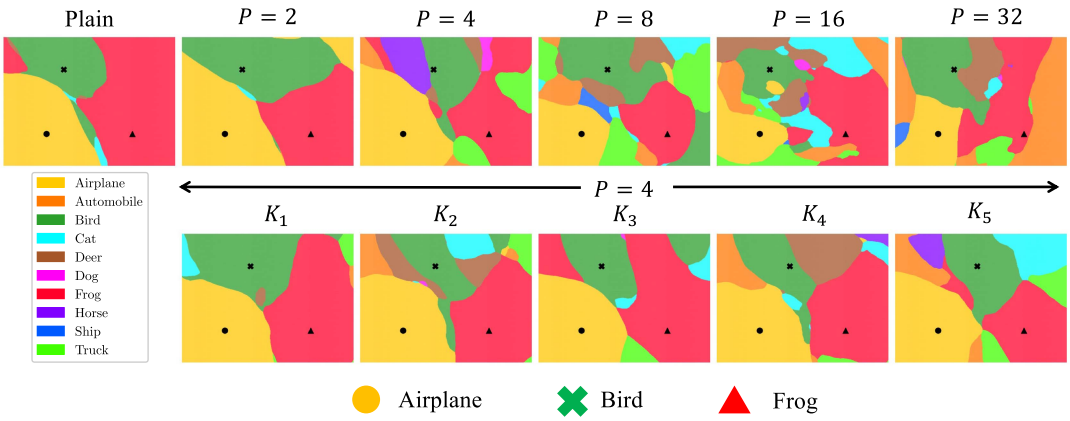


FIGURE 11. Decision boundary visualization for fully fine-tuned models with different block sizes and keys on CIFAR-10.

hardware/platform availability, the one-key-one-model image classification paradigm is feasible and has the potential to defend against adversarial examples in real-world settings (e.g., self-driving cars). Moreover, the previous works also showed that key-based models are diverse and can be used in an ensemble for traditional image classification scenarios [23], [24].

Limitations: In this paper, we deployed p-norm bounded adversaries (specifically AA strategy [66]), which is also used in RobustBench [73]. However, in real-world settings, the adversary is unknown. Another limitation is that we applied a LoRA dimension, $r = 16$, and a value of scaling factor 16 only. It is our first attempt to use LoRA, and we showed that key-based models can easily be obtained by using the latest fine-tuning techniques. Although LoRA fine-tuning was significantly efficient regarding computational cost, our empirical experiments on LoRA models in the current limited settings were not robust against adversarial examples. Therefore, further improvements in LoRA models are required for one-key-one-model adoption. In its current form, LoRA models are not suitable for one-key-one-model applications settings. However, LoRA models may be considered to form an ensemble. Therefore, this may open up a new direction of research.

Future Work: We shall further investigate different attacks, such as patch attacks and unrestricted adversarial examples, to evaluate key-based models. We shall also improve adversarial robustness in LoRA fine-tuned models in our future work.

VI. CONCLUSION

In this paper, we proposed to leverage pre-trained models and use the latest fine-tuning techniques to a key-based defense so that key-based defended models can be easily proliferated even on the ImageNet scale. We stress that such defended models can potentially be deployed on devices for one-key-one-model application scenarios. Our proposed fine-tuned models can easily be obtained even on limited computing resources. Experiment results showed that our fine-tuned models achieved a comparable clean accuracy compared to a non-defended model. Moreover, given the condition that attackers do not know the secret key, our fine-tuned models outperformed state-of-the-art models in terms of both clean and robust accuracy. We also carried out adaptive attacks to further evaluate the proposed fine-tuned models. The results confirmed that our fully fine-tuned models are resistant to adaptive attacks, and LoRA fine-tuned models, in their current form, are not robust against adaptive attacks.

REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[3] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[4] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 6645–6649.

[5] C. Szegedy et al., "Intriguing properties of neural networks," in *Proc. Int. Conf. Learn. Representations*, 2014.

[6] B. Biggio et al., "Evasion attacks against machine learning at test time," in *Proc. Mach. Learn. Knowl. Discov. Databases*, 2013, pp. 387–402.

[7] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. Int. Conf. Learn. Representations*, 2015.

[8] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Proc. Artif. Intell. Saf. Secur.*, 2018, pp. 99–112.

[9] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing robust adversarial examples," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 284–293.

[10] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, 2017, pp. 506–519.

[11] K. Eykholt et al., "Robust physical-world attacks on deep learning visual classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1625–1634.

[12] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: Attacks and defenses for deep learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 9, pp. 2805–2824, Sep. 2019.

[13] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 274–283.

[14] F. Tramer, N. Carlini, W. Brendel, and A. Madry, "On adaptive attacks to adversarial example defenses," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 1633–1645.

[15] O. Taran, S. Rezaeifar, and S. Voloshynovskiy, "Bridging machine learning and cryptography in defence against adversarial attacks," in *Proc. Eur. Conf. Comput. Vis. Workshops*, 2018, pp. 267–279.

[16] A. MaungMaung and H. Kiya, "Block-wise image transformation with secret key for adversarially robust defense," *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 2709–2723, 2021.

[17] H. Kiya, A. MaungMaung, Y. Kinoshita, S. Imaizumi, and S. Shiota, "An overview of compressible and learnable image transformation with secret key and its applications," *APSIPA Trans. Signal Inf. Process.*, vol. 11, no. 1, 2022, Art. no. e11.

[18] A. MaungMaung and H. Kiya, "Encryption inspired adversarial defense for visual classification," in *Proc. IEEE Int. Conf. Image Process.*, 2020, pp. 1681–1685.

[19] R. Iijima, S. Shiota, and H. Kiya, "A random ensemble of encrypted vision transformers for adversarially robust defense," *IEEE Access*, vol. 12, pp. 69206–69216, 2024.

[20] A. Kerckhoffs, "La cryptographie militaire," *J. Des Sci. Militaires*, vol. 9, pp. 5–38, 1883.

[21] S. Garg, S. Jha, S. Mahloujifar, and M. Mohammad, "Adversarially robust learning could leverage computational hardness," in *Proc. Algorithmic Learn. Theory*, 2020, pp. 364–385.

[22] A. A. Rusu, D. A. Calian, S. Gowal, and R. Hadsell, "Hindering adversarial attacks with implicit neural representations," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 18910–18934.

[23] O. Taran, S. Rezaeifar, T. Holotyak, and S. Voloshynovskiy, "Machine learning through cryptographic glasses: Combating adversarial attacks by key-based diversified aggregation," *EURASIP J. Inf. Secur.*, vol. 2020, no. 1, pp. 1–18, 2020.

[24] A. MaungMaung and H. Kiya, "Ensemble of key-based models: Defense against black-box adversarial attacks," in *Proc. IEEE Glob. Conf. Consum. Electron.*, 2021, pp. 95–98.

[25] D. Su, H. Zhang, H. Chen, J. Yi, P.-Y. Chen, and Y. Gao, "Is robustness the cost of accuracy? A comprehensive study on the robustness of 18 deep image classification models," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 631–648.

[26] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," in *Proc. Int. Conf. Learn. Representations*, 2017.

[27] H. Shen, H. Chang, B. Dong, Y. Luo, and H. Meng, "Efficient LLM inference on CPUs," 2023, *arXiv:2311.00502*.

[28] A. Orhon, M. Siracusa, and A. Wadhwa, "Stable diffusion with core ML on apple silicon," 2022. [Online]. Available: https://machinelearning.apple.com/research/stable-diffusion-coreml-apple-silicon

[29] I. Goodfellow, N. Papernot, S. Huang, Y. Duan, and P. Abbeel, "Attacking machine learning with adversarial examples," 2017. [Online]. Available: https://openai.com/index/attacking-machine-learning-with-adversarial-examples/

[30] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. Int. Conf. Learn. Representations*, 2018.

[31] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: A simple and accurate method to fool deep neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2574–2582.

[32] P.-Y. Chen, Y. Sharma, H. Zhang, J. Yi, and C.-J. Hsieh, "EAD: Elastic-net attacks to deep neural networks via adversarial examples," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 10–17.

[33] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Proc. IEEE Eur. Symp. Secur. Privacy*, 2016, pp. 372–387.

[34] T. B. Brown, N. Carlini, C. Zhang, C. Olsson, P. Christiano, and I. Goodfellow, "Unrestricted adversarial examples," 2018, *arXiv:1809.08352*.

[35] L. Engstrom, B. Tran, D. Tsipras, L. Schmidt, and A. Madry, "Exploring the landscape of spatial robustness," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 1802–1811.

[36] Y. Song, R. Shu, N. Kushman, and S. Ermon, "Constructing unrestricted adversarial examples with generative models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 8322–8333.

[37] K. Xu et al., "Adversarial t-shirt! evading person detectors in a physical world," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 665–681.

[38] S. Komkov and A. Petiushko, "ADVHat: Real-world adversarial attack on arcface face ID system," in *Proc. Int. Conf. Pattern Recognit.*, 2021, pp. 819–826.

[39] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 1528–1540.

[40] N. Guetta, A. Shabtai, I. Singh, S. Momiyama, and Y. Elovici, "Dodging attack using carefully crafted natural makeup," 2021, *arXiv:2109.06467*.

[41] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer, "Adversarial patch," 2017, *arXiv:1712.09665*.

[42] A. Shafahi et al., "Adversarial training for free!," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 3353–3364.

[43] E. Wong, L. Rice, and J. Z. Kolter, "Fast is better than free: Revisiting adversarial training," in *Proc. Int. Conf. Learn. Representations*, 2020.

[44] P. de Jorge Aranda et al., "Make some noise: Reliable and efficient single-step adversarial training," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 12881–12893.

[45] Y. Sharma and P.-Y. Chen, "Attacking the Madry defense model with L_1-based adversarial examples," 2017, *arXiv:1710.10733*.

[46] E. Wong and Z. Kolter, "Provable defenses against adversarial examples via the convex outer adversarial polytope," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5286–5295.

[47] A. Raghunathan, J. Steinhardt, and P. Liang, "Certified defenses against adversarial examples," in *Proc. Int. Conf. Learn. Representations*, 2018.

[48] J. Cohen, E. Rosenfeld, and Z. Kolter, "Certified adversarial robustness via randomized smoothing," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 1310–1320.

[49] M. Hein and M. Andriushchenko, "Formal guarantees on the robustness of a classifier against adversarial manipulation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 2266–2276.

[50] O. Poursaeed, I. Katsman, B. Gao, and S. Belongie, "Generative adversarial perturbations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4422–4431.

[51] H.-T. D. Liu, M. Tao, C.-L. Li, D. Nowrouzezahrai, and A. Jacobson, "Beyond pixel norm-balls: Parametric adversaries using an analytically differentiable renderer," in *Proc. Int. Conf. Learn. Representations*, 2019.

[52] J. Buckman, A. Roy, C. Raffel, and I. Goodfellow, "Thermometer encoding: One hot way to resist adversarial examples," in *Proc. Int. Conf. Learn. Representations*, 2018.

[53] C. Guo, M. Rana, M. Cisse, and L. van der Maaten, "Countering adversarial images using input transformations," in *Proc. Int. Conf. Learn. Representations*, 2018.
[54] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille, "Mitigating adversarial effects through randomization," in *Proc. Int. Conf. Learn. Representations*, 2018.
[55] F. Liao, M. Liang, Y. Dong, T. Pang, X. Hu, and J. Zhu, "Defense against adversarial attacks using high-level representation guided denoiser," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1778–1787.
[56] Z. Niu, Z. Chen, L. Li, Y. Yang, B. Li, and J. Yi, "On the limitations of denoising strategies as adversarial defenses," 2020, *arXiv:2012.09384*.
[57] Y. Song, T. Kim, S. Nowozin, S. Ermon, and N. Kushman, "Pixeldefend: Leveraging generative models to understand and defend against adversarial examples," in *Proc. Int. Conf. Learn. Representations*, 2018.
[58] A. P. Steiner, A. Kolesnikov, X. Zhai, R. Wightman, J. Uszkoreit, and L. Beyer, "How to train your ViT? Data, augmentation, and regularization in vision transformers," *Trans. Mach. Learn. Res.*, 2022. [Online]. Available: https://openreview.net/forum?id=4nPswr1KcP
[59] E. J. Hu et al., "LoRA: Low-rank adaptation of large language models," in *Proc. Int. Conf. Learn. Representations*, 2022.
[60] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3320–3328.
[61] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.
[62] T. Ridnik, E. Ben-Baruch, A. Noy, and L. Zelnik-Manor, "ImageNet-21 K pretraining for the masses," in *Proc. 34th Conf. Neural Inf. Process. Syst. Datasets Benchmarks Track (Round 1)*, 2021.
[63] R. Bommasani et al., "On the opportunities and risks of foundation models," 2021, *arXiv:2108.07258*.
[64] A. Dosovitskiy et al., "An image is worth 16 × 16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Representations*, 2021.
[65] N. Carlini et al., "On evaluating adversarial robustness," 2019, *arXiv:1902.06705*.
[66] F. Croce and M. Hein, "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 2206–2216.
[67] A. MaungMaung, I. Echizen, and H. Kiya, "Hindering adversarial attacks with multiple encrypted patch embeddings," in *Proc. IEEE Asia Pacific Signal Inf. Process. Assoc. Annu. Summit Conf.*, 2023, pp. 1398–1404.
[68] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.
[69] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Univ. Toronto, Tech. Rep., 2009.
[70] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11976–11986.
[71] C. Liu et al., "A comprehensive study on robustness of image classification models: Benchmarking and rethinking," 2023, *arXiv:2302.14301*.
[72] N. D. Singh, F. Croce, and M. Hein, "Revisiting adversarial training for imagenet: Architectures, training and generalization across threat models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2024, vol. 36.
[73] F. Croce et al., "Robustbench: A standardized adversarial robustness benchmark," in *Proc. 35th Conf. Neural Inf. Process. Syst. Datasets Benchmarks Track (Round 2)*, 2021.
[74] H. Chefer, S. Gur, and L. Wolf, "Transformer interpretability beyond attention visualization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 782–791.
[75] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 13689–13698.
[76] G. Somepalli et al., "Can neural nets learn the same model twice? Investigating reproducibility and double descent from the decision boundary perspective," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 13699–13708.