Efficient Adversarial Training with Membership Inference Resistance

Ran Yan[1], Ruiying Du[1,2(✉)], Kun He[1], and Jing Chen[1,3]

[1] Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China
{yanran22,hekun,chenjing}@whu.edu.cn
[2] Collaborative Innovation Center of Geospatial Technology, Wuhan 430079, China
duraying@whu.edu.cn
[3] Rizhao Institute of Information Technology, Wuhan University, Rizhao 276800, China

Abstract. Deep cross-media computing faces adversarial example attacks, adversarial training is an effective approach to enhance the robustness of machine learning models via adding adversarial examples into the training phase. However, existing adversarial training methods increase the advantage of membership inference attacks, which aim to determine from the model whether an example is in the training dataset. In this paper, we propose an adversarial training framework that guarantees both robustness and membership privacy by introducing a tailor-made example, called reverse-symmetry example. Moreover, our framework reduces the number of required adversarial examples compared with existing adversarial training methods. We implement the framework based on three adversarial training methods on FMNIST and CIFAR10. The experimental results show that our framework outperforms the original adversarial training with respect to the overall performance of accuracy, robustness, privacy, and runtime.

Keywords: Adversarial training · adversarial examples · membership inference attacks

1 Introduction

Deep Neural Networks (DNNs) have been employed across various real-world applications in deep cross-media computing, such as intelligent image analysis [7, 17], natural language processing [1], and speech recognition [4]. Unfortunately, deep learning models trained by DNNs are found to be vulnerable to evasion attacks [3,12], in which an attacker misleads deep learning models by adding imperceptible perturbations to natural examples, called adversarial examples. In practice, those adversarial examples can be used for crime by tricking models deployed in daily applications, such as facial recognition [15].

Q. Liu et al. (Eds.): PRCV 2023, LNCS 14425, pp. 474–486, 2024.
https://doi.org/10.1007/978-981-99-8429-9_38

One of the promising approaches to defend against evasion attacks is *adversarial training*, which can be classified into two categories: *empirical* and *verifiable* [20]. The core idea of empirical adversarial training is to add adversarial examples in the training phase of a model to enhance the robustness [10]. To make the adversarial examples more representative of the adversarial domain, adversarial training usually employs multiple iterative processes [18], such as Projected Gradient Descent (PGD) [8]. Therefore, empirical adversarial training suffers from high runtime in the training phase. On the other hand, verifiable adversarial training estimates the adversarial domain around natural examples in the training phase, which usually does not significantly increase the runtime [5,11]. However, since the estimation is done for the worst case, verifiable adversarial training sacrifices the accuracy of the model.

In addition to the above-mentioned performance issues, existing adversarial training methods also increase the privacy risk of deep learning models [20]. Membership inference attack is a typical privacy threat, in which an attacker tries to determine whether a given example is used to train a model [16]. This kind of privacy leakage is dangerous in many applications. Taking medical recognition as an example, an attacker can infer the membership information of a patient's medical record from a special disease diagnosis model by the membership inference attack, which violates the patient's privacy. Experimental results in [20] showed that, compared with the normally trained models, both empirical and verifiable adversarial training lead to an increase in the advantage of membership inference attacks.

In this paper, we aim to reinforce the paradigm of current empirical adversarial training so as to reduce the privacy risk and runtime while maintaining robustness and accuracy. We focus on empirical adversarial training due to its high accuracy compared with verifiable methods. The key insight is that the adversarial examples generated in the training phase are clustered in specific areas, which means that the model may only focus on local features of the adversarial domain. Therefore, we introduce additional tailor-made examples to reduce the model's attention to the adversarial domain, which also reduces the privacy risk. Since those tailor-made examples can be efficiently generated and they retain the features of natural examples, we can replace parts of adversarial examples with those tailor-made examples to reduce the runtime without significantly reducing robustness and accuracy.

In summary, we make the following main contributions.

- We propose a framework that can integrate and enhance existing adversarial training methods. Moreover, we extend our framework to tune the trade-offs between accuracy, robustness, privacy, and runtime.
- To demonstrate the effectiveness of our framework, we implement the framework based on three empirical adversarial training methods on FMNIST and CIFAR10. The results show that our framework has better overall performance than the underlying methods.

– To explore the trade-offs in our framework, we implement 9 variants on FMNIST and CIFAR10. The results provide helpful guidance for developers to choose settings that meet their various requirements.

2 Background and Related Work

2.1 Workflow of Empirical Adversarial Training

Empirical adversarial training is an effective way to resist adversarial examples. The main idea of empirical adversarial training is to transform the problem of finding a robust model into an optimization problem that minimizes the combination of natural loss and robust loss.

At the beginning of the training phase, a model is initialized by determining its architecture and hyper-parameters, such as learning rate and step size. Then, the model is repeatedly trained for a certain number of training rounds. Finally, an optimal model is outputted at the end of the training phase. In each training round of adversarial training, the model F_θ learns the training dataset through the following three procedures: (1) *Sampling.* A set of $(\mathbf{x}_{nat}, y)$, i.e., batch data $(\mathbf{X}_{nat}, \mathbf{y})$, is sampled from the training dataset as same as the way in natural training. (2) *Generating.* For each $(\mathbf{x}_{nat}, y)$ in the batch data, an adversarial example $\mathbf{x}_{adv}$ is generated. (3) *Updating.* The weights θ is updated by minimizing $\sum_{\mathbf{x}_{nat} \in \mathbf{X}_{nat}} \ell(F_\theta, (\mathbf{x}_{nat}, \mathbf{x}_{adv}, y))$. More specifically,

$$\begin{aligned} \ell(F_\theta, (\mathbf{x}_{nat}, \mathbf{x}_{adv}, y)) = \alpha \cdot \ell_n(F_\theta, (\mathbf{x}_{nat}, y)) \\ + (1 - \alpha) \cdot \ell_r(F_\theta, (\mathbf{x}_{nat}, \mathbf{x}_{adv}, y)), \end{aligned}$$

where ℓ_n is the natural loss as in natural training, i.e., cross-entropy loss, ℓ_r is the robust loss, and α is used to balance the natural loss and robust loss.

2.2 Membership Inference Attacks

Membership inference attacks aim at determining whether an example was in the training dataset of a given machine learning model [2,9]. This kind of attack may leak sensitive information of individuals once combined with background knowledge about the model. Yeom et al. [22] consider an example as a member of the training dataset if the final prediction corresponds with the ground-truth label. Salem et al. [14] set a threshold for the prediction confidence to determine membership, which does not require an attacker to decide the ground-truth label for target examples. Their experimental results show that this *confidence thresholding* method can obtain a similar inference accuracy with that of a complex attack methods. Song et al. [19] proposed to use the entropy of the confidence as the threshold to implement the membership inference attack. To improve the inference advantage, we use the confidence and its corresponding ground-truth label to compute cross entropy as the threshold, called *cross-entropy thresholding*, to implement the membership inference attack.

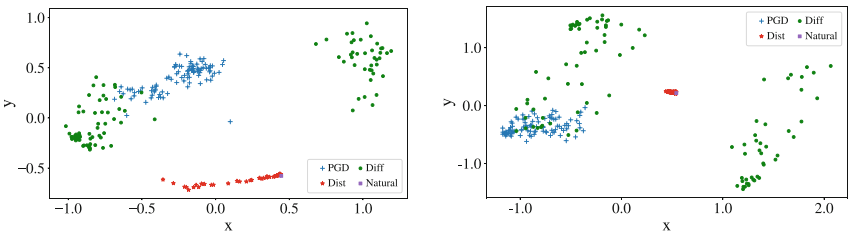
3 Our Method

We propose a privacy-enhancing adversarial training framework, called PINEAT, which is a recursive acronym for PINEAT Is Not Exactly Adversarial Training. After the design goals are stated, we give an overview of our framework and then describe the details.

3.1 Design Goals

Our framework aims to achieve the following goals.

- **Universal applicability.** Our framework should be compatible with existing (empirical) adversarial training methods. Specifically, we should not modify the generation of adversarial examples and loss functions.
- **Adversarial robustness.** The model trained by our framework should be able to resist adversarial examples. Moreover, the adversarial robustness should be similar to that of the underlying adversarial training method.
- **Membership privacy.** Compared with the underlying adversarial training method and even natural training method, the model trained by our framework should be resistant to the membership inference attack.
- **Runtime reduction.** The runtime of our framework should be lower than the runtime of the underlying adversarial training methods.



(a) The index of natural example is 300 in FMNIST.

(b) The index of natural example is 304 in FMNIST.

Fig. 1. Two natural examples and all their adversarial examples via PCA dimension reduction [13].

3.2 Overview

We have analyzed the adversarial examples generated in three main adversarial training methods, including *PGD-based* [10], *Distribution-based* (Dist-based) [18], and *Difference-based* (Diff-based) [23]. The key observation is that adversarial examples for the training dataset are clustered around specific areas of natural examples, as shown in Fig. 1. Therefore, the model may overfit in

those areas, i.e., adversarial domains around natural examples, resulting in an increased privacy risk of the model in adversarial domains.

To alleviate the model's excessive attention to adversarial domains, we can introduce additional examples in the training phase. These additional examples should increase the dispersion of all examples based on the same natural example used for training as much as possible. Therefore, we design a tailor-made example, called *reverse-symmetry example*, which is symmetrical with an adversarial example centered at the natural example. To keep the ground-truth label of the reverse-symmetry example consistent with that of the natural example, we require the reverse-symmetry example to satisfy the perturbation constraint as the adversarial example, i.e., $\mathbf{x}_{rs} \in \mathcal{B}_\epsilon(\mathbf{x}_{nat})$, where $\mathbf{x}_{rs}$ is the reverse-symmetry example and $\mathbf{x}_{nat}$ is the natural example.

To reduce the training runtime, we replace parts of adversarial examples whose generation is time-consuming with lightweight reverse-symmetry examples. Since reverse-symmetry examples retain the features of natural examples and enough adversarial examples are involved, this replacement will not reduce accuracy and robustness.

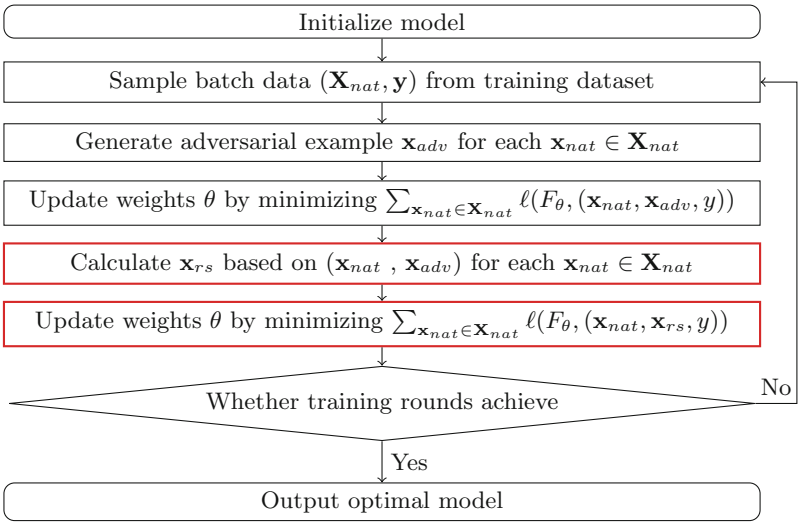


Fig. 2. Workflow of PINEAT, where the red boxes indicate the differences from empirical adversarial training. (Color figure online)

3.3 Design Details

PINEAT is a framework that extends the current empirical adversarial training paradigm, as shown in Fig. 2. Compared with existing adversarial training methods, PINEAT introduces two additional procedures to each training round.

First, an reverse-symmetry example is calculated based on the natural example and the corresponding adversarial example, satisfying $\mathbf{x}_{rs} + \mathbf{x}_{adv} = \mathbf{x}_{nat} * 2$, where $\mathbf{x}_{rs}$ is the reverse-symmetry example, $\mathbf{x}_{adv}$ is the adversarial example, and $\mathbf{x}_{nat}$ is the natural example. Since $\mathbf{x}_{adv}$ is generated under the perturbation constraint $\mathcal{B}_\epsilon(\mathbf{x}_{nat})$, $\mathbf{x}_{rs}$ also satisfies $\mathcal{B}_\epsilon(\mathbf{x}_{nat})$ and is indistinguishable from $\mathbf{x}_{nat}$ by human eye, which means that $\mathbf{x}_{rs}$ and $\mathbf{x}_{nat}$ have the same ground-truth label. Second, the model is updated again by minimizing the combination of loss on the natural examples and loss on reverse-symmetry examples. The loss function depends on the underlying adversarial training method.

Note that since the model is updated twice in a training round, our framework halves the number of training rounds required by the underlying adversarial training method. In other words, half of the adversarial examples generated in traditional adversarial training methods are replaced by reverse-symmetry examples in PINEAT. We denote this setting PINEAT-1/1, which means the ratio of the number of adversarial examples and the number of reverse-symmetry examples is 1:1.

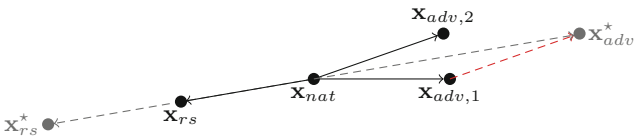


Fig. 3. Generation of reverse-symmetry examples in PINEAT-2/1. After generating two adversarial examples for the natural example, we aggregate the noises in $\mathbf{x}_{adv,1}$ and $\mathbf{x}_{adv,2}$ against the natural example and get $\mathbf{x}^\star_{adv}$. The red line represents the noise tensor $\mathbf{x}_{adv,2} - \mathbf{x}_{nat}$ that is translated from $\mathbf{x}_{nat}$ to $\mathbf{x}_{adv,1}$. Therefore, we have $\mathbf{x}^\star_{adv} - \mathbf{x}_{nat} = (\mathbf{x}_{adv,1} - \mathbf{x}_{nat}) + (\mathbf{x}_{adv,2} - \mathbf{x}_{nat})$. Then, we calculate $\mathbf{x}^\star_{rs}$ for $\mathbf{x}^\star_{adv}$ where $\mathbf{x}^\star_{rs} + \mathbf{x}^\star_{adv} = 2\mathbf{x}_{nat}$. Finally, we get $\mathbf{x}_{rs}$ by clip $\mathbf{x}^\star_{rs}$, i.e., $\mathbf{x}_{rs} = min(max(\mathbf{x}^\star_{rs}, \mathbf{x}_{nat} - \epsilon), \mathbf{x}_{nat} + \epsilon)$.

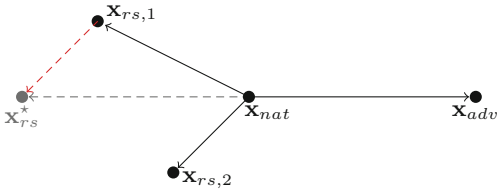


Fig. 4. Generation of reverse-symmetry examples in PINEAT-1/2. We calculate $\mathbf{x}^\star_{rs}$ which is symmetric with $\mathbf{x}_{adv}$ respect to $\mathbf{x}_{nat}$. Then, we decompose the noise in $\mathbf{x}^\star_{rs}$ into the two reverse-symmetry examples $\mathbf{x}_{rs,1}$ and $\mathbf{x}_{rs,2}$. The red line represent the noise tensor assigned to $\mathbf{x}^2_{rs}$, i.e., $\mathbf{x}^\star_{rs} - \mathbf{x}_{rs,1} = \mathbf{x}_{rs,2} - \mathbf{x}_{nat}$. In this setting, the $\mathbf{x}_{rs,1}$ and $\mathbf{x}_{rs,2}$ meet the $\mathcal{B}_\epsilon(\mathbf{x}_{nat})$ without cliping, due to $\mathbf{x}^\star_{rs} \in \mathcal{B}_\epsilon(\mathbf{x}_{nat})$.

Variants. We can adjust the adversarial/reverse-symmetry examples ratio to obtain different settings, which also means the number of model updates in a training round and the number of training rounds are changed accordingly. Specifically, in each training round of PINEAT-n/m, m reverse-symmetry examples are calculated for n adversarial examples, satisfying that $\mathbf{x}^{\star}_{adv} + \mathbf{x}^{\star}_{rs} = \mathbf{x}_{nat} * 2$, where $\mathbf{x}^{\star}_{adv}$ aggregates the noises in n adversarial examples (i.e., $\mathbf{x}^{\star}_{adv} - \mathbf{x}_{nat} = \sum_{i=1}^{n}(\mathbf{x}_{adv,i} - \mathbf{x}_{nat})$, where $\mathbf{x}_{adv,i}$ represents the i-th adversarial example) and the noise in $\mathbf{x}^{\star}_{rs}$ is decomposed into m reverse-symmetry examples. To generate adversarial examples, we use n different random seeds for one natural example. To calculate multiple reverse-symmetry examples, firstly we randomly generate $m-1$ reverse-symmetry examples under the perturbation constraint $\mathcal{B}_{\epsilon}(\mathbf{x}_{nat})$. Then, we calculate the last reverse-symmetry example by $\mathbf{x}_{rs,m} = \min(\max(\mathbf{x}^{\star}_{rs} - \sum_{j=1}^{m-1}\mathbf{x}_{rs,j}, \mathbf{x}_{nat} - \epsilon), \mathbf{x}_{nat} + \epsilon)$, where $\mathbf{x}_{rs,j}$ represents the j-th reverse-symmetry example. The function $\min(\max(\cdot, \mathbf{x}_{nat} - \epsilon), \mathbf{x}_{nat} + \epsilon)$ guarantees that the last reverse-symmetry example also satisfies the perturbation constraint $\mathcal{B}_{\epsilon}(\mathbf{x}_{nat})$. In PINEAT-$n/m$, the model is updated $n+m$ times in one training round thus our framework needs about $1/(n+m)$ of the number of training rounds required by the underlying adversarial training method. Figure 3 and 4 illustrate the generation of reverse-symmetry examples in PINEAT-2/1 and PINEAT-1/2, respectively.

4 Experimental Setup

4.1 Implementation, Dataset, and Model Architecture

The PGD-based adversarial training and Dist-based adversarial training are implemented based on TensorFlow, and the Diff-based adversarial training is implemented based on Pytorch. All implementations run on a server with Intel Xeon E5-2680, three Nvidia Tesla V100 32GB GPU, and 128G RAM. The datasets and model architectures used in our experiments are identical with those in [20]. Specifically, we adopt two standard datasets: *FMNIST* [21] and *CIFAR10* [6].

4.2 Baselines

We adopt four kinds of baselines: *natural training*, *original adversarial training*, *AdvNat-1/1*, and *AdvRan-1/1*. For the original adversarial training, we implement three methods: PGD-based adversarial training, Dist-based adversarial training, and Diff-based adversarial training. Following [20], we set the l_{∞} perturbation budget ϵ to be 0.1 on FMNIST and 8/255 on CIFAR10.

For each adversarial training, we develop two additional baselines: AdvNat-1/1 and AdvRan-1/1. In AdvRan-1/1, we replace half of adversarial examples with random examples near the natural example under the perturbation constraint $\mathcal{B}_{\epsilon}(\mathbf{x}_{nat})$. In AdvNat-1/1, we use the natural example to replace the half of adversarial examples.

Table 1. Accuracy, robustness, privacy, and runtime of the model trained by different methods on FMNIST and CIFAR10, where the bold number indicates the best value between the methods on the same dataset and the underlined number indicates the best value between the methods based on the same adversarial training on the same dataset.

DS	Method		Accuracy		Robustness		Privacy			Time
			A_{train}	A_{test}	R_{train}	R_{test}	P_{Enat}	P_{Eadv}	P_{MAX}	
FMNIST	Natural		**1.0000**	0.9281	0.0534	0.0514	0.5718	**0.5011**	0.5718	1.00 h
	PGD	Original	0.9993	0.9088	**0.9692**	0.6776	0.5910	0.6493	0.6493	6.18 h
		AdvNat-1/1	**1.0000**	0.9160	0.9148	0.6737	0.5877	0.6282	0.6282	3.88 h
		AdvRan-1/1	0.9999	0.9129	0.9661	0.6730	0.6008	0.6506	0.6506	3.93 h
		PINEAT-1/1	0.9977	0.9136	0.8791	0.6824	0.5683	0.6052	0.6052	3.90 h
	Dist	Original	0.9795	0.9084	0.6768	0.5139	0.5975	0.6043	0.6043	6.78 h
		AdvNat-1/1	**1.0000**	**0.9333**	0.5073	0.4494	0.5819	0.5562	0.5819	3.85 h
		AdvRan-1/1	0.9731	0.9090	0.3409	0.3108	0.5562	0.5330	0.5562	3.90 h
		PINEAT-1/1	0.9332	0.8990	0.6175	0.5371	**0.5410**	0.5517	**0.5517**	3.82 h
	Diff	Original	0.9933	0.9082	0.9009	0.7264	0.5807	0.5936	0.5936	8.83 h
		AdvNat-1/1	0.9830	0.9093	0.8353	0.7258	0.5601	0.5614	0.5614	4.60 h
		AdvRan-1/1	0.9828	0.9095	0.8367	0.7250	0.5621	0.5625	0.5625	5.13 h
		PINEAT-1/1	0.9832	0.9102	0.8495	**0.7305**	0.5616	0.5666	0.5666	5.12 h
CIFAR10	Natural		**1.0000**	**0.9528**	0.0000	0.0000	0.5769	**0.5001**	**0.5769**	11.18 h
	PGD	Original	**1.0000**	0.8704	**0.9838**	0.4726	0.7627	0.7892	0.7892	80.75 h
		AdvNat-1/1	0.9999	0.8900	0.8993	0.4679	0.6638	0.7264	0.7264	44.24 h
		AdvRan-1/1	0.9894	0.8703	0.7763	0.4572	0.6412	0.6747	0.6747	45.25 h
		PINEAT-1/1	0.9774	0.8687	0.7365	0.4706	0.6182	0.6490	0.6490	44.07 h
	Dist	Original	**1.0000**	0.9027	0.4035	0.2686	0.6760	0.6434	0.6760	73.90 h
		AdvNat-1/1	**1.0000**	0.9259	0.3666	0.2940	0.6148	0.5957	0.6148	39.35 h
		AdvRan-1/1	0.9996	0.8950	0.3127	0.2382	0.6629	0.6273	0.6629	39.05 h
		PINEAT-1/1	0.9996	0.9039	0.3093	0.2410	0.6560	0.6242	0.6560	39.67 h
	Diff	Original	0.9975	0.8917	0.7175	**0.4749**	0.6190	0.6541	0.6541	35.18 h
		AdvNat-1/1	0.9725	0.8771	0.5502	0.4398	0.5754	0.5916	0.5916	20.67 h
		AdvRan-1/1	0.9796	0.8885	0.4555	0.3906	**0.5748**	0.5830	0.5830	22.50 h
		PINEAT-1/1	0.9941	0.9053	0.5458	0.4223	0.5899	0.6000	0.6000	22.55 h

4.3 Metrics

We evaluate the performance of our basic framework and its variants from the following four aspects.

- *Accuracy.* This metric is the proportion of examples in the dataset that are categorized to the correct class. The accuracy of the models on the natural training example is denoted as A_{train} and the accuracy of the models on the natural test example is denoted as A_{test}.
- *Robustness.* We use the ability of a model to correctly classify adversarial examples to indicate the robustness of the model. The adversarial examples are generated via PGD method for each natural example in the dataset. We denote the accuracy on the adversarial examples generated from training dataset as R_{train}, and the accuracy on the adversarial examples from test dataset as R_{test}.
- *Privacy.* We use membership inference accuracy to measure the privacy of a model. We use P_{Enat} to denote the accuracy of CrossEntropy-thresholding

method which exploits natural examples, and P_{Eadv} to denote the accuracy of the method which exploits adversarial examples generated under $\mathcal{B}_\epsilon$.
– *Runtime.* This metric refers to the time for training a model in our experimental environment.

5 Experimental Results

5.1 Performance of Our Basic Framework

We evaluate the accuracy, robustness, privacy, and runtime of our basic framework PINEAT-1/1 and the baselines. The performance comparison is shown in Table 1.

Observation 1. *In terms of accuracy, the model trained by PINEAT-1/1 in most cases performs better than the original adversarial training method and AdvRan-1/1, but performs worse than AdvNat-1/1.*

We focus on the metric A_{test} since it can better reflect the accuracy of the model in practice. Diff-based PINEAT-1/1 outperforms the corresponding original method, AdvNat-1/1, and AdvRan-1/1 on both FMNIST and CIFAR10. However, PGD-based and Dist-based AdvNat-1/1 outperform others, with an increase of at most 3.43%.

Observation 2. *In terms of robustness, the model trained by PINEAT-1/1 performs similarly to the original adversarial training method and in most cases performs better than AdvNat-1/1 or AdvRan-1/1.*

We focus on the metric R_{test} since it can better reflect the robustness of the model in practice. On FMNIST, PINEAT-1/1 outperforms the original method while the latter outperforms AdvNat-1/1 and AdvRan-1/1. On CIFAR10, the original method in most cases outperforms others while the difference between PINEAT-1/1 and the original method is at most 5.26%.

Observation 3. *In terms of privacy, the model trained by PINEAT-1/1 performs better than the original adversarial training method, similar to AdvNat-1/1, and in most cases better than AdvRan-1/1.*

We focus on the metric P_{MAX} since in practice an attacker can implement various membership inference attacks and use the best result. PINEAT-1/1 outperforms the original method on both FMNIST and CIFAR10, with an increase of at least 2% and at most 14.02%. Moreover, PINEAT-1/1 sometimes even outperforms the natural training method.

Observation 4. *In terms of runtime, PINEAT-1/1 performs better than the original adversarial training method and performs similarly to AdvNat-1/1 and AdvRan-1/1.*

The runtime depends on the number of adversarial examples generated in the training phase, which is a time-consuming operation. Therefore, the natural training method always has the best runtime while the original adversarial training method always has the worst. PINEAT-1/1, AdvNat-1/1, and AdvRan-1/1 have a similar runtime since half of the adversarial examples are replaced in these methods. AdvNat-1/1 has the least runtime, because it misses a step compared to AdvRan-1/1 (generate random examples) and PINEAT-1/1 (calculate tailor-made examples).

Observation 5. *PINEAT-1/1 achieves a good balance between accuracy, robustness, privacy, and runtime.*

This observation can be obtained directly from the previous three observations. Specifically, in terms of overall performances, the best method on FMNIST and CIFAR10 is Diff-based PINEAT-1/1.

5.2 Performance of Our Framework Variants

In PINEAT-n/m, n adversarial examples and m reverse-symmetry examples are generated in a training round and the number of training rounds is determined by $m+n$ (i.e., $1/(n+m)$ of that of the underlying adversarial training method). We focus on the accuracy, robustness, privacy, and runtime with the change of adversarial/reverse-symmetry example ratio n/m. The performance comparison is shown in the supplementary material table.

Observation 6. *Given the number of training rounds, the accuracy increases as the adversarial/reverse-symmetry example ratio decreases in most cases.*

We focus on the metric A_{test} as in Observation 1. This observation applies to most cases, except for PGD-based and Dist-based PINEAT-1/m on FMNIST. We believe that when there are few adversarial examples, reverse-symmetry examples may disturb the model boundary. We think this phenomenon is related to the loss function. The loss function of Diff-based method contains the natural loss ($\alpha = 1/2$) while PGD-based and Dist-based methods only have the robust loss ($\alpha = 0$), thus the model boundary in the latter two methods is susceptible to interference of reverse-symmetry examples' bias when the example space is small. In addition, most (58.3%) variants perform better than the underlying adversarial training method and some variants are even close to the natural training method, such as Dist-based PINEAT-2/3 on FMNIST and Diff-based PINEAT-1/3 on CIFAR10.

Observation 7. *Given the number of reverse-symmetry examples calculated in a training round, the robustness decreases as the adversarial/reverse-symmetry example ratio decreases in most cases.*

We focus on the metric R_{test} as in Observation 2. This observation applies to most cases, except for four variants. In Dist-based PINEAT-1/2 and PINEAT-1/3 on CIFAR10, the robustness of models does not fall but increases. Dist-based adversarial examples are not of high quality, showing that robustness

of model trained by the original Dist-based is much lower than the other two original methods. Then, the random reverse-symmetry examples near the natural example maybe instead play the role of adversarial examples. In the PGD-based PINEAT-1/2 and PINEAT-1/3 on FMNIST, the robustness of models also increases. We believe that the PGD-based adversarial examples are too concentrated in specific area, and the reverse-symmetry examples hit possible adversarial area that the PGD-based adversarial examples did not consider before. In addition, most (64.4%) variants perform better than the underlying adversarial training method.

Observation 8. *Given the number of adversarial examples generated in a training round, the privacy becomes better as the adversarial/reverse-symmetry example ratio decreases.*

We focus on the metric P_{MAX} as in Observation 3. This observation applies to most cases, except for Dist-based PINEAT-1/4 on FMNIST. We believe that the special case is related to the characteristics of adversarial examples generated in Dist-based methods. The Dist-based methods tend to add noise to a small number of pixels in the image and add extremely negligible noise on the remaining pixels. In the Dist-based PINEAT-1/4 on FMNIST, the adversarial/reverse-symmetry example ratio is very small and the multiple reverse-symmetry examples are very similar to the natural examples. That means the training process is similar to natural training, but the privacy result is still better than that of natural model. In addition, all variants perform better than the underlying adversarial training method and most (55.9%) variants are even better than the natural training method.

Observation 9. *The runtime decreases as the adversarial/reverse-symmetry example ratio decreases.*

This observation can be inferred from Observation 6. In terms of runtime, our methods are superior to the underlying adversarial training method.

6 Conclusion

In this paper, we propose a framework based on empirical adversarial training, named PINEAT. With the design of reverse-symmetry examples, our framework can eliminate the negative impact of adversarial training on the model privacy while retaining its robustness advantage. Our experimental results show that the basic framework and its variants are better than the traditional adversarial training methods in terms of the overall performances.

Acknowledgements. This research was supported in part by the National Key R&D Program of China under grant No. 2022YFB3102100, the National Natural Science Foundation of China under grants No. 62076187, 62172303, the Key R&D Program of Hubei Province under grant No. 2022BAA039, and Key R&D Program of Shandong Province under grant No. 2022CXPT055.

References

1. Andor, D., et al.: Globally normalized transition-based neural networks. In: ACL (2016). https://doi.org/10.18653/v1/p16-1231
2. Carlini, N., Liu, C., Erlingsson, Ú., Kos, J., Song, D.: The secret sharer: evaluating and testing unintended memorization in neural networks. In: USENIX Security Symposium (2019)
3. Carlini, N., Wagner, D.A.: Towards evaluating the robustness of neural networks. In: S&P (2017). https://doi.org/10.1109/SP.2017.49
4. Deng, L., Hinton, G.E., Kingsbury, B.: New types of deep neural network learning for speech recognition and related applications: an overview. In: ICASSP (2013). https://doi.org/10.1109/ICASSP.2013.6639344
5. Gowal, S., et al.: Scalable verified training for provably robust image classification. In: ICCV (2019). https://doi.org/10.1109/ICCV.2019.00494
6. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Master's thesis, University of Toronto (2009)
7. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS (2012)
8. Kurakin, A., Goodfellow, I.J., Bengio, S.: Adversarial machine learning at scale. CoRR (2016). https://arxiv.org/abs/1611.01236
9. Leino, K., Fredrikson, M.: Stolen memories: leveraging model memorization for calibrated white-box membership inference. In: USENIX Security Symposium (2020). https://www.usenix.org/conference/usenixsecurity20/presentation/leino
10. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: ICLR (2018)
11. Mirman, M., Gehr, T., Vechev, M.T.: Differentiable abstract interpretation for provably robust neural networks. In: ICML (2018)
12. Papernot, N., McDaniel, P.D., Jha, S., Fredrikson, M., Celik, Z.B., Swami, A.: The limitations of deep learning in adversarial settings. In: EuroS&P (2016). https://doi.org/10.1109/EuroSP.2016.36
13. Pearson, K.: LIII. on lines and planes of closest fit to systems of points in space. London Edinburgh Dublin Philos. Mag. J. Sci. (1901). https://doi.org/10.1080/14786440109462720
14. Salem, A., Zhang, Y., Humbert, M., Berrang, P., Fritz, M., Backes, M.: ML-leaks: model and data independent membership inference attacks and defenses on machine learning models. In: NDSS (2019)
15. Sharif, M., Bhagavatula, S., Bauer, L., Reiter, M.K.: Accessorize to a crime: real and stealthy attacks on state-of-the-art face recognition. In: CCS (2016). https://doi.org/10.1145/2976749.2978392
16. Shokri, R., Stronati, M., Song, C., Shmatikov, V.: Membership inference attacks against machine learning models. In: S&P (2017). https://doi.org/10.1109/sp.2017.41
17. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015). https://arxiv.org/abs/1409.1556
18. Sinha, A., Namkoong, H., Duchi, J.C.: Certifying some distributional robustness with principled adversarial training. In: ICLR (2018)
19. Song, L., Mittal, P.: Systematic evaluation of privacy risks of machine learning models. In: USENIX Security Symposium (2021)
20. Song, L., Shokri, R., Mittal, P.: Privacy risks of securing machine learning models against adversarial examples. In: CCS (2019). https://doi.org/10.1145/3319535.3354211

21. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. CoRR (2017). https://arxiv.org/abs/1708.07747
22. Yeom, S., Giacomelli, I., Fredrikson, M., Jha, S.: Privacy risk in machine learning: analyzing the connection to overfitting. In: IEEE CSF (2018). https://doi.org/10.1109/CSF.2018.00027
23. Zhang, H., Yu, Y., Jiao, J., Xing, E.P., Ghaoui, L.E., Jordan, M.I.: Theoretically principled trade-off between robustness and accuracy. In: ICML (2019). https://proceedings.mlr.press/v97/zhang19p.html