Pre-training Robust Feature Extractor Against Clean-label Data Poisoning Attacks

Ting Zhou[*]
Shandong University
ting.zhou@mail.sdu.edu.cn

Hanshu Yan[*]
ByteDance
hanshu.yan@bytedance.com

Lei Liu
Shandong University
l.liu@sdu.edu.cn

Jingfeng Zhang
RIKEN-AIP
jingfeng.zhang@riken.jp

Bo Han
Hong Kong Baptist University
bhanml@comp.hkbu.edu.hk

Abstract

In the transfer learning paradigm, models pre-trained on large datasets are employed as foundation models in various downstream tasks. However, this paradigm exposes downstream practitioners to data poisoning threats. Poisoning attackers craft malicious samples on foundation models, then inject these samples into re-training datasets to manipulate the behaviors of models at inference. In this work, we propose an upstream defense strategy that significantly reduces the success rate of various data poisoning attacks. Our defense aims to pre-train robust foundation models by reducing adversarial feature distance and increasing inter-categories feature distance. Experiments demonstrate the excellent defense performance of the proposed strategy towards state-of-the-art clean-label attacks in the transfer learning setting.

1 Introduction

Deep neural networks (DNNs) currently achieve state-of-the-art performance in real-world applications. The impressive success of DNNs is highly dependent on massive amounts of data and computing resources. However, in some special fields like bioinformatics and robotics, data acquisition is expensive, and data annotation is time-consuming and labor-intensive. To obtain performant models with limited resources, practitioners often turn to low-cost training methods, e.g., *transfer learning* methods. Transfer learning starts with a model pre-trained on a large dataset, and then refines this model for downstream tasks.

Nowadays, many large datasets are scraped from data on the internet or users' publicly provided data. Models pre-trained on such datasets are vulnerable to data poisoning attacks, resulting in security risks for downstream users. In clean-label poisoning attacks [14, 21, 1] under transfer learning scenarios, attackers craft poison samples on the pre-trained feature extractor by adding human-imperceptible perturbations. Then poison samples will be injected into the re-training dataset with the intention of manipulating the behavior of the system at inference time. As shown in Figure 1(d), when downstream practitioners re-train the network with a poisoned dataset, they will obtain a poisoned model for misclassifying test samples.

In the transfer learning scenario, existing defense mechanisms against data poisoning mainly focus on the model re-training stage. For example, poison filter defenses [12, 5] filter the poisoned samples before model re-training, robust training defense [3] re-trains the model by data augmentation with

[*] Equal contribution

ML Safety Workshop, 36th Conference on Neural Information Processing Systems (NeurIPS 2022).

crafted proxy poison samples. However, such methods will be performed multiple times with different downstream tasks. That means defenses are repeated numerous times on the same foundation model. Intuitively, multiple downstream defenses increase costs, and it is unrealistic to urge all downstream users to master poisoning attack and defense knowledge.

To achieve upstream defenses, we propose a robust pre-training strategy. The defense manipulates the feature distribution of the pre-trained model through the following two points. One is to increase the inter-categories feature distance, and the other is to simulate poison samples with adversarial samples to reduce the feature distance between poison samples and clean samples. As illustrated in Figure 1(b), in experiments, we pre-train robust feature extractors with Adversarial Training [10] and Prototype Conformity Loss [11] to prevent poisoning attacks. Experiments show that our defense strategy can successfully decrease the attack success rate.

In summary, this work has two contributions: 1) We propose an upstream defense strategy on transfer learning by manipulating the feature distribution of pre-trained feature extractors, which effectively enhances the robustness of models against clean-label poisoning attacks. 2) We improve the loss function in adversarial training to defend against clean-label poisoning attacks, which builds a bridge connecting adversarial robustness and poisoning robustness.

2 Related Work

Data poisoning is an attack where attackers maliciously modify the training data to degrade the test performance of machine learning models. Different from evasion attacks [2, 4, 15], poisoning attacks manipulate the model by adding perturbed samples to training sets instead of controlling model inputs at inference time. In this paper, we focus on targeted clean-label data poisoning [14, 21, 1].

There are two main types of defense against clean-label data poisoning: poisoned data filter defenses [12, 5] and robust training defense [3]. To filter poisoned samples, Peri et al. (2020) [12] detected poisoned data by deep KNN to compare the class label of poison with its k neighbors in feature space. They also realized adversarial pre-training defense as a baseline in their paper. Another filtering method detected poisons by scores re-training samples with cosine similarity influence estimator [5]. From a robust training perspective, Geiping et al. (2021) [3] proposed a robust training framework that trained networks with adversarially poisoned data in the place of (test-time) adversarial examples.

For clean-label poisoning attacks under transfer learning scenarios, previous works are all defenses after pre-training. Although Peri et al. (2020) [12] first tried to train robust feature extractors adversarially to defend against clean-label data poisoning, they failed to provide a theoretical explanation for the defensive effects from the perspective of poisoning attacks. In this work, we present a general defense strategy in pre-training and verify our strategy through experimental instances.

3 Method

3.1 Defense Strategy

Under the Feature Collision Attack [14], an attacker first selects a base sample x_b, and tries to craft a poison x_p by adding imperceptible perturbations. The perturbations are designed to make x_p as same as target x_t in the feature space. To generate a poison, an attacker has to solve the following minimization optimization:

$$x_p = \arg\min_x ||f(x) - f(x_t)||^2 \qquad \textbf{s.t.}\ ||x - x_b||_\infty \leq \delta. \tag{1}$$

Where δ is the perturbation constraint, and f is the fixed pre-trained feature extractor denoting the function that propagates input images through the network to the penultimate layer. By introducing the parameter μ, the ℓ_∞-norm constraints of Eq.(1) can be relaxed to:

$$x_p = \arg\min_x ||f(x) - f(x_t)||^2 + \mu||x - x_b||^2. \tag{2}$$

The parameter $\mu > 0$ makes a trade-off between the two terms. Obviously, the attackers aim to lead x_p and x_t to collide in the feature space.

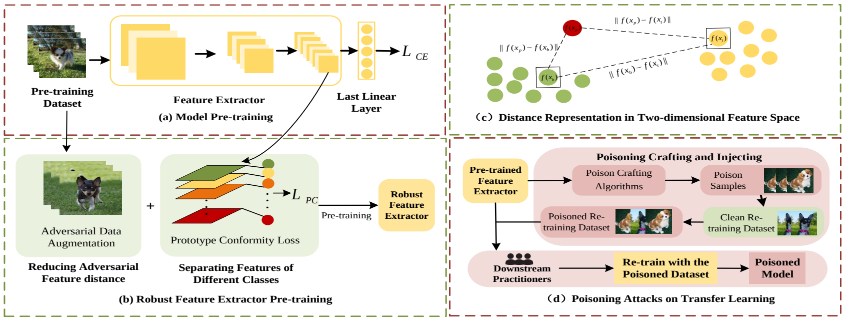
Pre-training Dataset
Feature Extractor
Last Linear Layer
L_{CE}
(a) Model Pre-training
$\| f(x_p) - f(x_t) \|$
$f(x_p)$
$\| f(x_p) - f(x_b) \|$
$f(x_t)$
$f(x_b)$
$\| f(x_b) - f(x_t) \|$
(c) Distance Representation in Two-dimensional Feature Space
Adversarial Data Augmentation
+
Prototype Conformity Loss
L_{PC}
Pre-training
Robust Feature Extractor
Reducing Adversarial Feature distance
Separating Features of Different Classes
(b) Robust Feature Extractor Pre-training
Pre-trained Feature Extractor
Poisoning Crafting and Injecting
Poison Crafting Algorithms
Poison Samples
Poisoned Re-training Dataset
Clean Re-training Dataset
Downstream Practitioners
Re-train with the Poisoned Dataset
Poisoned Model
(d) Poisoning Attacks on Transfer Learning

Figure 1: Robust feature extractor pre-training and poisoning attacks. (a) Normal pre-training. (b) Pre-training robust feature extractor with our defense strategy. (c) Simple feature representation. The lines between samples represent the sample distance in the feature space, which conforms to the triangular inequality. (d) Poisoning attacks under transfer learning scenarios.

To make the model robust to attacks, our defense strategy in reverse with poison optimization to solving $\max ||f(x_p) - f(x_t)||$. In this way, solving Eq.(2) becomes difficult. However, x_p and x_b are unknown in the model pre-training. Thus, we turn to optimize the lower bound of $||f(x_p) - f(x_t)||$. Based on the triangle inequality, we get the following:

$$||f(x_p) - f(x_t)|| \geq ||f(x_b) - f(x_t)|| - ||f(x_p) - f(x_b)||. \tag{3}$$

From Eq.(3), we depart the optimization $||f(x_p) - f(x_t)||$ into two terms. The term $||f(x_b) - f(x_t)||$ is the feature distance between x_b and x_t. The term $||f(x_p) - f(x_b)||$ is the feature distance between x_p and x_b. In Figure 1(c), we provide a 2D feature visualization of Eq.(3). Then, our proposed strategy formulate the maximization lower bound of $||f(x_p) - f(x_t)||$ as the following two optimizations:

$$\max ||f(x_b) - f(x_t)|| \tag{4a}$$

$$\min ||f(x_p) - f(x_b)||. \tag{4b}$$

3.2 Realization of Defense Strategy

Maximize the Class Feature Distance When defending in pre-training, a defender is unaware of the attacker's choice of x_b and x_t. As x_b and x_t are selected from different classes, we solve Eq.(4a) by separating features of different classes. There are many class feature separation methods, such as triplet loss [13] and variants of softmax [9, 8]. Here, we adopt prototype conformity loss [11].

Prototype Conformity Loss (PC-Loss) [11] is an adversarial defense method that forces the features for each class to lie inside a convex polytope and maximally separates the polytope from the polytopes of other classes. Given a dataset $\mathbb{D} = \{(x_i, y_i)\}_{i=1}^n$ and a network with parameters θ, the prototype conformity loss is formulated as:

$$L_{PC}(x, y) = \sum_i \{||f(x_i) - w_{y_i}^c||_2 - \frac{1}{k-1} \sum_{j \neq y_i} (||f(x_i) - w_j^c||_2 + ||w_{y_i}^c - w_j^c||_2)\}. \tag{5}$$

Where $\omega_{y_i}^c$ denotes the trainable class centroids of label y_i and k is the number of classes, and $f(x_i)$ denotes the feature of an image x_i with the label y_i. The overall loss function used for training our feature extractor is given by:

$$L(x, y) = (1 - \alpha) L_{CE}(x, y) + \alpha L_{PC}(x, y). \tag{6}$$

Where L_{CE} is the cross-entropy loss and $0 < \alpha < 1$ makes a trade-off between the two losses.

Adversarial Data Augmentation Considering x_p is unknown at the pre-training stage, we pre-train models with the adversarial sample to simulate x_p, and the corresponding clean sample represent x_b. Let L denote the loss function, x_{adv} is an adversarial example of the original data x_i with ℓ_∞-norm, and ϵ is the constraint budget. Generally, ℓ_∞-norm is considered to be a ℓ_∞-ball centered at x_i. Here, x_{adv} is in the ℓ_∞-ball of x_i and x_p is in the ℓ_∞-ball of x_b. We utilize x_{adv} to simulate x_p because of their similarity. The minimization problem Eq.(4b) can be truned into a min-max problem:

$$\min_\theta \frac{1}{n} \sum_{i=1}^n \left\{ \max_{||x_{adv} - x_i||_\infty \leq \epsilon} L_{CE}(x_{adv}, y_i) + ||f(x_{adv}) - f(x_i)|| \right\}. \tag{7}$$

The feature distance in the loss function encourages x_{adv} to stay close to x_i in the feature space. In this way, the feature representation of x_p is encouraged to be close to that of x_b. When trainingfeature extractor with adversarial samples and joint PC-CE loss, we calculate L_{CE} with adversarial data and calculate L_{PC} with clean data as follows:

$$L(x, y) = (1 - \alpha)\{L_{CE}(x_{adv}, y) + ||f(x_{adv}) - f(x_i)||\} + \alpha L_{PC}(x, y). \tag{8}$$

In this work, we solve the min-max optimization with the adversarial training(AT) method. Adversarial training [10] is a defense against test-time attacks. Compared with the normal training [17], AT effectively enhances the robustness of neural networks [18]. From various adversarial training methods [4, 7, 16, 19, 20], we adopt the classical projected-gradient-descent (PGD) [10].

Table 1: Defense against targeted clean-label poisoning attacks. We report the Attack Success Rate(%) of FC, CP, and BP and the test accuracy(%) of pretraining and retraining on the CIFAR-10 test set. The best defense performance in each column is in bold.

	Attack		FC		CP		BP	
Network	Defense	Pre. Acc	ASR	Re. Acc.	ASR	Re. Acc.	ASR	Re. Acc.
ResNet-18	None	94.42	100.0	92.02±0.32	100.0	91.34±0.28	100.0	91.51±0.30
	PCL	94.83	47.5	92.89±0.26	75.0	92.68±0.21	82.5	92.87±0.23
	AT	90.15	12.5	86.98±0.24	82.5	86.82±0.25	90.0	86.92±0.25
	AT-PCL	90.12	**7.5**	87.93±0.15	**72.5**	87.73±0.23	**67.5**	87.73±0.23
ResNet-50	None	94.60	100.0	91.84±0.41	100.0	92.00±0.58	100.0	91.23±0.45
	PCL	94.85	7.5	94.05±0.23	27.5	93.84±0.24	60.0	93.82±0.20
	AT	90.44	5.0	87.89±0.32	82.5	87.65±0.31	87.5	87.61±0.34
	AT-PCL	90.34	**0.0**	89.31±0.23	**27.5**	89.00±0.33	**37.5**	88.98±0.30

4 Experiments

We evaluate the effectiveness of our proposed defense against three targeted clean-label poisoning attacks: Feature Collision (FC) [14], Convex Polytope (CP) [21], and Bullseye Polytope (BP) [1]. Similar to the experimental setting in [21], our evaluation follows the pre-train then fine-tune paradigm on CIFAR-10 [6]. The dataset splitting follows Zhu et al.(2019) [21]. As attackers, we employ the white-box targeted attack, using the same frozen feature extractor to craft poisons and re-train. The poison constraint $\delta = 25.5/255$. we randomly select four pairs of <target class, poison class>, each pair with ten targets. For all attacks, we measure Attack Success Rate(ASR) over 40 attack instances to evaluate the defense effectiveness.

To evaluate the defense effectiveness, we pre-train the model with the proposed strategy. When pre-training models without any defense, we standard pre-train models on CIFAR-10 for 120 epochs. To realize Eq.(7), we adversarially pre-train models with PGD-10 ($\epsilon = 4/255$) for 200 epochs. For training with PC-Loss, We first train models for 100 epochs with L_{CE} and then use the loss in Eq.(6) or Eq.(8) for 120 epochs. Here, we set α to be 0.3. After pre-training, we use the feature extractors to craft poisons, then inject perturbed images into the re-training dataset to poison the model.

Results We compare the defense effect on models pre-trained with AT, PC-Loss(PCL), and AT joint PC-Loss(AT-PCL). From Table 1, we observe that both AT and PC-Loss reduce the attack success rate, and AT-PCL performs better than employing AT or PC-Loss alone. We also evaluate the test accuracy of pre-training and re-training on the CIFAR-10 test set. We find that AT defense hurts test accuracy, which is common for adversarially trained networks. In transfer learning scenarios, our defense provides effective training strategy options to pre-train a robust foundation model against targeted clean-label poisoning.

5 Conclusion

In this work, we propose an upstream defense strategy against targeted clean-label poisoning attacks in transfer scenarios. We realize the proposed strategy through pre-training feature extractors with adversarial data augmentation and inter-class feature separation. Empirical results demonstrate the effectiveness of the proposed defense in enhancing DNNs' robustness against poisoning attacks.

References

[1] Hojjat Aghakhani, Dongyu Meng, Yu-Xiang Wang, Christopher Kruegel, and Giovanni Vigna. Bullseye polytope: A scalable clean-label poisoning attack with improved transferability. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 159–178. IEEE, 2021.

[2] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 387–402. Springer, 2013.

[3] Jonas Geiping, Liam H Fowl, Gowthami Somepalli, Micah Goldblum, Michael Moeller, and Tom Goldstein. What doesn't kill you makes you robust (er): How to adversarially train against data poisoning. 2021.

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

[5] Zayd Hammoudeh and Daniel Lowd. Simple, attack-agnostic defense against targeted training set attacks using cosine similarity. In *ICML Workshop on Uncertainty and Robustness in Deep Learning*, 2021.

[6] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[7] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *Artificial intelligence safety and security*, pages 99–112. Chapman and Hall/CRC, 2018.

[8] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 212–220, 2017.

[9] Weiyang Liu, Yandong Wen, Zhiding Yu, and Meng Yang. Large-margin softmax loss for convolutional neural networks. *arXiv preprint arXiv:1612.02295*, 2016.

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

[11] Aamir Mustafa, Salman Khan, Munawar Hayat, Roland Goecke, Jianbing Shen, and Ling Shao. Adversarial defense by restricting the hidden space of deep neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3385–3394, 2019.

[12] Neehar Peri, Neal Gupta, W Ronny Huang, Liam Fowl, Chen Zhu, Soheil Feizi, Tom Goldstein, and John P Dickerson. Deep k-nn defense against clean-label data poisoning attacks. In *European Conference on Computer Vision*, pages 55–70. Springer, 2020.

[13] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[14] Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. *Advances in neural information processing systems*, 31, 2018.

[15] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[16] Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. Spatially transformed adversarial examples. *arXiv preprint arXiv:1801.02612*, 2018.

[17] Hanshu Yan, Jiawei Du, Vincent YF Tan, and Jiashi Feng. On robustness of neural ordinary differential equations. *arXiv preprint arXiv:1910.05513*, 2019.

[18] Hanshu Yan, Jingfeng Zhang, Gang Niu, Jiashi Feng, Vincent Tan, and Masashi Sugiyama. Cifs: Improving adversarial robustness of cnns via channel-wise importance-based feature selection. In *International Conference on Machine Learning*, pages 11693–11703. PMLR, 2021.

[19] Jingfeng Zhang, Xilie Xu, Bo Han, Gang Niu, Lizhen Cui, Masashi Sugiyama, and Mohan Kankanhalli. Attacks which do not kill training make adversarial learning stronger. In *International conference on machine learning*, pages 11278–11287. PMLR, 2020.

[20] Jingfeng Zhang, Jianing Zhu, Gang Niu, Bo Han, Masashi Sugiyama, and Mohan Kankanhalli. Geometry-aware instance-reweighted adversarial training. *arXiv preprint arXiv:2010.01736*, 2020.

[21] Chen Zhu, W Ronny Huang, Hengduo Li, Gavin Taylor, Christoph Studer, and Tom Goldstein. Transferable clean-label poisoning attacks on deep neural nets. In *International Conference on Machine Learning*, pages 7614–7623. PMLR, 2019.

Table 2: We report the Attack Success Rate(%) of FC, CP, and BP on feature extractors pre-trained on Tiny ImageNet, and we also report the test accuracy(%) of pre-training and re-training. The best defense performance in each column is in bold.

Attack			FC		CP		BP	
Network	Defense	Pre. Acc	ASR	Re. Acc.	ASR	Re. Acc	ASR	Re. Acc
ResNet-18	None	66.4	100.0	67.17±0.35	100.0	66.99±0.32	100.0	66.78±0.29
	PCL	62.80	12.5	61.73±0.46	37.5	61.78±0.33	42.5	61.74±0.38
	AT	54.78	2.5	61.59±0.36	55	61.55±0.31	60	61.64±0.25
	AT-PCL	54.28	**2.5**	59.80±0.25	**32.5**	59.76±0.19	**35**	59.80±0.20

A Experimental Details

Dataset Splitting On the CIFAR-10 dataset, we take the first 4800 images in each training set class to form a pre-training dataset with 48000 images. For the remaining 200 images of each class in the training set, we choose the first 50 images to form a clean re-training dataset, and the other 150 images are used as the selecting pool of base images.

Training Settings When pre-training models without any defense, we standard pre-train models on CIFAR-10 with SGD for 120 epochs. A batch size of 128 and a learning rate of 0.1 (×0.1 at epochs 80 and 100) are used. For AT, we adversarially pre-train models with PGD-10 ($\epsilon = 4/255$) for 200 epochs. A batch size of 128 and a learning rate of 0.1 (×0.1 at epochs 90, 120, and 150) are used. For pre-training with PC-Loss, models are trained with SGD for 220 epochs, starting with a learning rate of 0.1, which decays by a factor of 10 after epochs 80, 120, 160, and 200. We first train models for 100 epochs with L_{CE} and then use loss in Eq.(6) or Eq.(8). We utilize Eq.(6) for standard training and Eq.(8) for adversarial training.

When re-training models on the poisoned dataset, we only fine-tune the final linear classifier for 100 epochs. We use Adam with a learning rate of 0.1 (×0.1 at epochs 60 and 80) to overfit.

Attack Settings and Evaluation We use the same frozen feature extractors to attack and evaluate in white-box scenarios. For FC, we generate one poisoning image for each attack. For CP and CP, we generate five poisoned images per attack. We perform 500 iterations on the poison perturbations optimization in each experiment. The target/poison label pairs are randomly selected as <airplane/horse>, <bird/automobile>, <deer/automobile>, and <frog/cat>. For each label pair, we attack ten targets, resulting in 40 attack instances.

B Defense on Properly Transfer Learned Models

To simulate real transfer learning, we pre-train ResNet-18 feature extractors on Tiny ImageNet, which are fine-tuned on CIFAR-10 data. We take the entire training set of Tiny ImageNet as the pre-training dataset. The re-training set includes 5000 images formed from the first 500 images in each class on the CIFAR-10 training set. For standard training and adversarial training, we pre-train models with SGD for 200 epochs. A batch size of 128 and a learning rate of 0.1 (×0.1 at epochs 90, 120, and 150) are used. As for pre-training with L_{CE}, the α is set as 0.5, and the other settings are the same as training on the CIFAR-10. We report the defense results in Table 2. The pre-training test accuracy is measured on the Tiny ImageNet test set, and the re-training test accuracy is evaluated on the CIFAR-10 test set.

C Ablation Studies

Feature Loss in Adversarial Training We explore the impact of feature loss in adversarial training. We adversarially pre-train models with L_{CE} and the loss Eq.(8). The perturbation budget is 8/255. In Table 3, We show that the feature loss in plays a positive role in the defense.

Different Adversarial Budget ϵ We conduct experiments to assess the defense impact of the adversarial budget. We test the FC and BP on ResNet-18 models adversarially pre-trained with

Table 3: Defense effectiveness of AT with L_{CE} (AT$_{L_{CE}}$) and AT with the loss in Eq.(8) (AT). The perturbation budget $\epsilon = 8/255$. We report the Attack Success Rate(%) of FC, CP, and BP.

	CIFAR-10				Tiny ImageNet	
	ResNet18		ResNet50		ResNet18	
Attack	AT$_{L_{CE}}$	AT	AT$_{L_{CE}}$	AT	AT$_{L_{CE}}$	AT
FC	0.0	0.0	0.0	0.0	0.0	0.0
CP	45.0	47.5	37.5	32.5	15.0	12.5
BP	65.0	57.5	47.5	45.0	17.5	12.5

Table 4: Defense evaluation of different adversarial budget ϵ.

Attack		FC		BP	
Adversarial budget	Pre. Acc(%)	ASR(%)	Re. Acc.(%)	ASR(%)	Re. Acc(%)
ϵ = 4/255	90.15	12.5	86.98±0.24	90	86.92±0.25
ϵ = 6/255	87.68	2.5	84.07±0.16	65	83.82±0.21
ϵ = 8/255	85.36	**0.0**	79.85±0.23	**57.5**	79.32±0.28

different budgets ϵ. From Table 4, We observe that a large ϵ has a better defense effect. Unfortunately, the increase of ϵ leads a decay in model pre-training accuracy. To make a trade-off between robustness and performance, we set $\epsilon = 4/255$ at other experiments.

Defense Evaluation of Different α in Joint Loss We investigate the defense effect of different α on models pre-trained with loss in Eq.(6) and Eq.(8). In adversarial training, the perturbation budget is 4/255.

From Table 5, we observe that a larger α has a better defense effect. The results show that feature separation between classes improves the robustness of foundation models. But as α increases, the defense of AT with PC-Loss gradually weakens that of ST with PC-Loss. It may be that AT trains models using adversarial samples, which results in the model learning less class feature separation of clean samples. Note that a large α is not always safe. It will lead the model parameters to pay too much attention to feature separation and make pre-training fail. For downstream practitioners, if their task requires defense against both poisoning and escape attacks, a base model pre-trained with the AT joint feature separation method would be a better option. If the downstream system requires high accuracy and only needs to focus on poisoning attacks, thus practitioners could choose pre-trained models using only the feature separation method.

Table 5: Defense evaluation of different α in joint loss.

Network	α	Training Strategy	Pre. Acc(%)	ASR-FC(%)	ASR-BP(%)
ResNet18	α=0.3	ST	94.83	47.5	82.5
		AT	90.12	7.5	67.5
	α=0.4	ST	94.80	27.5	42.5
		AT	89.78	7.5	65
	α=0.5	ST	94.66	2.5	40.0
		AT	90.14	5.0	55.0
	α=0.6	ST	94.76	0.0	20.0
		AT	90.60	2.5	55.0
	α=0.8	ST	94.56	0.0	12.5
		AT	19.18		
ResNet50	α=0.3	ST	94.85	7.5	60.0
		AT	90.34	0.0	37.5
	α=0.5	ST	94.46	0.0	5.0
		AT	90.60	0.0	5.0
	α=0.8	AT	22.29		
	α=0.9	ST	45.25		