Unsupervised Perturbation based Self-Supervised Adversarial Training

Zhuoyi Wang
University of Texas at Dallas
Richardson, TX, USA
zxw151030@utdallas.edu

Yu Lin
University of Texas at Dallas
Richardson, TX, USA
yxl163430@utdallas.edu

YiFan Li
University of Texas at Dallas
Richardson, TX, USA
yifan.li.1112@gmail.com

Feng Mi
University of Texas at Dallas
Richardson, TX, USA
Feng.Mi@utdallas.edu

Zachary Tian
Texas Academy of Mathematics and Science
Denton, TX, USA
ztian360@gmail.com

Latifur Khan
University of Texas at Dallas
Richardson, TX, USA
lkhan@utdallas.edu

ABSTRACT

Deep neural networks (DNNs) are vulnerable to adversarial attacks. Existing adversarial defense approaches mostly use a large number of labels during the training step to improve the model's robustness. However, the labeling typically requires a lot of resources and is time-consuming, especially when the annotation is hard to generate (e.g., an emergency scene in autopilot). In this paper, we propose an instance-level unsupervised perturbation to replace the supervised class-level adversarial sample in the robust training. The unsupervised perturbation is generated on various transformed views of single input, which aims to make the model confuse the instance-level discrimination of this specific input. We further introduce the contrastive learning based adversarial learning(UPAT), which maximizes the agreement between the transformed instance with its corresponding unsupervised perturbed output, and encourages the model to suppress the vulnerability in the embedding space. We conduct comprehensive experiments on three image benchmarks, and the quantitative results demonstrate that our defense approach consistently outperforms prior state-of-the-art techniques, by improving the defense ability efficiently on various white and black box attacks.

KEYWORDS

Unsupervised Perturbation, Self-supervised learning, Adversarial Training, Deep Neural Network

ACM Reference Format:
Zhuoyi Wang, Yu Lin, YiFan Li, Feng Mi, Zachary Tian, and Latifur Khan. 2018. Unsupervised Perturbation based Self-Supervised Adversarial Training. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/1122445.1122456

Woodstock '18, June 03–05, 2018, Woodstock, NY
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06...$15.00
https://doi.org/10.1145/1122445.1122456

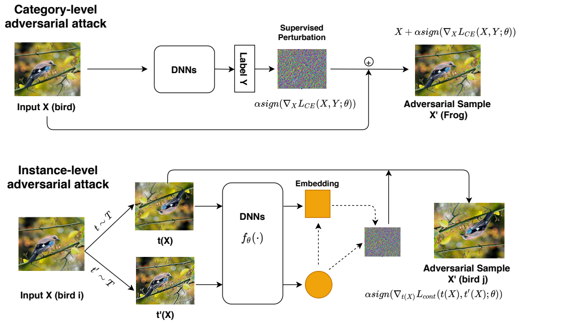

Category-level adversarial attack
Input X (bird)
DNNs
Label Y
Supervised Perturbation
$\alpha sign(\nabla_X L_{CE}(X, Y; \theta))$
$\oplus$
$X + \alpha sign(\nabla_X L_{CE}(X, Y; \theta))$
Adversarial Sample X' (Frog)
Instance-level adversarial attack
Input X (bird i)
$t \sim T$
$t' \sim T$
t(X)
t'(X)
DNNs
$f_\theta(\cdot)$
Embedding
Adversarial Sample X' (bird j)
$\alpha sign(\nabla_{t(X)} L_{cont}(t(X), t'(X); \theta))$

Figure 1: Difference between the class-level and instance-level adversarial attack.

1 INTRODUCTION

Despite outstanding performance on various deep learning/data mining scenarios[4, 11, 13, 16], deep neural networks (DNNs) are susceptible to adversarial attacks [12, 28]. By adding small indistinguishable perturbations on the inputs, the adversarial example for an original image is produced, making the DNNs output a wrong prediction with high probability. This phenomenon is known as a class-level adversarial attack (as shown in the first row of Figure 1), which attracts increasing concerns on the safety-critical applications, such as self-driving cars and speech recognition[5, 21, 25, 27, 29, 30]. Although many existing approaches have been proposed to improve the model robustness against adversarial examples, such as adversarial training [12, 28], Generate Adversarial Network[18]. They require a large amount of labeled data during the training step, which may result in the following challenges: 1) *resource consuming*, the collection of the labeled data and relevant adversarial training are expensive, so they restrict relative further applicability to real-world large-scale datasets [22]; 2) *label leakage*[20] which may cause the model over-fitting on some specific perturbations, and weaken the generalization of the model on other unseen attacks (for example, a model trained on L_∞ and test on L_2). Therefore, such a phenomenon shows the weakness of existing defense approaches.

In this paper, we focus on the label-free setting based adversarial pre-training, which does not require the label to generate the class-level perturbation, for the adversarial training step. We propose a contrastive representation learning (UPAT) framework to enhance the generalization and robustness of the DNN model. The intuition is to generate unsupervised perturbation, an instance-level based adversarial sample, then apply the contrastive learning[6, 8, 15] on both adversary and clean input to reduce the vulnerability in the embedding space. As shown in Figure. 1, the instance-level adversarial attack on a single transformed sample (e.g., x_i) aims to confuse the model's instance-discrimination, and make the model misclassify this sample as another instance (e.g., x_j). The generation of such adversarial sample could be formulated by maximizing a comparison metric (e.g., MSE) between the perturbation and the transformed[10] input. Next, we use the generated perturbation on the adversarial training step to learn robust DNNs, specifically, UPAT, to maximize the agreement between the transformed samples and such perturbations by contrastive learning [8]. Our work aims to suppress the vulnerability and obtain a more robust embedding space, to defend against the adversarial perturbation. The proposed UPAT pipeline could benefit the subsequent fine-tuning operation, such as the adversarial training and input-process [31]. Besides our work, [17] and [7] also apply the self-supervised learning on the adversarial robustness. However, both of them still require labeled data to generate the adversarial samples, and the perturbation under the label-free problem setting is still leaving unexplored.

Our contributions are summarized as follows: (1) We introduce the unsupervised perturbation, which is an instance-level adversarial sample without labeling requirement; (2) we propose the adversarial contrastive representation learning framework to improve the instance-discrimination on clean input and corresponding instance-wise adversarial perturbations. UPAT aims to enhance the model robustness and further reduce the label leakage on unseen attacks; (3) we provide extensive experimental validation of UPAT under the strong and unseen type of the white-box attacks, the results demonstrate the proposed UPAT outperforms the cutting-edge supervised adversarial learning approaches.

2 RELATIVE WORK

2.1 Adversarial Sample

Adversary is typically generated by adding small perturbation to clean input, it aims to make AI system producing erroneous outputs. For the training set $D = \{(x_i, y_i)\}_{i=1}^{N}$ which has total N labeled samples x_i, a supervised learning model f_θ that has a mapping function from input to the corresponding label: $f_\theta(x_i) = y_i$, where θ is the parameters of model f. The adversarial attack allows an adversary to eavesdrop the optimization and gradients of the existing model. For the given clean input x_i to a target model, the adversarial attack δ is required to perturb the input into x with bounded magnitude p-norm ($p = 1, 2, \infty$) as: $f(x+\delta) \neq f(x)$ with $||\delta||_p <= \epsilon$. Here, the strength of the perturbations δ should not be greater than ϵ so that the perturbations remain imperceptible to people's eyes[14]. Such formulation generalizes across different types of gradient attacks, like the PGD [2], which performs the universal first-order adversary, with K iterated step to form the attack:

$$x_i^{K+1} = \prod_B (x_i^k + \alpha sign(\nabla_{x_i} L_{CE}(x_i, y_i; \theta))) \tag{1}$$

2.2 Defense Mechanisms.

Various defense mechanisms have been employed to combat the threat from adversarial attacks. The most common method is Adversarial Training (AT), which is based on augmenting the training dataset with adversarial examples [12, 24, 33]. The main idea is to minimize the loss of such adversarial perturbations, which is often called adversarial learning[22]. They solve the min-max optimization problem that for the adversarial sample $x' \in B$ (or described as adding δ on input x_i with $||\delta||_\infty \leq \epsilon$), the generic form is:

$$\underset{\theta}{\operatorname{argmin}} \mathbb{E}_{(x_i, y_i) \in D} [\max_{x' \in B} \ell(x_i', y_i; \theta)] \tag{2}$$

Recent works focus on improving the robustness of learned embedding space, such as TRADES[32], which applies Kullback-Leibler divergence loss between a clean input and its adversarial version to obtain more robust latent space. We generate the self-supervised instance-wise adversarial sample to improve the model robustness and it is different from the previous class-level based adversarial training.

2.3 Self-Supervised Learning

For now, [9, 17] approaches introduce a self-supervision technique to train a robust embedding, with the rotation prediction or transformation ensemble and co-optimized with the adversarial pre-training. The main advantage of Self-Supervised Learning (SSL) is that it generates supervised learning problems out of unlabeled data, and optimizes a feature representation based on them. Previous SSL encourage the model to solve a pretext task for representation learning, which will be later used for a (down-stream) specific supervised learning task. These tasks usually involve hiding certain information about the input, and trains the model to recover the missing information [26].

In this paper, we apply the recently contrastive learning[8, 15] based SSL that leverages the instance-level identity. This approach mainly applies the contrastive loss to maximize the similarity between different augmentations of the "same" input in the latent space, and minimize the agreement between the "different" inputs, like the SimCLR[8]. Such approaches has shown highly effective on learned embedding.

3 APPROACH

3.1 Unsupervised Perturbation

Different from the class-level adversarial samples that apply the gradient directions specific to the model parameters θ, then create adversarial samples x_{adv} to make the model confuse on class-boundary. We propose the unsupervised perturbation, which is the instance-wise attack that perturbs the representation space without any class label. Specifically, given a sample of an input instance, we generate a perturbation to confuse the model by maximizing the loss on feature embedding from different transforms (on the same input), so the model would make the wrong decision to classify it

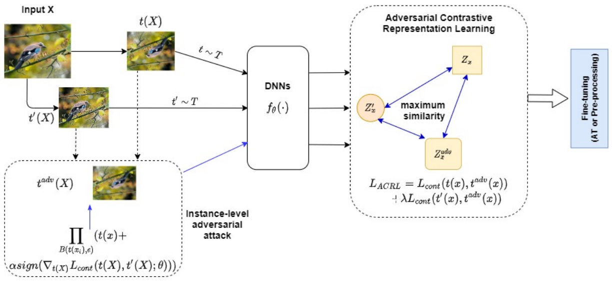


Figure 2: Overview of the UPAT for the pre-training and fine-tuning process.

as another sample. The metric losses such as MSE function could be used for discrimination among the transformed (or perturbed) instances. The perturbation is encouraged to maximize such metric loss on the representation for instance-discrimination, the formula is described as follows:

$$
\begin{aligned}
T(x_i)^{k+1} &= \prod_{B(T(x_i),\epsilon)} [T(x_i)^k + \alpha sign(\nabla_{T(x_i)^k} L(t(x_i), t'(x_i)))] \\
s.t, \qquad & T(x_i) \in \{t(x_i), t'(x_i)\}
\end{aligned} \tag{3}
$$

Where $t(x)$ and $t(x)'$ are the augmented input under the stochastic data transformations $t, t' \in \mathcal{T}$, the α is the step size of the attacks, and L is the MSE loss function. Finally, the instance-level adversarial sample of x_i could be denoted as $x_i^{adv} = T(x_i) + \delta$, and the $T(\cdot)$ indicates one of the augmented operations from $\{t(\cdot), t'(\cdot)\}$. Because of the same identity of different transformed input $t(x)$ and $t'(x)$, both of them are selected to generate instance-level attacks and find optimal perturbation. We summarize our self-supervised learned adversarial sample in Algorithm. 1:

Algorithm 1 Instance-level Unsupervised Perturbation

```
Require: $f_\theta$ - feature extractor; $g$ - projector; $x$ - clean input; $\mathcal{T}$ - input transformation family; $K$ - iteration times; $\epsilon$ - perturbation bound
For $x$, get the augmented output $t(x)$, $t'(x)$, $(t, t' \in \mathcal{T})$
for $k = 1 \rightarrow K$ do
    Forward pass $t(x)$, $t'(x)$ through $f_\theta$ and $g$, get $z, z'$
    Select one type of augmented $T \in \{t, t'\}$ as $T(x)$
    Compute gradient $g^k = \nabla_{T_x} L(z, z')$.
    Perturbation: $T(x)^{k+1} = T(x)^k - \alpha sign(g^k)$
    Projection: $x^{adv} = clip(T(x)^{k+1}, x - \epsilon, x + \epsilon)$
end for
return $x_{adv}$
```

3.2 Contrastive Representation Leaning for Robustness

Contrastive Leaning applies the stochastic data augmentation t that randomly obtains from an augmentation family $\mathcal{T}$ ($\mathcal{T}$ like random cropping, random color distortion). It selects two transformations $t, t' \in \mathcal{T}$ and augments the input sample x_i into $t(x_i), t'(x_i)$, where these two samples retain the instance-level discrimination of the same input. Finally, a feature encoder $f_\theta(\cdot)$ is applied with a projector $g(\cdot)$(a two-layer perceptron) to map the (augmented) input x_i into a 128-dimensional latent vector $z_i : z_i = g(f_\theta(t(x_i)))$. For N examples with $2N$ augmented points, the self-supervised contrastive loss could be defined as:

$$
L_{Cont} = \sum_{i=1}^{2N} L_{Cont}^i = -\log \frac{\exp[sim(z_i, z_i')/\tau]}{\sum_{k=1}^{2N} \mathbb{1}_{k \neq i} \exp[sim(z_i, z_k)/\tau]} \tag{4}
$$

Where the $\mathbb{1}_{k \neq i}$ is an indicator of note whether $i = k$ or not (if so, returns 1; else, return 0), and the τ is a temperature parameter ($\tau = 0.5$). For the similarity measurement $sim(i, j)$, the cosine similarity ($sim(a, b) = \frac{a^\top b}{||a|| \, ||b||}$) is the typically used score between normalized instance pair.

Based on the contrastive learning and adversarial sample, we describe how to improve the robustness of embedding space via our UPAT approach. This adversarial learning framework is similar to the supervised adversarial learning method [22] in Eq. 5. Since our approach focuses on unlabeled data scenario, the cross-entropy loss based class-level training is not suitable in our approach. The min-max formulation of instance-level based adversarial training could be described as:

$$
\underset{\theta}{\operatorname{argmin}} \, \mathbb{E}_{(x) \in D} [\max_{||\delta||_\infty \leq \epsilon} L_{Cont}(t'(x) + \delta, t(x); \theta)] \tag{5}
$$

The adversarial samples are generated through instance-level attacks through Eq. 3. Finally, we organize the contrastive objective function to maximize the similarity between transformed examples and their instance-wise perturbation in the embedding space, which suppress the vulnerability and improve the adversarial robustness. Unlike the existing approach in Eq. 4, we apply the instance-level adversarial examples as additional elements in the positive set, and formulate our contrastive adversarial training objective in the following ways:

$$
\begin{aligned}
L_{UPAT}(t(x_i), t'(x_i), x_i^{adv}) &= L_{cont}(t(x_i), x_i^{adv}) + \lambda L_{cont}(t'(x_i), x_i^{adv}) \\
= -\log \frac{\exp[sim(z_i, z_i^{adv})/\tau]}{\sum_{k \neq i} \exp[sim(z_k, z^{adv})/\tau]} &- \lambda \log \frac{\exp[sim(z_i', z_i^{adv})/\tau]}{\sum_{k \neq i} \exp[sim(z_k, z^{adv})/\tau]}
\end{aligned} \tag{6}
$$

Where the x_i^{adv} is the adversarial perturbation of an augmented sample $t(x)$ or $t'(x)$, the UPAT loss is regarded as regularization on the contrastive between adversarial examples and transformed clean samples, such instances act with the same instance-level identity.

3.3 Fine-Tuning (FT) for Robustness Evaluation

The UPAT adversarially trains the model without the requirement of data annotation and it is a self-supervised pre-training process that works on a representation space. In order to evaluate the learned embedding for down-stream class-level classification, it is necessary to leverage a linear layer $f_l(\cdot)$ on top of the fixed layer $g(f_\theta(\cdot))$, and fine-tune it with various training strategies. Here we major consider the adversarial training (AT) strategy on the fine-tuning step, which trains a linear classifier with class-level adversarial samples for the specific attack (like ℓ_∞, ℓ_2), we follow the Eq. 5 that only optimizes the parameters of the linear model $f_l(\cdot)$. Also, other robust training methods like pre-pocessing[31] could also be applied here.

4 EXPERIMENT

4.1 Experimental Setup:

We implement UPAT through Pytorch[1], and all the experiments are done on a single Nvidia RTX 2080 GPU. We choose the ResNet-18 [16] as the network backbone without any fully connected layers, our experiments are executed on two common datasets: **CIFAR10**, **CIFAR100**[2], all images are normalized into [0, 1] and the output dimension from projector $g(\cdot)$ is set to 128. In the adversarial contrastive pre-training, we use the same data transformation setting as discussed in SimCLR[8], which includes color distortions and random gaussian blur. For the parameters setting during this step, we set the batch size as 256, and apply the standard stochastic gradient descent (SGD) with momentum to 0.9, weight decay to 0.001, $\lambda = 1/256$ for 300 epochs and perturbation $\epsilon = 0.03$ with step size $\alpha = 2/256$ in Eq. 3. Next for the evaluation in the fine-tuning step, we choose the AT on the linear layer after the frozen feature extractor $g(f_\theta)$, we set 100 epochs with the learning rate of 0.5, and the learning rate would drop by a factor of 10 at 50 epoch. The PGD attack would be used to generate class-level adversarial examples for the AT, which performs ℓ_∞ attack with $\alpha = 8/256$ in 10 steps.

For each attack type, we compare UPAT with state-of-the-art defense techniques based on supervised and self-supervised training: the common adversarial training **AT** [22], Adversarial Logit Pairing **ALP** [19], and **TRADES**[32][3]. For the self-supervised approaches, we compare our method with the SimCLR fine-tuning with adversarial training **SimCLR-AT**, **SS-OOD**[17][4] and **SAT** [7]. We also evaluate our model on diverse scenarios: linear classifier for fine-tuning **UPAT-Linear**, adversarial training for fine-tuning **UPAT-AT**. For each competitor, we report the accuracy and adversarial accuracy as the percentage of adversarial points that are correctly classified.

4.2 White-box Attacks:

To evaluate defensive ability against the white box attacks, we compare diverse scenarios of UPAT with other supervised/self-supervised learning-based defense methods, under different attacking strategies (like $\epsilon = 16/255$, and more steps in the PGD with $\epsilon = 8/255$). We choose the adversarial(robust) accuracy as the evaluate metric, we test the model under PGD attacks with 20, and 100 steps, then set the step-size $\alpha = 2/256$ for 20, and $\alpha = 0.3/256$ for 100 as same as[22]. Table. 1 shows a comparison of the performance on other competitors against the different white-box attacks. Results show that although all the vanilla model is extremely vulnerable to adversarial attacks, the performance of UPAT-AT is better than most existing approaches against the ℓ_∞ attack. Specifically, we observe that after fine-tuning on down-stream tasks (e.g., AT), our approach could even outperform a supervised adversarial learning-based approach, like the AT and ALP, and also obtain comparable performance to the best one TRADES. A similar phenomenon could also be observed in other self-supervised learning approaches. This significant observation shows that the self-supervised pre-training could improve the model's robustness on various PGD step attacks, through enhancing the instance discrimination in the embedding space. Finally, our approach could get better performance compared with other self-supervised methods. For example, our work improves the empirical state-of-the-art robust accuracy around 1.67 - 2.84% under PGD-20 attack with different ϵ value, on CIFAR-10 dataset. It shows that the contrastive learning between the instance-level perturbation and transformed data could suppress the distortion in the embedding space to ensure a more robust representation.

4.3 Black-box Attacks:

In this section, we verify the robustness of our models under the black-box (transfer-based) attacks. We use the same network and parameter setting that is specified in the white-box attack, then apply different attacking strategies on the copy network to generate black-box adversarial examples. For both datasets, we use the same attack parameters as in the Madry's model [22]. Here, we set the parameter $\epsilon = 8/255$ and $16/255$ under the PGD (black-box) method, with 20 iterations and 0.003 step size to attack other defense models. We select the adversarial training (AT) and TRADES source as the source models, then compute the perturbation directions according to the source models' gradients on the input to generate adversarial perturbations. The generated adversarial samples are used to test on different defense models. The results are described in Table. 2. We compare the state-of-the-art approaches against UPAT with different attack settings. From the Table. 2, it is obvious that UPAT is superior to the adversarial training approaches against TRADES and AT with a reasonable margin on most of the cases and similar performance in few cases.

4.4 Generalizing Robustness on Unseen Attacks:

Here, we evaluate the generalization of the model that trained over different adversarial learning approaches against unseen types of attacks. We introduce and compare with the recently proposed **robust-union**[23]. The ℓ_∞ is set as the seen attack, ℓ_2 with $\epsilon = 0.5, 1$, ℓ_1 with $\epsilon = 12, 24$ as the unseen attack. We show the robust accuracy on the CIFAR-10 dataset, results are the averages over 5 runs with standard deviations in parenthesis. We show the performance of the generalizing robustness on different white-box attacks and compare it with other supervised or self-supervised learning-based defense approaches. The result from Table. 3 shows that UPAT outperforms most other competitors regarding higher robustness on the unseen types of attacks. More specifically, in case of ℓ_2, when ϵ changes from 0.5 to 1.0, the adversarial accuracy of ACRL-Linear and ACRL-AT do not vary significantly, AT, Robust-Union and TRADES drop with a large margin (e.g., TRADES changes from 58.4% to 47.1%). Furthermore, on the single unseen ℓ_2 attack with $\epsilon = 0.5$, we achieve at least 5.3% robustness improvement compared with the supervised approach (e.g., Robust-Union), and 3.1% higher than the most effective self-supervised approach SAT. This proves that introducing the instance-wise attacks into contrastive learning in the latent embedding space is an effective approach that ensures the general robustness against other attacks.

4.5 Parameter Sensitivity

We select the various values of the parameter λ and test the sensitivity on UPAT-Linear (in Fig.3(a)). It can be seen that the value of λ should be set at a relative low range, as higher λ would cause a slight

[1] https://pytorch.org/

[2] https://www.cs.toronto.edu/ kriz/cifar.html

[3] https://github.com/yaodongyu/TRADES

[4] https://github.com/hendrycks/ss-ood

Table 1: Experimental results with white box attacks on ResNet18 trained on the CIFAR-10 and CIFAR-100, all models are trained under ℓ_∞ attack, we show the average results over totally 5 runs.

Train type	Approach	Acc on CIFAR-10					Acc on CIFAR-100				
		Clean	$\epsilon =$ 8/255	$\epsilon =$ 16/255	PGD20	PGD100	Clean	$\epsilon =$ 8/255	$\epsilon =$ 16/255	PGD20	PGD100
Supervised	L_{ce}	**93.02%**	0.56%	0	0	0	**71.42%**	0	0	0	0
	AT	84.25%	44.55%	15.06%	43.59%	43.02%	53.95%	20.25%	6.11%	20.04%	19.71%
	ALP	85.12%	46.96%	17.12%	46.12%	45.47%	55.82%	22.45%	7.74%	22.94%	22.47%
	TRADES	83.11%	**53.11%**	**23.92%**	**52.40%**	**51.98%**	58.53%	24.58%	**10.51%**	24.31%	**23.90%**
Self Supervised + FT	SimCLR-AT	85.33%	22.74%	12.82%	22.37%	21.84%	51.77%	8.83%	4.26%	8.18%	6.99%
	SS-OOD	83.37%	50.92%	21.14%	50.35%	50.12%	52.69%	25.30%	10.18%	24.93%	24.11%
	SAT	84.51%	51.53%	26.73%	51.17%	51.05%	54.21%	27.05%	10.83%	26.81%	26.47%
	UPAT-Linear	83.82%	45.74%	20.05%	42.56%	40.71%	56.44%	21.64%	6.63%	20.91%	20.21%
	UPAT-AT	81.17%	53.20%	29.57%	52.83%	52.57%	53.04%	29.23%	12.18%	28.77%	27.42%

Table 2: Performance of UPAT against black box attacks on the CIFAR-10, during the experiments includes both SGD and adversarial training.

TargetSource	ϵ = 8/255		ϵ = 16/255	
	AT	*TRADES*	*AT*	*TRADES*
AT	-	77.3±0.3	-	64.2±0.2
ALP	63.6±0.2	78.4±0.3	**45.1±0.3**	67.0±0.2
TRADES	61.2±0.2	-	41.7±0.3	-
UPAT-Linear	68.1±0.2	77.9±0.1	43.1±0.2	65.4±0.2
UPAT-AT	**69.4±0.2**	**79.5±0.2**	44.3±0.2	**67.7±0.1**

Table 3: The results of the generalizing robustness, all models are trained under the ℓ_∞ with $\epsilon = 8/255$. We set the PGD ℓ_2, ℓ_1 attack with different ϵ as the unseen attack.

Methods	ℓ_∞	ℓ_2	ℓ_2	ℓ_1	ℓ_1
	$\epsilon = 8/255$	$\epsilon = 0.5$	$\epsilon = 1.0$	$\epsilon = 12$	$\epsilon = 24$
	Seen	*Unseen*	*Unseen*	*Unseen*	*Unseen*
AT	44.6±0.6	59.1±0.4	46.3±0.5	55.9±0.3	44.2±0.3
Robust-Union	48.7±0.4	**64.5±0.4**	**52.1±0.4**	**56.8±0.3**	45.9±0.4
TRADES	**53.3±0.5**	58.4±0.3	47.1±0.3	55.3±0.2	**46.4±0.3**
SimCLR-AT	7.3±0.4	27.3±0.3	22.1±0.3	25.1±0.2	20.3±0.2
SAT	51.5±0.5	66.7±0.3	63.5±0.2	75.4±0.3	72.2±0.2
UPAT-Linear	45.3±0.4	65.8±0.3	62.2±0.3	75.8±0.3	73.5±0.3
UPAT-AT	53.2±0.6	69.8±0.4	66.4±0.3	77.5±0.2	75.3±0.2

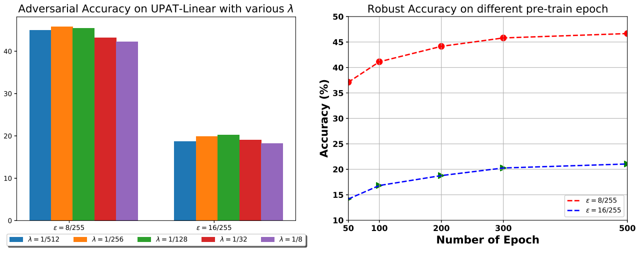
Table 4: Ablation study on the instance-level attack.

$T(\cdot)$ **Selection**	Clean Accuracy (%)	PGD-20 with $\epsilon = 8/255$	PGD-100 with $\epsilon = 8/255$	$\epsilon = 16/255$ with PGD-20
Original x	**84.77**	40.14	38.23	16.62
$t(x)$	83.39	42.03	40.05	**19.24**
$t'(x)$	83.85	**42.56**	**40.71**	19.05

drop of accuracy against different perturbation attacks. We also verify the learning curve of UPAT under increasing epoch number in Fig.3(b), our method could acquires sufficiently stable robust accuracy after around 300 epochs.

4.6 Ablation Study

4.6.1 Metric loss for generating perturbation. As described in Sec. 3.1, various metric loss functions can be used to compute the similarity between two transformed samples $(t(x), t'(x))$ in the embedding space. Here, we compare two different metric losses: the mean square error (MSE), and contrastive loss. Table. 5 shows that the contrastive loss is more effective on robustness accuracy compared to MSE. Therefore, in the experiment, we use the contrastive loss as the metric in attack generation.



(a)

(b)

Figure 3: (a). λ ablation experiment results with white box attacks on UPAT-Linear model. (b)Evaluation of the UPAT-AT performance on white-box attack over different PGD iteration and ϵ value.

Table 5: Ablation study on the metric loss type for attacking.

Metric Loss	Clean	PGD attack with ϵ = 8/255	
	Accuracy (%)	PGD-20 on UPAT-Linear	PGD-100 on UPAT-Linear
Contrastive Loss	83.85±0.31	42.56±0.38	40.71±0.42
MSE Loss	85.39±0.28	42.15±0.32	40.04±0.36

4.6.2 $T(\cdot)$ selection for instance-level attack. To generate the instance-level adversarial sample, we need to decide which identity we will select for a self-generate adversary in Eq. 3. Here, the original input x, the transformed image $t(x)$, and another augmentation of original input $t'(x)$ share the same identity. We evaluate all these three instances in the instance-wise attacks under the UPAT-linear setting. In Table. 4, we observe that the original x is still useful for the self-generate adversary on the clean input classification. However, compared to transform version $(t(x), t'(x))$, x shows relatively low robust accuracy during an adversarial attack. Therefore, the instance-level attack is be of help to ensure stable instance-identity performance in the embedding space, and it is not sensitive to transform strategy.

5 CONCLUSION

To solve the challenge of improving adversarial robustness under the label-free setting, this paper proposes the UPAT that focuses on suppressing the vulnerability in the representation space, and improves the model robustness. We first implement the instance-level unsupervised perturbation, which confuses the model on the instance discrimination of a single input. Then we combine the perturbation with adversarial contrastive training, and maximize the agreement between transformed input with its corresponding adversarial output. We evaluate our method on multiple benchmarks under both seen and unseen white-box attacks, finally obtain superior robustness to the other state-of-the-art approaches. In the future work, we would further apply such defense mechanism into the real-world application, such as information retrieval system [1, 3] and so on.

REFERENCES

[1] Satyen Abrol and Latifur Khan. 2010. Twinner: understanding news queries with geo-content using twitter. In *Proceedings of the 6th Workshop on Geographic information Retrieval*. 1–8.

[2] Anish Athalye, Nicholas Carlini, and David Wagner. 2018. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *ICML* (2018).

[3] Mamoun Awad, Latifur Khan, Farokh Bastani, and I-Ling Yen. 2004. An effective support vector machines (SVMs) performance using hierarchical clustering. In *16th IEEE international conference on tools with artificial intelligence*. 663–667.

[4] Gbadebo Ayoade, Vishal Karande, Latifur Khan, and Kevin Hamlen. 2018. Decentralized IoT data management using blockchain and trusted execution environment. In *IRI*. IEEE, 15–22.

[5] Nicholas Carlini and David Wagner. 2017. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 39–57.

[6] Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C Duchi, and Percy S Liang. 2019. Unlabeled data improves adversarial robustness. In *NeurIPS*. 11192–11203.

[7] Kejiang Chen, Yuefeng Chen, Hang Zhou, Xiaofeng Mao, Yuhong Li, Yuan He, Hui Xue, Weiming Zhang, and Nenghai Yu. 2020. Self-Supervised Adversarial Training. In *ICASSP*. IEEE, 2218–2222.

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. *ICML* (2020).

[9] Tianlong Chen, Sijia Liu, Shiyu Chang, Yu Cheng, Lisa Amini, and Zhangyang Wang. 2020. Adversarial Robustness: From Self-Supervised Pre-Training to Fine-Tuning. In *CVPR*. 699–708.

[10] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. 2019. Autoaugment: Learning augmentation policies from data. *NeurIPS* (2019).

[11] Gianmarco De Francisci Morales, Albert Bifet, Latifur Khan, Joao Gama, and Wei Fan. 2016. Iot big data stream mining. In *KDD*. 2119–2120.

[12] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. *ICLR* (2015).

[13] Ahsanul Haque, Zhuoyi Wang, Swarup Chandra, Yupeng Gao, Latifur Khan, and Charu Aggarwal. 2016. Sampling-based distributed kernel mean matching using spark. In *IEEE International Conference on Big Data (Big Data)*. 462–471.

[14] Jamie Hayes and George Danezis. 2018. Learning universal adversarial perturbations with generative models. In *2018 IEEE Security and Privacy Workshops (SPW)*. 43–49.

[15] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2019. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722* (2019).

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

[17] Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. 2019. Using self-supervised learning can improve model robustness and uncertainty. In *NeurIPS*. 15663–15674.

[18] Guoqing Jin, Shiwei Shen, Dongming Zhang, Feng Dai, and Yongdong Zhang. 2019. Ape-gan: Adversarial perturbation elimination with gan. In *ICASSP*. IEEE, 3842–3846.

[19] Harini Kannan, Alexey Kurakin, and Ian Goodfellow. 2018. Adversarial logit pairing. *arXiv preprint arXiv:1803.06373* (2018).

[20] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. 2017. Adversarial machine learning at scale. *ICLR* (2017).

[21] Bai Li, Changyou Chen, Wenlin Wang, and Lawrence Carin. 2019. Certified adversarial robustness with additive noise. In *Advances in Neural Information Processing Systems*. 9464–9474.

[22] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards deep learning models resistant to adversarial attacks. *ICML* (2018).

[23] Pratyush Maini, Eric Wong, and J Zico Kolter. 2020. Adversarial robustness against the union of multiple perturbation models. *ICML* (2020).

[24] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *CVPR*. 2574–2582.

[25] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. 2017. Practical black-box attacks against machine learning. In *ASIACCS*. ACM, 506–519.

[26] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. 2016. Context encoders: Feature learning by inpainting. In *CVPR*. 2536–2544.

[27] Lea Schönherr, Katharina Kohls, Steffen Zeiler, Thorsten Holz, and Dorothea Kolossa. 2019. Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding. *NDSS* (2019).

[28] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199* (2014).

[29] Thomas Tanay and Lewis Griffin. 2016. A boundary tilting persepective on the phenomenon of adversarial examples. *arXiv preprint arXiv:1608.07690* (2016).

[30] Lei Wang, Li Liu, and Latifur Khan. 2004. Automatic image annotation and retrieval using subspace clustering algorithm. In *Proceedings of the 2nd ACM international workshop on Multimedia databases*. 100–108.

[31] Yuzhe Yang, Guo Zhang, Dina Katabi, and Zhi Xu. 2019. Me-net: Towards effective adversarial robustness with matrix estimation. *ICML* (2019).

[32] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P Xing, Laurent El Ghaoui, and Michael I Jordan. 2019. Theoretically principled trade-off between robustness and accuracy. *ICML* (2019).

[33] Stephan Zheng, Yang Song, Thomas Leung, and Ian Goodfellow. 2016. Improving the robustness of deep neural networks via stability training. In *CVPR*. 4480–4488.