TARGETED ADVERSARIAL SELF-SUPERVISED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

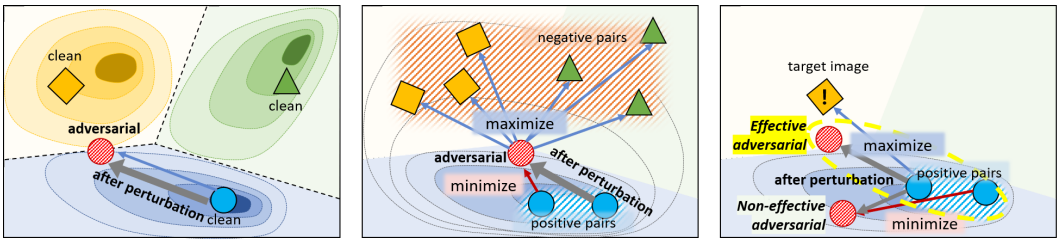
Recently, unsupervised adversarial training (AT) has been extensively studied to attain robustness with the models trained upon unlabeled data. To this end, previous studies have applied existing supervised adversarial training techniques to self-supervised learning (SSL) frameworks. However, all have resorted to untargeted adversarial learning as obtaining targeted adversarial examples is unclear in the SSL setting lacking of label information. In this paper, we propose a novel targeted adversarial training method for the SSL frameworks. Specifically, we propose a target selection algorithm for the adversarial SSL frameworks; it is designed to select the most confusing sample for each given instance based on similarity and entropy, and perturb the given instance toward the selected target sample. Our method significantly enhances the robustness of an SSL model without requiring large batches of images or additional models, unlike existing works aimed at achieving the same goal. Moreover, our method is readily applicable to general SSL frameworks that only uses positive pairs. We validate our method on benchmark datasets, on which it obtains superior robust accuracies, outperforming existing unsupervised adversarial training methods.

1 INTRODUCTION

Enhancing the robustness of deep neural networks (DNN) is a critical challenge for their real-world applications. DNNs have been known to be vulnerable to adversarial attacks using imperceptible perturbations (Goodfellow et al., 2015), corrupted images (Hendrycks & Dietterich, 2019), and images with shifted distributions (Koh et al., 2021), which cause the attacked DNN models to perform incorrect predictions. A vast volume of prior studies has proposed to leverage adversarial training (AT) (Madry et al., 2018); AT explicitly uses generated adversarial examples with specific types of perturbations (e.g., ℓ_∞-norm attack) when training a DNN model. Most of these previous AT studies have considered supervised learning settings (Madry et al., 2018; Zhang et al., 2019; Wu et al., 2020; Wang et al., 2019) in which we can utilize class label information to generate adversarial examples. On the other hand, achieving robustness in a self-supervised learning (SSL) setting has been relatively understudied despite the recent success of SSL in a variety of tasks and domains.

SSL frameworks (Dosovitskiy et al., 2015; Zhang et al., 2016; Tian et al., 2020b; Chen et al., 2020; He et al., 2020; Grill et al., 2020; Chen & He, 2021) have been proposed to learn transferable visual representations by solving for pretext tasks constructed out of the training data (Dosovitskiy et al., 2015; Zhang et al., 2016). A popular SSL approach is contrastive learning (e.g., SimCLR (Chen et al., 2020), MoCo (He et al., 2020)), which learns to maximize the similarity across positive pairs, each of which contains differently augmented samples of the same instance, while minimizing the similarity across different intances. Recently, to establish robustness in these SSL frameworks, RoCL (Kim et al., 2020) and ACL (Jiang et al., 2020) have proposed adversarial SSL methods based on contrastive learning frameworks. They have demonstrated improved robustness without leveraging any labeled data. However, both of these adversarial SSL frameworks are inefficient as they require a large batch size in order to attain good performances either on clean or adversarial samples.

Recent SSL frameworks (Grill et al., 2020; Chen & He, 2021; Zbontar et al., 2021) mostly resort to maximizing the consistency across two differently augmented samples of the same instance, using an additional momentum encoder (Grill et al., 2020), without any negative pairs or additional networks (Chen & He, 2021; Zbontar et al., 2021). Such non-contrastive SSL frameworks using only



(a) supervised adversarial attack (b) contrastive adversarial attack (c) positive-only similarity adversarial attack

Figure 1: **Motivation.** In supervised adversarial learning (a), perturbation is generated to maximize the cross-entropy loss, which will push adversarial examples to the decision boundaries of other classes. In adversarial contrastive SSL (b), perturbation is generated to minimize the similarity (red line) between positive pairs while maximizing the similarity (blue lines) between negative pairs. Then, the adversarial examples may be pushed to the space of other classes as negative samples may mostly contain samples from other classes. However, in positive-only adversarial SSL (c), minimizing the similarity (red) between positive pairs have weaker constraints in generating effective adversarial examples than supervised AT or contrastive-based SSL. To overcome such a limitation, we suggest a selective targeted attack that maximizes the similarity (blue) to the most confusing target instance (yellow square in (c)).

positive pairs are shown to obtain representations with superior generalization performance compared to contrastive counterparts in a more efficient manner. However, leveraging untargeted adversarial attacks in these SSL frameworks results in a suboptimal performance. BYORL (Gowal et al., 2021a), an adversarial SSL framework using only positive pairs, obtains much lower robust accuracies than those of adversarial contrastive-learning SSL methods on the benchmark datasets (Table 3). Then, what is the cause of such suboptimal robustness in a non-contrastive adversarial SSL framework?

We observe that this limited robustness mainly comes from the suboptimality of untargeted attacks; adversarial examples generated by the deployed untargeted attacks are ineffective in improving robustness in non-contrastive adversarial SSL frameworks. As shown in Figure 1, the attack in the inner loop of the adversarial training loss, designed to minimize the distance between two differently augmented samples, perturbs a given example to a random position in the latent space. Thus, the generated adversarial samples have little impact on the final decision boundaries. Contrarily, in contrastive SSL frameworks, the samples are perturbed toward negative samples to maximize the instance classification loss, most of which belong to different classes. Thus, the ineffectiveness of the untargeted attacks in non-contrastive SSL frameworks mostly comes from their inconsideration of other instances.

To tackle this issue, we propose **T**argeted **A**ttack for **RO**bust **S**elf-**S**upervised learning (TAROSS). TAROSS is designed to enhance robustness of a non-contrastive SSL framework with only positive pairs, such as BYOL (Grill et al., 2020) and SimSiam (Chen & He, 2021), by conducting **targeted attacks**, which perturbs the given instance toward a target. However, this leads to the question of which direction we want to perform the targeted attack to, that is unclear in unsupervised adversarial learning without class labels. To address this point, we consider attacking the instance toward another instance, and further perform an empirical study of which target instances help enhance robustness as opposed to randomly selected target instances, in a targeted attack. Based on our observation, we propose a simple yet effective target selection algorithm based on the similarity and entropy between instances.

Our main contributions can be summarized as follows:

- We demonstrate that achieving comparable robustness in the positive-only self-supervised learning (SSL) with contrastive-based SSL is difficult due to ineffective adversarial inputs generated by untargeted attacks.
- We perform an empirical study of different targeted attacks for non-contrastive adversarial SSL frameworks using only positive pairs. Then, based on the observation, we propose a novel targeted adversarial attack scheme which perturbs the target sample in the direction to the most confusing instance to it, based on similarity and entropy.
- We experimentally confirm that the proposed targeted adversarial SSL framework is able to obtain significantly high robustness, outperforming the state-of-the-art contrastive- and positive-only adversarial SSL methods.

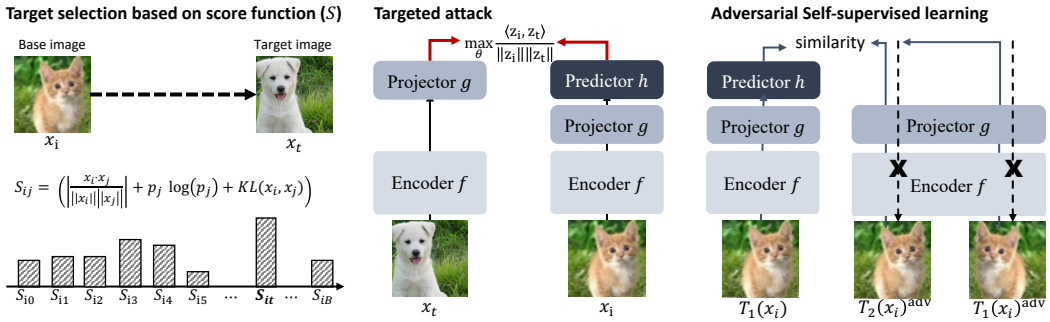


Figure 2: **Overview of TAROSS.** Our approach consists of three parts: target selection, targeted attack, and adversarial self-supervised learning (SSL). We propose a simple and effective similarity- and entropy-based target selection algorithm that selects the maximum score target based on score function ($\mathcal{S}$) (left). When we select the target image (x_t) for each instances, we conduct a targeted attack, which maximizes the similarity between the instance (x_i) and the targeted image (x_t) (middle). Then, we train the model with targeted adversarial examples and clean images as SSL (right).

2 Related Work

Adversarial Training Szegedy et al. (2013) showed that imperceptible perturbation to an input image may lead a DNN model to misclassify a given input into a false label, demonstrating the vulnerabilities of DNN models against adversarial attacks. Goodfellow et al. (2015) proposed the Fast Gradient Sign Method (FGSM), which perturbs a given input to add imperceptible noise in the gradient direction of decreasing the loss of a target model. They also demonstrated that training a DNN model over perturbed as well as clean samples improves the robustness of the model against FGSM attacks. Follow-up works (Kurakin et al., 2016; Carlini & Wagner, 2017) proposed diverse gradient-based strong attacks, and Madry et al. (2018) proposed a projected gradient descent (PGD) attack and a robust training algorithm leveraging a minimax formulation; it finds an adversarial example that achieves a high loss while minimizing the adversarial loss over given data points. Due to a surge of interest in achieving robustness, various defense mechanisms (Song et al., 2017; Buckman et al., 2018; Dhillon et al., 2018) have been proposed.

However, Athalye et al. (2018) showed that many of the previous studies depend on gradient masking, which results in obfuscated gradients in the representation space. At the same time, this renders a target model highly vulnerable to stronger attacks that circumvent obfuscated gradients. TRADES (Zhang et al., 2019) proposed to minimize the Kullback-Leibler divergence (KLD) over clean examples and their adversarial counterparts, thus enforcing consistency between their predictions. They further showed that there is a theoretical trade-off relationship in achieving both clean accuracy and robustness. Recently, leveraging additional unlabeled data (Carmon et al., 2019) and conducting additional attacks (Wu et al., 2020) have been proposed. Carmon et al. (2019) proposed using Tiny ImageNet (Le & Yang, 2015) as pseudo labels, and Gowal et al. (2021b) proposed using generated images from generative models to learn richer representations with additional data.

Self-Supervised Learning Due to the high annotation cost of labeling data, SSL has recently gained a large attention (Dosovitskiy et al., 2015; Zhang et al., 2016; Tian et al., 2020a;b; Zbontar et al., 2021). Previously, SSL focused on solving a pre-task problem of collaterally obtaining visual representation, such as solving the jigsaw puzzle (Noroozi & Favaro, 2016), predicting the relative position of two regions (Dosovitskiy et al., 2015), or impainting the masked area (Pathak et al., 2016). Contrastive SSL coined with data augmentation (Chen et al., 2020; He et al., 2020) has achieved impressive performance in SSL. On the other hand, other previous studies employed a momentum network to learn visual differences between two augmented images (Grill et al., 2020) or to mimic the momentum network with stop-gradient (Chen & He, 2021).

Adversarial Self-Supervised Learning The first adversarial SSL method employed contrastive learning to achieve a high level of robustness (Kim et al., 2020; Jiang et al., 2020) without any class labels. Adversarial self-supervised contrastive learning (Kim et al., 2020; Jiang et al., 2020) generated an instance-wise adversarial example that maximizes the contrastive loss against their positive and negative samples by conducting untargeted attacks. Both methods achieved robustness

Table 1: Experimental results against PGD ℓ_∞ attacks on ResNet18 trained on the CIFAR-5. The CIFAR-5 consists of airplane, automobile, bird, cat, and deer from CIFAR-10.

Self-supervised framework	Method	Clean	PGD ℓ_∞
Contrastive based approach			
SimCLR	ACL Jiang et al. (2020)	80.84	39.16
SimCLR	RoCL Kim et al. (2020)	87.74	47.60
Non-contrastive based approach with untargeted attack			
BYOL	Untargeted attack	75.40	21.00
SimSiam	Untargeted attack	66.36	36.53
Non-contrastive based approach with targeted attack ***(Ours)***			
BYOL	Randomly targeted attack	83.50	**36.50**
BYOL	Entropy and Similarity based targeted attack	**87.08**	35.18
SimSiam	Randomly targeted attack	77.08	47.58
SimSiam	Entropy and Similarity based targeted attack	**79.54**	**49.50**

with the cost of requiring large computation power due to a large batch size for contrastive learning. On the other hand, Gowal et al. (2021a) utilized only positive samples to obtain adversarial examples by maximizing the similarity loss between the latent vectors from the online and target networks, enabling this method free for batch size. However, it exhibited relatively poorer robustness than that of self-supervised contrastive learning, even with an advanced SSL framework. Despite this advanced framework, i.e., non-contrastive SSL, that employs only positive pairs, robustness is not guaranteed with a simple combination of untargeted adversarial learning and advanced SSL. To overcome such vulnerability in non-contrastive SSL, we propose a targeted attack leveraging a novel score function designed to improve robustness.

3 Targeted Adversarial Self-supervised Learning

Adversarial SSL has been proposed to apply existing supervised adversarial learning to models trained with SSL; this method generates adversarial examples in an untargeted manner and leverages these examples to improve robustness. Despite previous works being rather a straightforward combination of two approaches, we argue that leveraging untargeted adversarial attacks still leaves large room for robustness to improve.

3.1 Attack in Adversarial Self-supervised Learning

Adversarial supervised training. To explain adversarial SSL, we first recap adversarial supervised training with our notations. We denote the dataset $\mathcal{D} = \{X, Y\}$, where $x \in X$ is a training example, and $y \in Y$ is its corresponding label. In this supervised learning task, a model is $f_\theta : X \to Y$, where θ is a set of the model parameters to train.

Given $\mathcal{D}$ and f_θ, an *adversarial attack* generates an adversarial example by adding a perturbation to a given source image that maximizes the loss within a certain radius from it (e.g., ℓ_∞ norm balls). We define this adversarial ℓ_∞ attacks as follows:

$$\delta^{t+1} = \Pi_{B(0,\epsilon)}\Big(\delta^t + \alpha \texttt{sign}\Big(\nabla_{\delta^t}\mathcal{L}_{\text{CE}}\big(f(\theta, x+\delta^t), y\big)\Big)\Big), \tag{1}$$

where $B(0,\epsilon)$ is the ℓ_∞ norm-ball with radius ϵ, Π is the projection function to the norm-ball, α is the step size of the attacks, and $\texttt{sign}(\cdot)$ is the sign of the vector. Also, the perturbation δ is the accumulated perturbations by $\alpha\texttt{sign}(\cdot)$ over multiple iterations t, and $\mathcal{L}_{\text{CE}}$ is the cross-entropy loss. In the case of PGD (Madry et al., 2018), the attack starts from a random point within the epsilon ball and performs t gradient steps, to obtain a perturbed sample $x^{\texttt{adv}}$.

Adversarial training. (AT) is a straightforward way to improve the robustness of the model by minimizing the training loss embedding the adversarial described above as an inner loop, as follows:

$$\mathcal{L}_{\text{AT}} = \max_{\delta \in B(x,\epsilon)} \mathcal{L}_{\text{CE}}\big(f(\theta, x+\delta), y\big). \tag{2}$$

Untargeted adversarial self-supervised learning. Previously proposed adversarial SSL methods (Jiang et al., 2020; Kim et al., 2020; Gowal et al., 2021a) naturally combined the adversarial supervised training and SSL frameworks. Therefore, all previous works design the inner loop of

Algorithm 1 Targeted Attack Robust Self-Supervised Learning (TAROSS)

Require: Dataset $\mathcal{D}$, transformation function $\mathbf{T}$, model f, parameter of model θ, target score function $\mathcal{S}$, constant w
  for $\texttt{iter} \in$ number of iteration **do**
    for $x_i \in$ miniBatch $B = \{x_1, \ldots, x_m\}$ **do**
      for $\texttt{r}$ in 2 **do**
        transform input $\mathbf{T}_r(x_i)$
        Find target images $\mathbf{t}_r(x_j) = \mathcal{S}(\mathbf{T}_r(x_i), B)$
        Generate targeted adversarial examples $\mathbf{T}_r(x_i)^{adv} = \mathbf{T}_r(x_i) + \delta^t$
        $\delta^{t+1} = \Pi_{B(0,\epsilon)}\Big(\delta^t + \alpha \texttt{sign}\Big(\nabla_{\delta^t}\mathcal{L}_{\texttt{target}}\big(f(\theta, \mathbf{T}_r(x_i) + \delta^t), f(\theta, \mathbf{T}_r(x_j))\big)\Big)\Big)$
      end for
      Calculate training loss
      $\mathcal{L}_{\texttt{TAROSS}} = \mathcal{L}_{\texttt{selfsup}}(\mathbf{T}_1(x_i), \mathbf{T}_1(x_i)^{adv}) + w \cdot \mathcal{L}_{\texttt{selfsup}}(\mathbf{T}_1(x_i)^{adv}, \mathbf{T}_2(x_i)^{adv})$
    end for
    Optimize the weight θ over $\mathcal{L}_{\texttt{TAROSS}}$
  end for

adversarial attack with untargeted attack as follows:

$$\delta^{t+1} = \Pi_{B(0,\epsilon)}\Big(\delta^t + \alpha \texttt{sign}\Big(\nabla_{\delta^t}\mathcal{L}_{\texttt{loss}}\big(f(\theta, \mathbf{T}_1(x) + \delta^t), f(\theta, \mathbf{T}_2(x))\big)\Big)\Big), \tag{3}$$

where perturbation is generated to maximize the self-supervised loss $\mathcal{L}_{\texttt{loss}}$ that minimizes the similarity between positive pairs, and maximize the similarity between negative pairs if negative pairs exists (Jiang et al., 2020; Kim et al., 2020).

Targeted adversarial self-supervised learning We argue that leveraging untargeted adversarial attacks in an SSL framework using positive-only pairs still leaves large room for better robustness. We can simply improve robustness in non-contrastive SSL by changing the inner loop to a randomly targeted attack as follows:

$$\delta^{t+1} = \Pi_{B(0,\epsilon)}\Big(\delta^t + \alpha \texttt{sign}\Big(\nabla_{\delta^t}\mathcal{L}_{\texttt{target}}\big(f(\theta, x + \delta^t), f(\theta, x')\big)\Big)\Big), \tag{4}$$

where $\mathcal{L}_{\texttt{target}} = -\mathcal{L}_{\texttt{loss}}$, x' is a randomly selected target within the batch. To precisely elaborate our observation in the latter section, we introduce a score function ($\mathcal{S}(x, \texttt{batch})$) to select the target ($x'$) within the batch. The score function ($\mathcal{S}$) can be a random-sampling function or a designed function as Eq. 6. In other words, the score function outputs the target (x') corresponding to the base image (x), then the targeted attack generates perturbation that maximizes the similarity to target x'.

In Table 1, an untargeted attack makes the model more vulnerable when the non-contrastive SSL frameworks utilize only positive pairs, such as in BYOL (Grill et al., 2020), and SimSiam (Chen & He, 2021). However, when we modify the inner loop of the untargeted attack to the targeted attack with the simple random sampling score function as Eq. 4., non-contrastive SSL achieves better robustness.

3.2 Observations of Target Selection based on Score Function

In the previous section, we found that the targeted attack can improve robustness even with randomly selected target images in the batch. However, we argue there exist effective target images for each base input x that can further improve robustness. However, selecting effective target instances for the targeted attack is not trivial. To enable this, we propose leveraging the K-means clustering-based and similarity-based score functions. We mainly observe our score function on SimSiam (Chen & He, 2021) and futher describe the details of our method for observations in Appendix C.

K-means clustering-based target selection. Intuitively, perturbing the most confusing class can be a straightforward way of implementing a strong adversarial attack for better robustness. To validate our assumption, we first design the score function based on labels from K-means clustering. To find the most confusing instance, we label the pseudo class y_i on the source images x_i with K-means clustering and then find the most closest cluster y_t to the pseudo class y_i based on the centroid vector of each cluster. Accordingly, we can filter the instances that are labeled as y_t from the batch.

Among those, the score function selects the instance that is closest to the source images x_i.

Table 2: Results of our adversarial SSL with K-means clustering based-, and similarity score based- selection for targeted attack on CIFAR5.

Selection	Clean	PGD ℓ_∞
Untargeted*	66.36	36.53
Random	77.08	47.58
Similarity	76.04	48.73
K-means clustering	75.86	49.39

As shown in Table 2, the K-means clustering target selection score function shows better robustness than the random selection when performing targeted attacks. However, K-means clustering requires a lot of computation resources to derive a pseudo class label at every iteration, and it is vague to set the K value when there is no label information. Moreover, we cannot guarantee that the closest instance in the confused cluster is always the best candidate to improve robustness (Zhang et al., 2020) becase the closest cluster to an instance itself may differ from the closest cluster to the pseudo class cluster. For example, when the cat class cluster is close to the dog class cluster, there could be a cat instance close to horse images on the other side. Therefore, we also test the target selection score function based on the distance of each instance.

Similarity-based target selection. In SSL, the similarity between given instances is an important metric to learn a good visual representation without any class label information (Chen et al., 2020; Tian et al., 2020b). Therefore, we leverage the similarity score to find optimal target images in the adversarial SSL. Before designing the score function based on the similarity score, we first observe the influence scores (Koh & Liang, 2017) of training points that are generated by the targeted attacks with randomly selected targets. The influence score is designed to approximate the influence function in DNN to compute the degree to which the training points are responsible for a given prediction (Koh & Liang, 2017). We followed Koh & Liang (2017) to calculate the influence scores of the adversarial examples that are generated by the targeted attack with a variety of target images. However, since this approximation requires a lot of computations, and the possible pair for the targeted attack is squared of the batch size, we sampled 60,000 training points to observe the correlation between the similarity and influence scores.

We found that the input pairs having the similarity of around 0.0 have relatively high influence scores (Figure 7). To empirically verify whether the instances with the high influence score actually act as an effective target to show better robustness in adversarial SSL, we design an experiment in which we select target images based on the similarity. We experiment with five cases that have a different range of the similarity: r1: $-1.0 \sim -0.5$, r2: $-0.5 \sim -0.25$, r3: $-0.25 \sim 0.0$, r4: $0.0 \sim 0.25$, r5: $0.25 \sim 0.5$, and r6: $0.5 \sim 1.0$. As shown in Figure 3, when we select the target images between $-0.25 \sim 0.25$, it improves robustness compared to that using the random selection algorithm.

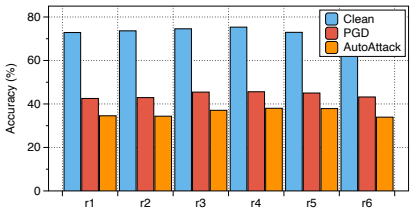


Figure 3: Robustness of the model that conduct targeted attack based on similarity score which selecting the target image that are in pre-defined range ($r1, \cdots, r6$).

3.3 Targeted Attack with Entropy- and Similarity-based Score Function

In this section, we describe our overall framework to achieve robustness in non-contrastive SSL performing targeted attacks wherein targets are selected by leveraging our proposed score function. The target score function is designed based on the observation in the previous section.

Positive-paired self-supervised learning. Our targeted attack is designed for an SSL framework that only employs positive pairs of transformed images. Therefore, we mainly describe our method on the SimSiam (Chen & He, 2021).

We first describe an SSL method, SimSiam (Chen & He, 2021), that learns visual representations with only positive pairs using a stop-gradient. Let us denote the dataset $\mathcal{D} = \{X\}$, and transformation set $\mathbf{T}$ that augments the images $x \in X$. SimSiam consists of the encoder f, followed by the projector g, and the predictor h; each of g and h is a multi-layer perceptron (MLPs). To learn visual representations, SimSiam maximizes the cosine similarity between the positive pairs as follows:

$$\mathcal{L}_{\texttt{SimSiam}}(\mathbf{T}_1(x), \mathbf{T}_2(x)) = -\frac{1}{2}\frac{p_1}{||p_1||_2} \cdot \frac{z_2}{||z_2||_2} - \frac{1}{2}\frac{p_2}{||p_2||_2} \cdot \frac{z_1}{||z_1||_2}, \tag{5}$$

where $p_1 = h(g(f(\mathbf{T}_1(x))))$ and $z_2 = g(f(\mathbf{T}_2(x)))$ are output vectors of the predictor h and the projector g, respectively. Before calculating the loss, SimSiam detaches the gradient on the z which called stop-gradient. Stop-gradient helps the model to form proper visual representations without any momentum network but makes the single model acts like a momentum network.

Entropy and similarity based target selection. From the observation in the previous paragraph, we argue that selecting target instances that are most confused and are adequately close to the base instance is effective for targeted attacks. For this, we design the score function based on the similarity and entropy values; it does not need any class information, as follows:

$$\mathcal{S}(\mathbf{T}_1(x)_i, \mathbf{T}_2(x)_j) = \frac{e_i}{||e_i||_2} \cdot \frac{e_j}{||e_j||_2} + (p_j/\tau) \log \big(p_j/\tau\big), \tag{6}$$

where $p_i = h(g(f(\mathbf{T}_1(x)_i)))$ and $e_i = f(\mathbf{T}_1(x)_i)$ are output vectors of predictor h and encoder f, respectively. Overall, the score function $\mathcal{S}$ consists of a cosine similarity term, and an entropy term. The cosine similarity is calculated between features of base images and candidate images in the differently augmented batch ($\mathbf{T}_2$). The entropy is calculated with the assumption that the vector p is an instance's logit as Caron et al. (2021). We design the score function to work as a combination of selection algorithms based on K-means clustering and similarity values. It naturally selects the target that is mostly confused with base images (Figure 1). We also verify our score function selects targets as our intention in the experiment section (Figure 4b).

Robust self-supervised learning with target attack (TAROSS). In non-contrastive SSL, we make a positive pair, i.e., $\mathbf{T}_1(x), \mathbf{T}_2(x)$, with differently transformed augmentation. As shown in Figure 2, to generate adversarial examples, we first select the target images $\mathbf{T}_1(x)', \mathbf{T}_2(x)'$ for each base image $\mathbf{T}_1(x), \mathbf{T}_2(x)$, respectively, which have the maximum score from score function ($\mathcal{S}$). Here, we employ score function ($\mathcal{S}$) in Eq. 6. Then, we generate adversarial examples, i.e., $\mathbf{T}_1(x)^{adv}, \mathbf{T}_2(x)^{adv}$, for each transformed input with our suggested targeted attack (Eq. 4) where $\mathcal{L}_{\texttt{target}} = -\mathcal{L}_{\texttt{SimSiam}}$. Finally, we maximize the agreement between adversarial images $\mathbf{T}_1(x)^{adv}$, and $\mathbf{T}_2(x)^{adv}$ with clean image $\mathbf{T}_1(x)$ as follows:

$$\mathcal{L}_{\texttt{TAROSS}} = \mathcal{L}(\mathbf{T}_1(x), \mathbf{T}_1(x)^{adv}) + w \cdot \mathcal{L}(\mathbf{T}_1(x)^{adv}, \mathbf{T}_2(x)^{adv}) \tag{7}$$

where $\mathcal{L}$ is Eq. 5. Since all three instances have the same identity, we maximize the similarity between the clean image and adversarial examples. Overall, TAROSS is summarized in Algorithm 1.

4 Experiment

In this section, we validate our TAROSS on non-contrastive adversarial SSL frameworks (Section 4.1). Moreover, we evaluate the robustness of the representation with different datasets and in transfer-learning tasks (Section 4.2). Finally, we analyze why our targeted attacks help achieve better robustness compared to models using untargeted attacks (Section 4.3).

Experimental setup. We validate our TAROSS on top of BYOL (Grill et al., 2020) and SimSiam (Chen & He, 2021). To compare the effectiveness of TAROSS, we also implement non-contrastive adversarial SSL using untargeted attacks in each SSL framework. All the models are trained on ResNet18 with ℓ_∞ PGD attacks with an attack step of 10 with epsilon $8/255$. We evaluate the robustness against AutoAttack (Croce & Hein, 2020a) and ℓ_∞ PGD attacks with epsilon $8/255$ using an attack step of 20 iterations. We further describe the details of our experimental settings in Appendix B. The code will be available in Anonymous.

4.1 Evaluating Robustness

Robustness with targeted/untargeted attack. In Table 3, as we observed in the previous section, when replacing untargeted attacks with targeted attacks in positive-only SSL (BYOL and SimSiam), it obtains robustness comparable to contrastive adversarial SSL (RoCL, ACL), which verifies our motivation. Therefore, it is feasible to improve robustness even in positive-only SSL while attaining the pros of positive-only SSL frameworks. Interestingly, the targeted attacks helped improve the clean accuracy in SSL when compared with those using untargeted attacks in positive-only SSL. We conjecture that untargeted attacks are not only inadequate to learn robust features but also hinder to learn good visual representation of natural images.

Table 3: Experimental results against white box attacks on ResNet18 trained on the CIFAR10. We evaluate adversarial SSL with linear evaluation and robust linear evaluation. Clean is the accuracy of clean images. All models are evaluated with PGD ℓ_∞ with 20 steps of $\epsilon = 0.0314$ and AutoAttack (Croce & Hein, 2020a). Untargeted attack maximizes the training loss between the differently augmented but the same instances to generate adversaries. To see the effectiveness of our methods, we test TAROSS on positive-only SSL, i.e., BYOL, SimSiam (grey box).

Train type	Self-supervised framework	Method	Clean	PGD ℓ_∞	AutoAttack
Supervised	-	$\mathcal{L}_{\text{CE}}$	92.82	0.00	0.00
	-	AT (Madry et al., 2018)	83.07	53.78	49.23
Self-supervised linear evaluation	SimCLR	RoCL (Kim et al., 2020)	78.14	42.89	27.19
	SimCLR	ACL (Jiang et al., 2020)	79.96	39.37	35.97
	BYOL	Untargeted attack	72.65	16.20	0.01
	BYOL	**TAROSS**	**84.52**	**31.19**	**21.01**
	SimSiam	Untargeted attack	71.78	37.28	32.41
	SimSiam	**TAROSS**	**74.06**	**44.71**	**36.39**
Self-supervised robust linear evaluation	SimCLR	RoCL (Kim et al., 2020)	76.53	47.51	30.22
	SimCLR	ACL (Jiang et al., 2020)	77.17	40.67	39.13
	BYOL	Untargeted attack	54.01	27.24	4.49
	BYOL	**TAROSS**	**74.33**	**40.84**	**29.91**
	SimSiam	Untargeted attack	68.88	37.84	31.44
	SimSiam	**TAROSS**	**76.19**	**45.57**	**37.25**

Table 4: Results of self-supervised adversarial learning evaluated with linear evaluation in STL10 and CIFAR10. Rob. stands for robust accuracy against PGD ℓ_∞ attack with 20 steps and $\epsilon = 0.0314$.

		STL10		CIFAR100	
Method	Attack	Clean	Rob.	Clean	Rob.
RoCL	Untarget	52.63	19.19	45.99	17.17
ACL	Untarget	54.14	19.69	41.09	15.31
BYOL	Untarget	61.02	12.14	53.43	14.81
SimSiam	Untarget	43.88	12.70	27.53	14.20
SimSiam	**TAROSS**	**46.38**	**21.32**	**36.02**	**22.18**

Table 5: Results of adversarial transfer learning to CIFAR10 and STL10 from CIFAR100 pretrained models followed the experimental setting of Shafahi et al. (2020). Rob. stands for robust accuracy against PGD ℓ_∞ 20 attack with ϵ =0.0314.

CIFAR100 →		CIFAR10		STL10	
Method	Attack	Clean	Rob.	Clean	Rob.
RoCL	Untarget	73.93	18.62	74.06	19.06
ACL	Untarget	61.68	15.66	59.56	13.53
BYOL	Untarget	71.06	16.57	63.53	12.93
SimSiam	Untarget	50.05	25.43	34.54	**24.82**
SimSiam	**TAROSS**	**50.50**	**25.44**	**43.13**	22.46

Robustness compared to contrastive-based approaches. We compare our TAROSS to contrastive adversarial SSL methods to show that our approach could make positive-only SSL to achieve robustness comparable to previous works. To this end, we evaluate adversarial SSL with linear evaluation and robust linear evaluation against AutoAttack (Croce & Hein, 2020a) and PGD (Madry et al., 2018) ℓ_∞ attacks. Notably, as shown in Table 3, the targeted attacks allow the positive-only SSL model to have better robustness than that of contrastive adversarial SSL (RoCL, ACL) in linear evaluation. Specifically, when we train the fc layer, our model achieves 36.39% of robustness against strong AutoAttack which shows the effectiveness of our proposed method. Interestingly, the SimSiam-based model has already obtained good robustness in the linear evaluation compared to the robust linear evaluation which takes $\times 4$ times to train. We conjecture that SimSiam-based TAROSS already has a robust well-cluster after the robust pretraining which does not need to find robust decision boundaries with the adversarial examples. Since the SimSiam directly maximizes the similarity between adversary and clean images only between the positive pairs with the single models, the gap between the representation of adversaries and clean examples may reduce.

4.2 Evaluating the Quality of Robust Representation

Robustness on multiple benchmarks datasets. We validate our method on multiple benchmark datasets, such as CIFAR100 and STL10. In Table 4, our TAROSS consistently shows good robustness across different benchmark datasets when comparing with adversarial SSL frameworks using untargeted attacks. Especially, when the dataset becomes larger, such as CIFAR100, our method achieves even better robustness than contrastive learning-based approaches, e.g., RoCL, and ACL, where these approaches are sensitive to the number of same class samples in the single batch size. From this result, we argue that our targeted adversarial SSL contributes to learning better robustness in the larger dataset than contrastive-based SSL.

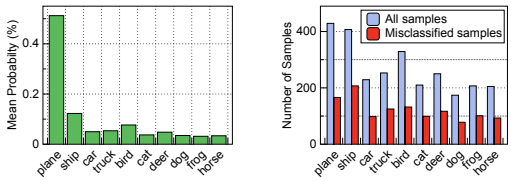
Transfer to different data. SSL is effective to utilize in several downstream tasks with a pretrained encoder. Therefore, we also evaluate how our robust pretrained features help in the different datasets in transfer tasks, which demonstrates the quality of our robust pretrained features in a different dataset.

We followed the experimental setting as supervised adversarial transfer learning tasks (Shafahi et al., 2020) which freeze the encoder and train only the fc layer. We pretrained the model on CIFAR100 and evaluate the robust transferability to STL10, and CIFAR10. Once our model is trained, we transfer the robust features to different kinds of datasets with fewer epochs with fewer parameters, which is efficient in that we do not need to conduct adversarial training from scratch. In Table 5, our model also shows impressive transferability both in CIFAR10 and STL10 compare to untargeted adversarial SSL. Moreover, while our model shows relatively low clean accuracy, TAROSS could obtain about 10% gain in robustness compared to contrastive-based adversarial SSL. From these results, we confirm that our methods generate robust representations that can be transferred to several downstream tasks, which previous adversarial SSL studies addressed.

4.3 Analysis of Targeted Attacks

In this section, we further analyze the targeted attacks in adversarial SSL to uncover evidence for why our proposed method is effective in adversarial SSL. We analyze which class is frequently selected when using our score function. Then, we visualize the difference between an untargeted attack and a targeted attack in the representation space.

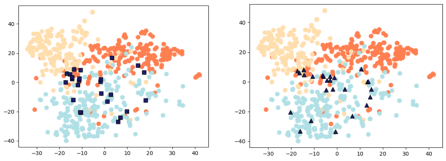
Analyze the distribution of the target class. We analyze which class is selected as a target by the score function ($\mathcal{S}$). We test the selected target of the single class (airplane) to the adversarial supervised trained models (Madry et al., 2018) to obtain the probability and prediction for analyzing the class distribution. As shown in Figure 4b, half of the target images are the same class as the base image. And the next most frequently selected class was the class with the second highest probability that the images of the corresponding class were most confused (Figure 4a). The score function we proposed selects images of the same class when the image in the center of the cluster, and images near the decision boundary that have a high probability to predict to a wrong class select images of a different class. Through these results, it can be confirmed that we designed a score function that appropriately uses K-means clustering to always select a different class from itself and heuristic target selection, which performs an instance-wise attack close to a similarity value, depending on the case, as we intended.



(a) Mean predict probability of base images (b) Distribution of class of targeted images

Figure 4: Analysis of target from score function ($\mathcal{S}$)

Visualization of embedding space. To observe the differences between images that are generated with targeted versus untargeted attacks, we visualize the embedding space of targeted attack examples and untargeted attack examples. In Figure 5, adversarial examples are denoted with dark blue, and clean examples are denoted with light blue; both are instances of the same class. As shown in Figure 5a, untargeted adversarial examples are located near clean examples. On the other hand, targeted adversarial examples are located near the boundary of the cluster (Figure 5b). This visualization shows that our targeted attack generates relatively more effective adversarial examples than untargeted attacks, which is likely to push the decision boundary to have a better robust representation.



(a) Untargeted attack (b) Targeted attack

Figure 5: Visualize embedding

5 Conclusion

In this paper, we showed that a simple combination of supervised adversarial training with SSL is highly suboptimal due to the ineffectiveness of adversarial examples generated by the untargeted adversarial attacks perturbed to random places without the consideration of decision boundaries. Based on this observation, we propose an instance-wise targeted attack scheme for an adversarial SSL framework. This scheme selects the target instance based on the similarity and entropy, such that the given instance is perturbed to be confused with the selected target. Our targeted adversarial SSL framework obtains representation that achieves better robustness than the state-of-the-art adversarial SSL frameworks, including contrastive ones, without using any negative pairs or additional networks. We believe that our work opens a door to future research on the search for more effective targeted attacks, for adversarial SSL.

Reproducibility Statement

- **Datasets.** We use CIFAR10, CIFAR100, STL10, and CIFAR5 datasets for our experiment. To see more details, please see Supplementary B.2.
- **Baselines.** We use following models as our baseline: RoCL (Kim et al., 2020), ACL (Jiang et al., 2020), and BYORL (Gowal et al., 2021a). To see the training details, please see Supplementary A
- **Robustness Test.** We test our model and baselines against PGD attacks (Madry et al., 2018) and AutoAttack (Croce & Hein, 2020a). To see the evaluation details for robustness, please see Section 4. Experimental setup., and Supplementary B.4.
- **TAROSS.** To train our model, please see Supplementary B.3.
- **Table 1.** To reproduce the results in Table 1, please see Section 3.1.
- **Table 2.** To reproduce the results in Table 2, please see Section 3.2. K-means clustering-based target selection., and Supplementary C
- **Figure 3.** To reproduce the results in Figure 3, please see Section 3.2. Similarity-based target selection, and Supplementary C
- **Figure 4, and 5.** To reproduce results of Figure 4, and 5, please see Supplementary D.

REFERENCES

Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *European Conference on Computer Vision*, pp. 484–501. Springer, 2020.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning*, 2018.

Jacob Buckman, Aurko Roy, Colin Raffel, and Ian Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. 2018.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE symposium on security and privacy (sp)*, pp. 39–57. IEEE, 2017.

Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, Percy Liang, and John C Duchi. Unlabeled data improves adversarial robustness. *Advances in Neural Information Processing Systems*, 2019.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9650–9660, 2021.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, 2020.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15750–15758, 2021.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International Conference on Machine Learning*, pp. 2206–2216. PMLR, 2020a.

Francesco Croce and Matthias Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *International Conference on Machine Learning*, pp. 2196–2205. PMLR, 2020b.

Guneet S Dhillon, Kamyar Azizzadenesheli, Zachary C Lipton, Jeremy Bernstein, Jean Kossaifi, Aran Khanna, and Anima Anandkumar. Stochastic activation pruning for robust adversarial defense. *International Conference on Learning Representations*, 2018.

Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(9):1734–1747, 2015.

Lijie Fan, Sijia Liu, Pin-Yu Chen, Gaoyuan Zhang, and Chuang Gan. When does contrastive learning preserve adversarial robustness from pretraining to finetuning? *Advances in Neural Information Processing Systems*, 34, 2021.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

Sven Gowal, Po-Sen Huang, Aaron van den Oord, Timothy Mann, and Pushmeet Kohli. Self-supervised adversarial robustness for the low-label, high-data regime. In *International Conference on Learning Representations*, 2021a.

Sven Gowal, Sylvestre-Alvise Rebuffi, Olivia Wiles, Florian Stimberg, Dan Andrei Calian, and Timothy A Mann. Improving robustness using generated data. *Advances in Neural Information Processing Systems*, 34:4218–4233, 2021b.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019.

Ziyu Jiang, Tianlong Chen, Ting Chen, and Zhangyang Wang. Robust pre-training by adversarial contrastive learning. In *Advances in Neural Information Processing Systems*, 2020.

Minseon Kim, Jihoon Tack, and Sung Ju Hwang. Adversarial self-supervised contrastive learning. *Advances in Neural Information Processing Systems*, 2020.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pp. 1885–1894. PMLR, 2017.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton Earnshaw, Imran Haque, Sara M Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. Wilds: A benchmark of in-the-wild distribution shifts. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5637–5664. PMLR, 18–24 Jul 2021.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

Ya Le and X. Yang. Tiny imagenet visual recognition challenge. 2015.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pp. 69–84. Springer, 2016.

Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2536–2544, 2016.

Ali Shafahi, Parsa Saadatpanah, Chen Zhu, Amin Ghiasi, Christoph Studer, David Jacobs, and Tom Goldstein. Adversarially robust transfer learning. *International Conference on Learning Representations*, 2020.

Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. *International Conference on Learning Representations*, 2017.

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.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *European Conference on Computer Vision*, 2020a.

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning. In *Advances in Neural Information Processing Systems*, 2020b.

Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*, 2019.

Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems*, 33, 2020.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pp. 12310–12320. PMLR, 2021.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P Xing, Laurent El Ghaoui, and Michael I Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, 2019.

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*, pp. 11278–11287. PMLR, 2020.

Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European Conference on Computer Vision*, pp. 649–666. Springer, 2016.

Supplementary Material

Targeted Adversarial Self-Supervised Learning

A BASELINES.

- **RoCL (Kim et al., 2020).** RoCL is SimCLR (Chen et al., 2020) based adversarial self-supervised learning methods. We experiment with the official code[1]. To make a fair comparison, we set the attack step to 10 as other baselines. We train the model with 1,000 epochs under the LARS optimizer with weight decay $2e-6$ and momentum with 0.9. For the learning rate schedule, we also followed linear warmup with cosine decay scheduling. We set a batch size of 512 for all datasets (CIFAR10, CIFAR100, STL10). For data augmentation, we use a random crop with 0.08 to 1.0 size, horizontal flip with a probability of 0.5, color jitter with a probability of 0.8, and grayscale with a probability of 0.2 for RoCL training.
- **ACL (Jiang et al., 2020)** ACL is SimCLR (Chen et al., 2020) based adversarial self-supervised learning methods. We conduct the experiment with the official code[2]. To make a fair comparison, we set the attack step to 10 as other baselines. We train the model with 1,000 epochs. We set a batch size of 512 for STL10 dataset. For CIFAR10, and CIFAR100, we use the official pretrained checkpoints. For data augmentation, we use a random crop with 0.08 to 1.0 size, horizontal flip with a probability of 0.5, color jitter with a probability of 0.8, and grayscale with a probability of 0.2 for ACL training. We set PGD dual mode which calculates both clean and adversarial during the training.
- **BYORL (Gowal et al., 2021a)** BYORL is BYOL (Grill et al., 2020) based adversarial self-supervised learning methods for low label regime. Since there is no official code for BYORL we implement the BYORL by ourselves. Though we mentioned as BYOL with untargeted attack in the Table 1, 3, 4 and 5, please note that BYOL with untargeted approach stands for Gowal et al. (2021a). We implement based on BYOL from a self-supervised learning library[3]. We use the same CIFAR-10 setting in the library except for normalization. We exclude normalization in the data augmentation. To make a fair comparison, we implement on the ResNet18 with attack step 10 of PGD. As shown in supplementary materials in Gowal et al. (2021a), when the model is trained with 10 steps in ResNet34 it shows 37.88% of robustness. We conjecture that we have a different performance from the original paper because the original paper employs 40 steps of PGD in WideResNet34 to obtain the reported robustness which requires extraordinary computation power.
- **AdvCL (Fan et al., 2021)** AdvCL is SimCLR (Chen et al., 2020) based adversarial self-supervised learning which employ pseudo labels from the model that is pretrained on ImageNet (Krizhevsky et al., 2012) data. Even though the outstanding performance of AdvCL, we exclude this model as our baseline because the proposed methods require the model that is trained with the labels of ImageNet which we assume to have no label information for pretraining the model.

B DETAILED DESCRIPTION OF EXPERIMENTAL SETUPS

B.1 RESOURCE DESCRIPTION.

All experiments are conducted with a two NVIDIA RTX 2080 Ti, except for the experiments with CIFAR100 experiments. For CIFAR100 experiments, two NVIDIA RTX 3080 are used. All experiments are processed in Intel(R) Xeon(R) Silver 4114 CPU @ 2.20GHz.

[1] https://github.com/Kim-Minseon/RoCL
[2] https://github.com/VITA-Group/Adversarial-Contrastive-Learning
[3] https://github.com/vturrisi/solo-learn

B.2 DATASET DESCRIPTION.

For experiments, we use CIFAR 10, CIFAR 100, CIFAR 5, and STL10. CIFAR 10 and CIFAR 100[4] consist of 50,000 training images and 10,000 test images with 10 and 100 classes, respectively. All CIFAR images are 32×32×3 resolution (width, height, and channel). CIFAR 5 is a subset of CIFAR 10 which contains all images from 5 classes: airplane, automobile, bird, cat, and deer. We also test with determined classes. The STL10[5] consists of 5,000 training images and 8,00 test images with 10 classes. All STL10 images are 96×96×3 resolution (width, height, and channel.) However, for our experiment in Table 4, 5, we resize the images into 32×32×3 resolution (width, height, and channel).

B.3 TRAINING DETAIL.

For all methods, we train on the ResNet18 (He et al., 2016) with ℓ_∞ attacks with the attack strength of $\epsilon = 8/255$ and the step size of $\alpha = 2/255$, with the number of inner maximization iteration set to $K = 10$. For the optimization, we train every model for 800 epochs using the SGD optimizer with the learning rate of 0.05, weight decay of $5\text{e}{-4}$, and the momentum of 0.9. For data augmentation, we use a random crop with 0.08 to 1.0 size, horizontal flip with a probability of 0.5, color jitter with a probability of 0.8, and grayscale with a probability of 0.2. We exclude normalization for adversarial training. We set the weight of adversarial similarity loss w as 2.0. We use batch size 512 with two GPUs.

In the score function, we calculate the similarity score term and entropy term as shown in Equation 6. First, to exclude the positive pairs' similarity score we set the similarity score between positive pairs to -1. Then, to calculate the overall score, after obtaining the similarity score and entropy of each sample, we normalize each component with Euclidean normalization to balance each component to score function.

B.4 EVALUATION DETAILS.

PGD ℓ_∞ attack. For all PGD ℓ_∞ attacks used in the test time, we use the projected gradient descent (PGD) attack with the strength of $\epsilon = 8/255$, with the step size of $\alpha = 8/2550$, and with the number of inner maximization iteration set to $K = 20$ with the random start.

AutoAttack. We further test against strong gradient based attack, i.e., AutoAttack (AA) Croce & Hein (2020a). AutoAttack is an ensemble attack of four different attacks (APGD-CE, APGD-T, FAB-T (Croce & Hein, 2020b), and Square (Andriushchenko et al., 2020)). AGPD-CE is an untargeted attack, APGD-T and FAB-T are targeted attacks. The Square is a black box attack. We use official code to test models[6].

Self-supervised learning. For self-supervised learning, we denote linear evaluation when we use only clean images to train the fully connected (fc) layer after the pretraining phase. When we denote robust linear evaluation, we train the fc layer with adversarial examples. While ACL uses partial fine-tuning to obtain their reported accuracy and robustness, to make a fair comparison we freeze the encoder and train only the linear layer. Robust fine-tuning is training all parameters including parameters of the encoder with adversarial examples. For linear evaluation, we followed baselines hyperparameters for each model. We train the baseline models with 150 epochs, 25 epochs, and 50 epochs for RoCL, ACL, and BYORL, respectively. We also followed their learning rate 0.1, 0.1, and 2×10^-3 for RoCL, ACL, and BYORL, respectively. On the other hand, we train our model with 100 epochs with a learning rate of 0.5 for linear evaluation. We use AT loss for robust linear evaluation except for ACL. For ACL, we use TRADES loss as the official code.

C EXPERIMENT DETAILS OF OUR TARGETED ATTACK

K-means cluster-based TAROSS Intuitively, perturbing to the most confusing class is a straightforward way of implementing a strong adversarial attack for better robustness. To validate our

[4] http://www.cs.toronto.edu/~kriz/cifar.html.
[5] https://ai.stanford.edu/~acoates/stl10/
[6] https://github.com/fra31/auto-attack

assumption, we design the score function based on labels from K-means clustering. We set K as 5 for CIFAR 5 in Table 2.

As show in Figure 6, we label the pseudo-class y_i on the source images x_i with the feature z_i. Then find the closest cluster y_t to pseudo class y_i based on the centroid vector c of each cluster. We use the cosine similarity function to measure the distance between the latent vector z and centroid vector c. To calculate the K-means cluster for each batch, K-means cluster-based TAROSS takes high computation.

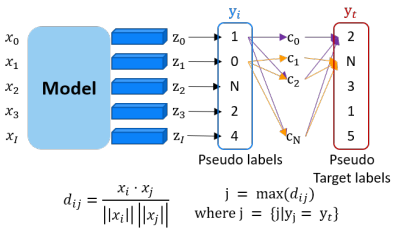
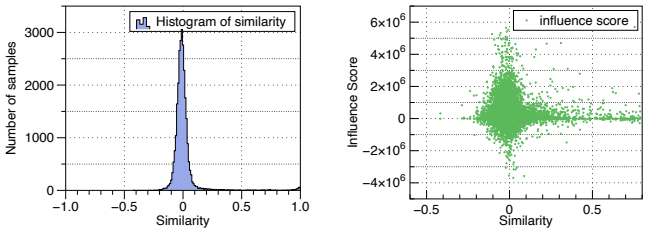


Figure 6: K-means clustering target selection

Similarity score based TAROSS To calculate the influence score during the training, we employ underfitted adversarial self-supervised model which learns visual representation a little with 400 epochs. To calculate all pairs of the training set, it requires $\mathbf{O}(\mathcal{D}^2)$. Moreover, the computation of influence score (Koh & Liang, 2017) for single pair also takes a lot due to approximation. Therefore, we randomly sampled a subset of the dataset from the training set. We use official code of influence score[7]. As shown in Figure 3, samples that are in $-0.25 \sim 0.25$ shows relatively high influence score.



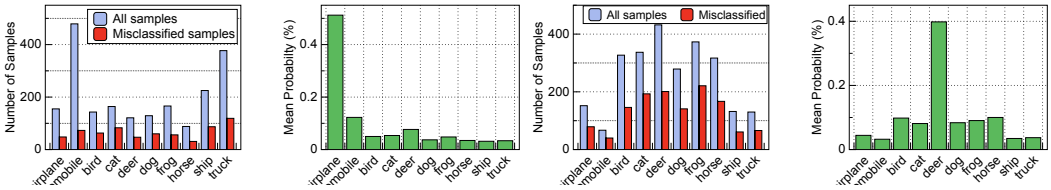
(a) Histogram of cosine similarity between sampled pairs.

(b) Scatter of influence score based on similarity

Figure 7: Analysis with influence score.

D Experimental Details of Analysis

Analysis the distribution of target class To analyze the target from the score function ($\mathcal{S}$), we employ an adversarially supervised trained model. We calculate the score function that is trained with our TAROSS on SimSiam. We use a train set. For each class, we calculate the mean predict probability which is the average of all softmax outputs of target images from the supervised trained model. Further, we also count the number of samples that are predicted for each class. In Figure 4, the results are target images of the airplane as a base image. There is a similar tendency even though we change the base class to other classes as shown in the following Figure 8.



(a) Distribution of class of target of *automobile*.

(b) Mean predict probability of *automobile*

(c) Distribution of class of target of *deer*.

(d) Mean predict probability of *deer*

Figure 8: Analysis of target distribution in different class

Visualization of embedding space To visualize the embedding of our targeted attack and untargeted attack, we use our TAROSS model that is trained on CIFAR10 as a feature extractor. We sample a few examples and conduct two types of attacks, untargeted attack, and targeted attack. To visualize more effectively we ignore the other seven classes in CIFAR10. We visualize clean examples from three classes and then visualize adversaries that are generated with our targeted attack and untargeted attack, respectively with dark blue.

[7] https://github.com/kohpangwei/influence-release