

# Decoupled Adversarial Contrastive Learning for Self-supervised Adversarial Robustness

Chaoning Zhang<sup>\*1</sup>, Kang Zhang<sup>\*1</sup>, Chenshuang Zhang<sup>1</sup>, Axi Niu<sup>2</sup>, Jiu Feng<sup>3</sup>,  
Chang D. Yoo<sup>1</sup>, and In So Kweon<sup>1</sup>

<sup>1</sup> Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea  
chaoningzhang1990@gmail.com, zhangkang@kaist.ac.kr

<sup>2</sup> Northwestern Polytechnical University, Xi'an, China

<sup>3</sup> Sichuan University, Chengdu, China

**Abstract.** *Adversarial training* (AT) for robust representation learning and *self-supervised learning* (SSL) for unsupervised representation learning are two active research fields. Integrating AT into SSL, multiple prior works have accomplished a highly significant yet challenging task: learning robust representation without labels. A widely used framework is adversarial contrastive learning which couples AT and SSL, and thus constitutes a very complex optimization problem. Inspired by the divide-and-conquer philosophy, we conjecture that it might be simplified as well as improved by solving two sub-problems: non-robust SSL and pseudo-supervised AT. This motivation shifts the focus of the task from seeking an optimal integrating strategy for a coupled problem to finding sub-solutions for sub-problems. With this said, this work discards prior practices of directly introducing AT to SSL frameworks and proposed a two-stage framework termed Decoupled Adversarial Contrastive Learning (DeACL). Extensive experimental results demonstrate that our DeACL achieves SOTA self-supervised adversarial robustness while significantly reducing the training time, which validates its effectiveness and efficiency. Moreover, our DeACL constitutes a more explainable solution, and its success also bridges the gap with semi-supervised AT for exploiting unlabeled samples for robust representation learning. The code is publicly accessible at <https://github.com/pantheon5100/DeACL>.

**Keywords:** Adversarial Contrastive Learning, Adversarial Training, Self-supervised Learning, Adversarial Robustness

## 1 Introduction

Despite the phenomenal success in a wide range of applications [27, 30, 64, 60], deep neural networks (DNNs) are widely recognized to be vulnerable to adversarial examples [48, 21]. Adversarial training (AT) and its variants have become the de facto standard approach for learning an adversarially robust model [36, 67]. AT targets robust generalization [46] which requires more data than standard training. In practice, however, samples with ground-truth (GT) labels are

---

<sup>\*</sup> Equal Contribution.

much more difficult to obtain than their unlabeled counterparts. To partly or fully remove the dependence on human annotation, unlabeled samples can be exploited for learning robust representation.

Multiple works [50, 7, 59, 38] have independently shown that unlabeled samples improve adversarial robustness in the semi-supervised setting. The performance of such semi-supervised AT, however, is often reported to be poor when only a small amount of labelled samples are available. Therefore, an interesting question is whether reasonable robustness can be achieved with *only* unlabeled samples. The past few years have witnessed substantial progress in the field of self-supervised learning (SSL) [10, 25, 12] for representation learning without GT labels. Inspired by such progress, multiple works [56, 31, 19, 32] have shown the success of adversarial contrastive learning (CL) for achieving robustness without labels, which constitutes a positive answer to the above question.

Nonetheless, robust SSL has been often recognized as a challenging problem due to its two mixed challenging goals: (a) *unsupervised* representation learning; (b) *robust* representation learning. The first goal can be readily realized by SOTA SSL frameworks, such as contrastive learning (CL)-based SimCLR [10], MoCo [26], while AT constitutes a go-to solution for the second goal. Thus, a line of works [56, 32, 31, 19] choose a natural strategy by introducing AT into SimCLR or MoCo to perform adversarial CL. Despite having such off-the-shelf solutions for both SSL and AT, how to effectively integrate the two techniques as an optimal solution remains not fully clear. Searching for such an optimal combining strategy is non-trivial because the two goals are entangled in the optimization. Moreover, SSL and AT often require different configuration choices for their respective goals, and combining them inevitably involves a trade-off between them. Inspired by the design philosophy of the divide-and-conquer algorithm, we conjecture that the task might be simplified by solving two sub-problems in a decoupled manner. This frustratingly simple motivation brings a fundamental shift for the focus of robust SSL: from *seeking an optimal combining strategy for a coupled problem* to *finding sub-solutions for sub-problems*.

To this end, this work discards the prior practice [56, 32, 31, 19] of introducing AT to SSL frameworks and proposes a new two-stage framework termed Decoupled Adversarial Contrastive Learning (DeACL). At stage 1, we perform standard (*i.e.* non-robust) SSL to learn instance-wise representation as a target vector. At stage 2, the obtained target vectors can be used for facilitating AT in a **pseudo-supervised** manner for learning robust representation. We find that DeACL significantly benefits from the configuration for second-stage AT being set differently from that of first-stage SSL.

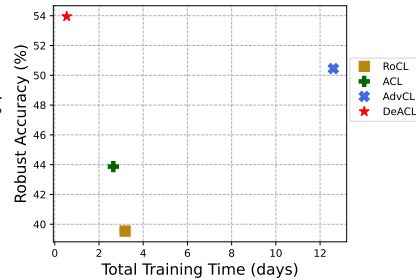


Fig. 1: Comparison of different methods on robust accuracy and total training time.

Except for enabling flexible yet simple configuration choices, another important side benefit of our DeACL is to require much fewer computation resources. At first sight, it might be counter-intuitive that two-stage approaches can be much faster than single-stage counterparts. Ignoring specific implementation details, the basic rationale is as follows. SSL typically requires  $M$  times more iterations than their supervised counterpart, and AT is often  $N$  times slower than standard training. Compared with supervised standard training, combining them into a single-stage makes it  $M \times N$  times slower, while our DeACL makes it only  $M + N$  times slower due to the disentangling effect. As shown in Figure 1, with SimCLR as the baseline SSL, our DeACL achieves state-of-the-art robustness while significantly reducing the required training time. The superior performance of our approach is also confirmed under adversarial full fine-tuning.

Overall, this work studies self-supervised robust representation learning. We summarize the contributions as follows:

- In contrast to existing works seeking an optimal strategy for combining SSL and AT to achieve robust SSL, our work investigates a different approach by solving two sub-problems in a divide-and-conquer manner, which yields a novel two-stage DeACL framework for robust SSL.
- The proposed DeACL has two advantages: (a) enabling flexible configuration for the two sub-problems; (b) requiring much fewer computation resources. Extensive experiments demonstrate that DeACL achieves SOTA robustness while significantly reducing the training time.
- Our DeACL also constitutes a more explainable solution for robust SSL and its success also bridges the gap with semi-supervised AT for exploiting unlabeled samples for robust representation learning.

## 2 Related works

The task of robust SSL lies in the intersection between SSL and AT to learn robust feature representation without GT labels. SSL and AT are two active research fields, for which we summarize their recent progress.

**Development in SSL.** The success of SSL has been demonstrated in a wide range of applications, ranging from natural language processing [33, 43, 16, 47, 39] to more recent vision tasks [34, 13, 17]. Without the need for GT labels annotated by the human, early SSL approaches leverage handcrafted “pretext” tasks, like solving Jigsaw puzzle [20] or predicting image rotation [40], while recent methods seek to learn augmentation-invariant representation [2, 26, 10, 8, 24]. To make the encoder augmentation-invariant, a commonly adopted practice is to minimize the distance between a positive pair, *i.e.* two views augmented from the same image based on a Siamese network architecture. A widely known issue in SSL is that the network might output an undesired constant, for which contrastive learning (CL) provides a satisfactory solution by maximizing the distance between negative samples, *i.e.* views of different images. CL has been widely investigated in [41, 29, 53, 69, 2, 28, 49, 10, 26, 51, 52, 57], contributing to

the progress of SSL. Recently, multiple works [12, 24, 18, 58, 3] have also explored non-contrastive SSL. A unified perspective on contrastive and non-contrastive SSL is provided in [66, 65].

**Development in AT.** To improve adversarial robustness, early works have attempted with various image processing or detection techniques, most of which, however, are found to give a false sense of robustness [6, 1, 15]. Currently, AT and its variants are widely recognized as powerful solutions to improve model robustness, among which Mardy-AT [36] and Trades-AT [67] are two widely used baselines. From the perspective of model architecture, AT often requires a larger model capacity [50, 55]. Moreover, [54, 42] have found that a smooth activation function, like parametric softplus, is often but not always [23] helpful for AT. From the perspective of tricks, [42] has performed a comprehensive evaluation for bags of tricks in AT and found that most of them provide no or trivial performance boost over Mardy-AT and Trades-AT if basic hyperparameters, such as weight decay, are set to proper values. From the perspective of data, [50, 7, 68] have shown that unlabeled data can be helpful for robustness improvement over a basic supervised baseline. However, those approaches still depend on a large amount of labeled samples. For example, [50, 7, 59] have shown that robust accuracy drops significantly when only 10% of the CIFAR10 labels are available. Universal AT [5] has also been investigated for defending against universal adversarial perturbations [37, 61, 4, 63, 62].

**Self-supervised adversarial robustness.** Clearly, self-supervised AT, *i.e.* achieving robustness with only unlabeled samples, can be even more challenging than the semi-supervised AT setting. Nonetheless, multiple recent works [9, 31, 32, 22, 19] have demonstrated encouraging success in this challenging yet highly significant direction. Prior attempts mainly focused on finding effective techniques to combine SSL and AT. What differentiates our approach from prior attempts [56, 31, 32, 22, 19] lies in disentangling robust SSL into two decoupled sub-problems (SSL and AT) which can be solved in two stages. In the following section, we will detail existing single-stage frameworks as well as the motivation behind our two-stage framework.

### 3 Proposed method

To avoid ambiguity, we start by presenting the problem of our interest, *i.e.* robust SSL, and common fine-tuning methods for evaluating the learned robust representation. Then, we briefly summarize how prior attempts [9, 31, 32, 22, 19] solve this problem in a single-stage framework. Compared with standard supervised training, either SSL or AT makes the optimization more complex, while simultaneously realizing SSL and AT clearly makes the problem complexity to an even higher level thus is difficult to solve. Inspired by the philosophy of the divide-and-conquer algorithm, we divide the complex robust SSL problem into two sub-problems: non-robust SSL and pseudo-supervised AT, and sequentially conquer them. We identify multiple important details that need to be configured differently for AT at stage 2 from standard SSL at stage 1.

### 3.1 Problem statement

**Robust SSL.** The goal of robust SSL is to learn robust feature representation with only unlabeled samples so that the model can be trained by a self-supervision loss, such as InfoNCE in CL-based SSL frameworks [26, 10]. Note that this is different from a semi-supervised setting, where labeled samples are used together with unlabeled dataset. By contrast, robust SSL *exclusively only* utilizes unlabeled dataset.

**Standard linear finetuning.** For quantitatively evaluating the learned representation, a common practice is to train a linear classifier (denoted as  $\phi_{\theta_c}$ ) on top of the pretrained encoder (denoted as  $f_{\theta_e}$ ) as:

$$\text{SLF: } \min_{\theta_c} \mathbb{E}_{(x,y) \in \mathcal{D}} \ell_{CE}(\phi_{\theta_c} \circ f_{\theta_e}(x), y), \quad (1)$$

where  $\ell_{CE}$  represents the supervised CE loss with GT-labels  $y$  over a certain dataset  $\mathcal{D}$ . This is often termed *standard linear finetuning* (SLF) [31, 19] since only a linear classifier is updated on Clean Examples (CEs). Training such a linear classifier allows access to the ground-truth labels; otherwise, the learned representation in the encoder cannot be evaluated. To not break the rule of the SSL task, the backward gradient can only be propagated to the linear classifier so that the pretrained encoder is fixed during the evaluation. The quality of learned robust representation is finally evaluated on the full model  $\phi_{\theta_c} \circ f_{\theta_e}$  by measuring its robust accuracy under PGD attack [36] or autoattack [15]. **Adversarial full finetuning.** Except for the above linear finetuning as the primary evaluation metric, one can also make the constraint less strict in the finetuning stage to perform adversarial fullfinetuning [32] (AFF). AFF allows the encoder to be updated during the finetuning as:

$$\text{AFF: } \min_{\theta_c, \theta_e} \mathbb{E}_{(x,y) \in \mathcal{D}} \ell_{CE}(\phi_{\theta_c} \circ f_{\theta_e}(x + \delta), y), \quad (2)$$

where  $\delta$  is adversarial perturbation. It is worth highlighting that the weight initialization from robust SSL significantly improves the convergence speed of supervised AT together with a non-trivial performance boost. Note that we do not consider standard full finetuning because it cannot generate a robust model.

**Basic setup.** Following [32, 31, 19], we adopt ResNet18 as the encoder architecture and investigate robustness on CIFAR10. Under the  $l_\infty$  constraint, we set the maximum allowable perturbation budget  $\epsilon$  to 8/255 during both training and evaluation. Following AdvCL[19], we evaluate the learned robust representation on three metrics: Standard Accuracy (SA), Robust Accuracy (RA) and Autoattack Accuracy (AA). SA is the classification evaluated on clean examples, while RA is evaluated on adversarial examples generated by 20-step PGD attacks. AA evaluates the model accuracy under Autoattack [15] for mitigating the concerns for the phenomenon of obfuscated gradient. We follow [19] for the settings of SLF and AFF (see the supplementary for a detailed setup).

### 3.2 Existing single-stage framework for robust SSL

Since contrastive learning (CL) is a widely proven effective technique in SSL for representation learning without labels, for which SimCLR [10] is a popular representative. Therefore, multiple works [31, 32, 19] have adopted SimCLR as the baseline SSL method and improved its robustness by combining it with AT. Let us briefly recap how SimCLR framework works. The optimization goal of CL is to make the anchor sample be attracted close to its positive sample, *i.e.* a different view augmented from the same image while being pushed away from its negative samples. The pipeline takes a batch of image samples as the input and processes it with a backbone encoder followed by a projector which is an MLP [10]. The output is a latent vector denoted as  $z$ . With  $\cdot$  indicating the cosine similarity between vectors and  $N$  indicating the number of negative samples, the contrastive InfoNCE is shown as:

$$\mathcal{L}_{CL} = -\log \frac{\exp(z_a \cdot z_b / \tau)}{\exp(z_a \cdot z_b / \tau) + \sum_{i=1}^N \exp(z_a \cdot z_i / \tau)}, \quad (3)$$

where  $z_a$  and  $z_b$  are a positive pair.  $\tau$  denotes the temperature hyperparameter. The negative samples are included to prevent a collapse mode where the model outputs a constant regardless of the inputs. Note that the above loss can be simplified to a cosine similarity loss by excluding negative samples.

**RoCL.** Introducing AT to the above CL, [32] is one of the pioneering works to propose a robust contrastive learning (RoCL) framework. Following the procedure in vanilla AT [36], RoCL first generates adversarial examples ( $z_a^{adv}$ ) by maximizing its cosine distance from  $z_b$  with multi-step PGD attacks and then updates the network by minimizing the cosine distance between all positive samples. In contrast to standard SSL, RoCL has three positive samples,  $z_a$ ,  $z_b$  and  $z_a^{adv}$ , which forms three contrastive losses for training the network.

**ACL.** Concurrent to RoCL [32], another work [31] proposes a similar SimCLR-based approach coined as adversarial contrastive learning (ACL). [31] has explored to improve the robustness of SimCLR with various attempts, among which a dual stream consisting of a standard2standard (S2S) and adversarial2adversarial (A2A) performs the best. S2S is a normal CL as introduced in Eq 3, while A2A replaces  $z_a$  and  $z_b$  with adversarial examples  $z_a^{adv}$  and  $z_b^{adv}$  which are generated by maximizing their cosine distance to each other.

**AdvCL.** Very recently, another SimCLR-based adversarial contrastive learning framework, which is coined as AdvCL in [19] to differentiate from ACL, has been proposed. In essence, without dual-stream design, AdvCL is more similar to RoCL than ACL. What differentiates AdvCL from them is its two distinctive components: (a) introducing another positive view which is augmented by keeping only high-frequency content; (b) adopting another supervision loss by utilizing an additional encoder pretrained on a much larger dataset (ImageNet). Empirically, these two designs improve its performance over RoCL and ACL by a large margin. As reported in [19], this performance boost is at the cost of being three times slower than RoCL and AdvCL. If the pretraining time on ImageNet is considered, the required computation resources can be more intimidating.

### 3.3 Decomposed Adversarial Contrastive Learning

**Motivation.** Divide-and-conquer is a widely used algorithm paradigm in ML to break down a complex problem into two (or more) sub-problems which can be easier to solve. Inspired by such design philosophy, we propose to divide the complex robust SSL into two sub-problems, *i.e.* (a) (non-robust) SSL and (b) (pseudo-)supervised AT. Such a decoupled optimization procedure simplifies the robust SSL by shifting the task focus from seeking an optimal strategy to combine SSL and AT to finding sub-solutions to sub-problems. Overall, with the motivation to decompose robust SSL, we propose a new two-stage framework, termed DeACL. For differentiation, we denote the encoder at stage 1 as  $f_{\theta_1}$  and that at stage 2  $f_{\theta_2}$ .

**Stage 1: non-robust SSL for optimizing  $f_{\theta_1}$ .** Following [31, 32, 19], this work mainly adopts SimCLR as the SSL method. Following [14], we train the model for 1000 epochs. A detailed setup is listed in the supplementary. The purpose of non-robust SSL is to obtain label-alike pseudo-targets for guiding the following pseudo-supervised AT.

**Stage 2: pseudo-supervised AT for optimizing  $f_{\theta_2}$ .** In vanilla supervised AT, the model training is guided by GT labels. Conceptually, the term “label” is often associated with human predefined classes, cat or dog for instance, which do not exist in the SSL. Thus, the representation vectors obtained from SSL are termed *targets* to differentiate from *labels*. Moreover, since the targets are generated by a SSL pretrained model instead of human annotation, we term them *pseudo-targets*. Specifically, the pseudo-targets refer to the instance-wise representation vectors by feeding the samples to a pretrained backbone encoder. They serve a similar role as GT labels to guide the supervised AT.

**Loss design.** We use the default SSL loss (Eq 3 for instance) to optimize  $f_{\theta_1}$  at stage 1 of our DeACL. At stage 2, we optimize the encoder  $f_{\theta_2}$  with the loss as:

$$L_{stage2} = CosSim(f_{\theta_2}(x), z_1) + \lambda CosSim(f_{\theta_2}(x^{adv}), f_{\theta_2}(x)), \quad (4)$$

where  $CosSim$  indicates cosine similarity loss and  $z_1$  indicates target vector generated from the pretrained  $f_{\theta_1}$ . The adversarial example  $x^{adv}$  is generated by maximizing  $CosSim(f_{\theta_2}(x^{adv}), z_1)$ . Following AdvC [19], 5-step PGD (with the step size  $\alpha = 2/255$ ) is adopted to generate  $x^{adv}$ . Eq 4 consists of two terms where the first one increases accuracy and the second one acts as a regularization loss to increase robustness.  $\lambda$  is a hyper-parameter for achieving a trade-off between accuracy and robustness. In this work, we set  $\lambda$  to 2 (see supplementary for the ablation study). This design is inspired by a SOTA loss in (supervised) Trades-AT [67]. A major difference from [67] is that we use  $CosSim$  instead of KL divergence to measure the distance. We empirically find that  $CosSim$  outperforms KL by a large margin. An alternative loss could be designed to directly minimize  $CosSim(f_{\theta_2}(x^{adv}), z_1)$ . Its performance is worse than Eq 4 (see the supplementary for ablation study), which aligns with the finding in prior works [67, 42].

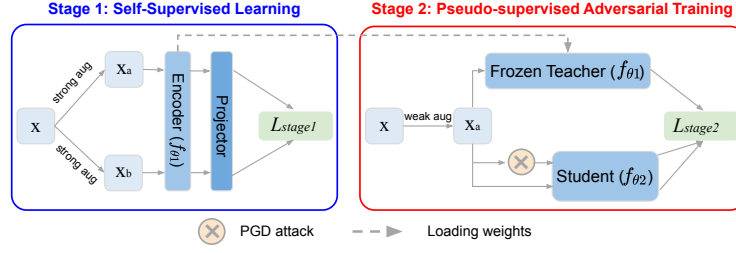


Fig. 2: Overall framework of DeACL. It consists of two stages. At stage 1, DeACL performs a standard SSL to obtain a non-robust encoder. At stage 2, the pre-trained encoder act as a teacher model to generate pseudo-targets for guiding a supervised AT on a student model. After two stages of training, the student model is the model of our interest.

**Overall framework.** The overall framework of our DeACL is shown in Figure 2, where the pretrained encoder in stage 1 is loaded and then frozen in stage 2 as a teacher model for generating pseudo-targets.

At stage 2, we can initialize  $f_{\theta_2}$  with random weights or pretrained weights by loading  $f_{\theta_1}$  to the student model. Empirically, we find that this only yields a small performance variation as shown in Table 1. In the beginning, loading pretrained (non-robust) weights boosts the convergence speed (see the supplementary), which is well expected. Since this convergence and performance boost is free, our DCL by default adopts this practice.

Table 1: Influence of weight initialization for the student model.

|                 | AA    | RA    | SA    |
|-----------------|-------|-------|-------|
| DeACL           | 45.31 | 53.95 | 80.17 |
| Student Scratch | 44.63 | 54.06 | 79.47 |

## 4 Advantages of our DeACL

### 4.1 Flexible configuration for SSL and AT

It has been shown in [42] that training configurations (*e.g.*, weight decay) can have a significant influence on robustness in supervised AT. However, it is not clear how to choose the optimal configurations in robust SSL, especially considering the differences of main configurations in SSL and supervised AT, as shown in Table 2. Since recent works [32, 31, 19] all train the model in a single stage, it is not clear whether configurations of SSL or supervised AT should be applied. RoCL and ACL both follow the configuration as standard SSL, such as small weight decay, strong augmentation, adopting a projector head, and preventing collapse with InfoNCE loss. For the weight decay, AdvCL adopts a value that lies between those for SSL and AT. For augmentation, AdvCL adopts strong augmentation on CEs and weak augmentation on AEs. [19] shows that their AdvCL performance drops by a large margin if strong augmentation is applied



on AEs. A major drawback for existing single-stage frameworks lies in seeking an optimal configuration that simultaneously fits both SSL and AT.

Table 2: Summarization of four configuration details in various settings.

| Configuration       | SSL    | AT   | Single-stage frameworks |        |             | Two-stage DeACL |      |
|---------------------|--------|------|-------------------------|--------|-------------|-----------------|------|
|                     |        |      | RoCL                    | ACL    | AdvCL       | SSL             | AT   |
| Weight decay        | 1e-5   | 5e-4 | 1e-6                    | 1e-6   | 1e-4        | 1e-5            | 5e-4 |
| Data augmentation   | Strong | Weak | Strong                  | Strong | Strong/weak | Strong          | Weak |
| Projector head      | Yes    | No   | Yes                     | Yes    | Yes         | Yes             | No   |
| Collapse prevention | Yes    | No   | Yes                     | Yes    | Yes         | Yes             | No   |

By contrast, our DeACL enables stage-specific configuration. In other words, we can freely choose the optimal experimental configurations during each stage. For the first stage which aims to train a standard model, we follow the open source SSL library [14] for optimal configuration. In the following, we detail why and how each configuration is set at stage 2 of our DeACL.

**Weight decay.** SSL typically has a strong regularization effect due to strong augmentation and the weight decay is often set to a relatively small value, 1e-5 for instance [14]. Supervised AT, however, often suffers from robust overfitting [44] and requires a large weight decay [42]. [42] has performed an extensive study on a bag of tricks on supervised AT, and has found that weight decay is the most significant factor, which has a much higher influence than tricks reported in most works. With the weight decay set to 5e-4, [42] has shown that most tricks bring no or marginal performance boost over the widely used Madry-AT [36] and Trades-AT [67]. Table 3 shows that a relatively large weight, *i.e.* 5e-4, is required for achieving high robustness and accuracy. Note that [42] also shows that 5e-4 is the optimal weight decay for vanilla supervised AT on the same dataset CIFAR10.

**Data augmentation.** SSL is widely known to require strong augmentation to learn augmentation-invariant representation, while supervised training (either standard or adversarial one) typically often adopts weak augmentation. On the CIFAR10 dataset, the strong augmentation consists of random resized crop, color jittering, color change, Gaussian blur, solarization and horizontal crop, while the weak augmentation only consists of random crop (after padding) and horizontal flip. The results in 4 show that applying strong aug-

Table 3: Influence of weight decay at stage 2 with the SSL results on CIFAR10.

| Weight Decay | SLF   |       |       |
|--------------|-------|-------|-------|
|              | AA    | RA    | SA    |
| 1e-6         | 34.77 | 40.45 | 81.88 |
| 1e-5         | 36.33 | 44.21 | 80.29 |
| 1e-4         | 43.26 | 52.39 | 80.21 |
| 5e-4         | 45.31 | 53.95 | 80.17 |
| 1e-3         | 43.07 | 53.32 | 78.24 |
| 5e-3         | 24.80 | 33.45 | 55.76 |

Table 4: Influence of data augmentation at stage 2 with the SSL results on CIFAR10.

| Augmentation |        | SLF   |       |       |
|--------------|--------|-------|-------|-------|
|              |        | AA    | RA    | SA    |
| Weak         | Weak   | 45.31 | 53.95 | 80.17 |
| Strong       | Strong | 6.93  | 17.43 | 48.12 |
| Weak         | Strong | 34.47 | 45.45 | 74.85 |
| Strong       | Weak   | 7.62  | 18.36 | 79.93 |

mentation on either clean examples (CEs) or adversarial examples (AEs) in Eq 4 yields significantly inferior performance.

**Projector head.** Projector head has become a de facto standard component in SSL to be added after the backbone during training for performance boost [10, 11, 45]. After the training is done, only the backbone is kept for the downstream task. In the case of supervised AT [36], no projector is used during training. The results in Table 5 show that adding the projector at stage 2 of our DeACL is harmful to both robustness and accuracy.

**Collapse prevention.** A widely known phenomenon in SSL is that the model output a constant output, *i.e.* collapse, if the loss only maximizes the cosine similarity between a pair of positive samples. A widely used

approach to mitigate this phenomenon is to introduce a contrastive component, *i.e.* simultaneously minimizing the cosine similarity between negative samples (See Eq 3). The results in Table 6 show that adding a contrastive component decreases the performance by a large margin. At stage 2 of our DeACL, there is low or no risk of collapse because it is supervised by distinctive pseudo-targets.

**Takeaway on the configuration.** Overall, our above investigation shows that our DeACL significantly benefits from the fact that the configurations for AT in our DeACL can be set differently from those in SSL. The best configurations at stage 2 of our DeACL are the same as those in supervised AT, which is reasonable considering the second stage our DeACL conducts a *i.e.* pseudo-supervised AT. As shown in Table 7, thanks to the flexible configuration, our DeACL achieves superior performance over existing single-stage frameworks. Notably, our proposed DeACL achieves the highest robustness for both AA and RA on both CIFAR10 and CIFAR100. For SA, our DeACL outperforms all existing methods except for AdvCL on CIFAR10. On CIFAR100, our DeACL outperforms all existing methods by a large margin.

Performing a similar investigation on existing single-stage frameworks might also bring a performance boost. However, it is not guaranteed considering the configuration trade-off between SSL and AT. Moreover, such a search for optimal configuration in single-stage frameworks can be intimidating if taking computation resources into account. In the following, we discuss another advantage of our DeACL for significantly reducing the training time.

## 4.2 Two-stage DeACL is faster than single-stage frameworks

As shown in Table 7, among the three single-stage frameworks (RoCL, ACL, AdvCL), the very recent AdvCL achieves the best robustness but at the cost of

Table 5: Influence of projector head at stage 2 with the SSL results on CIFAR10.

|               | AA    | RA    | SA    |
|---------------|-------|-------|-------|
| w/o projector | 45.31 | 53.95 | 80.17 |
| w/ Projector  | 42.48 | 50.60 | 78.80 |

Table 6: Influence of collapse prevention at stage 2 with the SSL results on CIFAR10.

|                         | AA    | RA    | SA    |
|-------------------------|-------|-------|-------|
| w/o collapse prevention | 45.31 | 53.95 | 80.17 |
| w/ collapse prevention  | 34.18 | 39.32 | 72.19 |

Table 7: SLF results on CIFAR10 and CIFAR100. All the methods are evaluated with ResNet18 under the same condition following [19]. We report three metrics (Auto Attack (AA), Robust Accuracy (RA), Standard Accuracy (SA)) as well as the pretraining time (in days). For all metrics, the best performance is highlighted in **bold**.

| Method | CIFAR10      |              |              | CIFAR100     |              |              | Computation resource |     |             |
|--------|--------------|--------------|--------------|--------------|--------------|--------------|----------------------|-----|-------------|
|        | AA(%)        | RA(%)        | SA(%)        | AA(%)        | RA(%)        | SA(%)        | Time                 | GPU | Total       |
| AP-DPE | 16.07        | 18.22        | 78.30        | 4.17         | 6.23         | 47.91        | 10.11                | 1   | 10.11       |
| RoCL   | 23.38        | 39.54        | 79.90        | 8.66         | 18.79        | 49.53        | 1.59                 | 2   | 3.18        |
| ACL    | 39.13        | 42.87        | 77.88        | 16.33        | 20.97        | 47.51        | 2.65                 | 1   | 2.65        |
| AdvCL  | 42.57        | 50.45        | <b>80.85</b> | 19.78        | 27.67        | 48.34        | 3.15                 | 4   | 12.60       |
| DeACL  | <b>45.31</b> | <b>53.95</b> | 80.17        | <b>20.34</b> | <b>30.74</b> | <b>52.79</b> | 0.45                 | 1   | <b>0.45</b> |

significantly more training time. As noted in [19], their superior performance is partly attributed to introducing additional views as well as additional pseudo supervision regularization from encoder pretrained on ImageNet. These two design choices are also the reason that makes their AdvCL significantly slower. Compared with them, our DeACL requires the least training time, which is mainly attributed to the effect of disentangling SSL and AT. It is worth highlighting that our DeACL achieves a significant performance boost over RoCL and ACL, without relying on additional high-frequency views or additional supervision from ImageNet pretrained models. These two techniques might further improve the performance of our DeACL, and we leave such investigation for future work. We do not include them in this work to make our DeACL simple and fast.

**Rationale for why DeACL is fast.** Given that RoCL and ACL do not use the two design choices as AdvCL, why are they still significantly slower than our DeACL? At first sight, it seems counterintuitive that the two-stage DeACL can be faster. The rationale is briefly discussed as follows. SSL and AT are both widely known to require much longer training time than their standard supervised counterpart. Specifically, SSL often requires  $M$  (10 for instance) times more training iterations (epochs) due to the lack of GT labels. AT makes the iteration-wise training time  $N$  (7 for instance) times longer because generating adversarial examples with the commonly used multi-step PGD attack is very slow. Directly solving a robust SSL requires  $M \times N$  times more training time, while disentangling them into two stages is expected to only require  $M + N$  times more training time. In practice, training time can be more than complex than the above reasoning rationale, depending on the implementation details.

## 5 Additional experimental results

The results in the above section demonstrate that our DeACL outperforms existing single-stage frameworks by a large margin while requiring significantly less

Table 8: AFF results on CIFAR10 and CIFAR100. All the methods are evaluated with ResNet18 under the same condition following [19]. For all metrics (AA, RA, SA), our DeACL achieves the best performance which is highlighted in **bold**.

| SSL-AT     | CIFAR10      |              |              | CIFAR100     |              |              |
|------------|--------------|--------------|--------------|--------------|--------------|--------------|
|            | AA(%)        | RA(%)        | SA(%)        | AA(%)        | RA(%)        | SA(%)        |
| Supervised | 46.19        | 49.89        | 79.86        | 21.61        | 25.86        | 52.22        |
| AP-DPE     | 48.13        | 51.52        | 81.19        | 22.53        | 26.89        | 55.27        |
| RoCL       | 47.88        | 51.35        | 81.01        | 22.28        | 27.49        | 55.10        |
| ACL        | 49.27        | 52.82        | 82.19        | 23.63        | 29.38        | 56.61        |
| AdvCL      | 49.77        | 52.77        | 83.62        | 24.72        | 28.73        | 56.77        |
| DeACL      | <b>50.39</b> | <b>54.18</b> | <b>83.95</b> | <b>25.48</b> | <b>29.65</b> | <b>59.86</b> |

training time. Here, we further conduct extra experiments to verify the effectiveness of our approach from different angles.

**AFF results.** Table 8 reports the AFF results on both CIFAR10 and CIFAR100. Compared to the results in Table 7, for all methods, AFF brings a consistent performance boost over SLF, which is expected since AFF also allows the encoder to be updated. Similar to the trend with SLF, we observe that our DeACL achieves SOTA performance for all the three considered metrics on both CIFAR10 and CIFAR100.

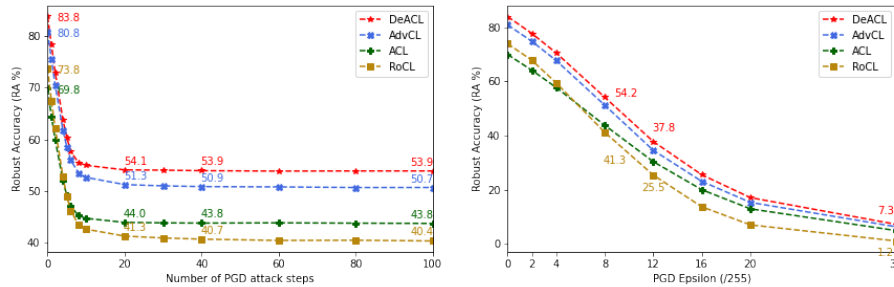


Fig. 3: RA on CIAR10 with various PGD steps and  $\epsilon$ . Our DeACL consistently achieves the best performance.

**Influence of attack steps and perturbation magnitude.** For the RA, by default we use 20-step PGD, *i.e.* PGD-20, with  $\epsilon$  set to  $l_\infty$  8/255. Here, we evaluate with various steps and  $\epsilon$  values. The results in Figure 3 show that our DeACL consistently outperforms existing methods by a non-trivial margin.

**Qualitative results.** With t-SNE [35], the visualization of learned representation on CIFAR10 is shown in Figure 4, where each point is colored by its GT label. The class boundary of our DeACL is clearer than that of existing methods, which suggests that DeACL might be more robust to adversarial perturbation.

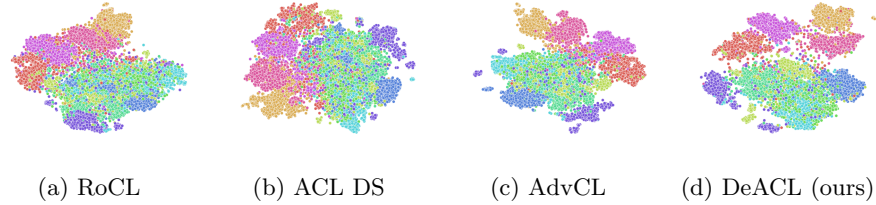


Fig. 4: Visualization of robust representation learned with different methods. Our method gives a more clear classification boundary than existing methods.

## 6 Implications of our findings

### 6.1 Towards a more explainable solution

It is desirable to have a more explainable solution for a given task. For the task of robust SSL, the success of existing solutions based on a single-stage framework is much more difficult to explain because it couples two sub-problems. Our DeACL decouples the task, which makes the solution significantly more explainable. For example, an interesting question to ask in robust SSL is how much the SSL framework choice influences the performance. With the existing single-stage framework, such influence is much more difficult to analyze due to its interaction with AT. Note that SSL framework often has its optimal configuration setting. As we can see from Section 4.1, such configuration detail can have a significant influence when considering AT. With our DeACL, we can adopt a unified configuration in the second-stage AT to exclude such influence of configuration on AT. With this said, Table 9 reports the influence of SSL frameworks. We observe an interesting phenomenon that CL-based frameworks tend to outperform non-CL-based frameworks for achieving higher robustness but possibly at the cost of slightly lower accuracy. [56] claims that adversarial momentum contrastive learning (AMOC) outperforms adversarial contrastive learning by showing a non-trivial performance gain of their AMOC over ACL. Since many configurations in MoCov2 and SimCLR are different and many other details like how to generate adversarial examples are also very different, their conclusion might be not fully convincing. Our results in Table 9 show that MoCov2 and SimCLR achieve comparable RA and SA, suggesting their non-trivial performance boost is likely to be caused by the influence of different configurations on AT.

Table 9: Results on Cifar10 with various SSL frameworks at stage 1 of our DeACL.

| SSL frameworks   | SLF   |       |       |
|------------------|-------|-------|-------|
|                  | AA    | RA    | SA    |
| SimCLR [10]      | 45.31 | 53.95 | 80.17 |
| MoCo v2 [11]     | 46.29 | 53.97 | 80.56 |
| BYOL [24]        | 44.14 | 52.42 | 80.89 |
| BarlowTwins [58] | 41.31 | 50.47 | 80.88 |
| VICReg [3]       | 43.65 | 50.56 | 82.10 |

## 6.2 Towards a unified perspective on semi/self-supervised AT

It is worth noting that a similar two-stage approach is also used in semi-supervised AT for exploiting unlabeled dataset. the success of such a two-stage approach in both semi-supervised and self-supervised settings suggests a new unified perspective on how to effectively exploit unlabelled dataset for learning robust representation. Intuitively, in the semi-supervised settings, the unlabeled dataset can also be used to improve the robustness in a single-stage manner through a regularization loss for instance. However, [50] has found that such a single-stage approach achieves inferior performance than the two-stage approach. Given that our two-stage DeACL also outperforms existing single-stage SOTA baselines in self-supervised setting, it suggests a unified understanding on semi-supervised and self-supervised AT: pseudo-targets (either pseudo-labels or pseudo-vectors) are all you need for exploiting unlabeled dataset to learn robust representation.

Despite such a unified perspective, it is also important to note distinctions between them. Their core distinction lies in their different motivations. A semi-supervised setting allows access to labeled datasets and the motivation of using unlabeled images is to use more samples. Note that more samples are used for training a robust model at stage 2 of semi-supervised AT than those used at stage 1 for training a non-robust model. By contrast, our DeACL in the self-supervised setting is *not* motivated to increase the sample size and the second stage uses the same number of samples as those at stage 1. Instead, the motivation of the two-stage procedure in our DeACL lies in decomposing the robust SSL. Moreover, what connects the two stages is the supervision targets which are pseudo-labels and pseudo-vectors in semi-supervised and self-supervised settings, respectively. Due to this difference, at stage 2, our DeACL needs to adopt the cosine similarity loss instead of the commonly adopted CE loss.

## 7 Conclusion

This work revisits the task of robust SSL for learning robust representation without labels. Discard the practice of seeking an optimal strategy to combine SSL and AT, we propose a novel two-stage framework termed DeACL. Our DeACL enables independent configuration for SSL and AT for achieving SOTA robustness by using significantly smaller training resources. Extensive results confirm the effectiveness and efficiency of our DeACL over existing single-stage frameworks by a significant margin. Our findings also have non-trivial implications for pushing (a) towards a more explainable solution for robust SSL and (b) towards a unified perspective of understanding on semi/self-supervised AT regarding how to effectively exploit unlabeled samples for robust representation learning.

**Acknowledgments:** This work was partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government(MSIT) (No. 2022R1A2C201270611), and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. 2022-0-00951, Development of Uncertainty-Aware Agents Learning by Asking Questions).

## References

1. Athalye, A., Carlini, N., Wagner, D.: Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In: ICML (2018)
2. Bachman, P., Hjelm, R.D., Buchwalter, W.: Learning representations by maximizing mutual information across views. NeurIPS (2019)
3. Bardes, A., Ponce, J., LeCun, Y.: Vicreg: Variance-invariance-covariance regularization for self-supervised learning. arXiv preprint arXiv:2105.04906 (2021)
4. Benz, P., Zhang, C., Imtiaz, T., Kweon, I.S.: Double targeted universal adversarial perturbations. In: ACCV (2020)
5. Benz, P., Zhang, C., Karjauv, A., Kweon, I.S.: Universal adversarial training with class-wise perturbations. ICME (2021)
6. Carlini, N., Wagner, D.: Adversarial examples are not easily detected. In: ACM Workshop on Artificial Intelligence and Security (2017)
7. Carmon, Y., Ragunathan, A., Schmidt, L., Liang, P., Duchi, J.C.: Unlabeled data improves adversarial robustness. NeurIPS (2019)
8. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. arXiv preprint arXiv:2006.09882 (2020)
9. Chen, T., Liu, S., Chang, S., Cheng, Y., Amini, L., Wang, Z.: Adversarial robustness: From self-supervised pre-training to fine-tuning. In: CVPR (2020)
10. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: ICML (2020)
11. Chen, X., Fan, H., Girshick, R., He, K.: Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297 (2020)
12. Chen, X., He, K.: Exploring simple siamese representation learning. In: CVPR (2021)
13. Chen, X., Xie, S., He, K.: An empirical study of training self-supervised vision transformers. ICCV (2021)
14. da Costa, V.G.T., Fini, E., Nabi, M., Sebe, N., Ricci, E.: Solo-learn: A library of self-supervised methods for visual representation learning. JMLR (2022)
15. Croce, F., Hein, M.: Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In: ICML (2020)
16. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (2019)
17. El-Nouby, A., Touvron, H., Caron, M., Bojanowski, P., Douze, M., Joulin, A., Laptev, I., Neverova, N., Synnaeve, G., Verbeek, J., et al.: Xcit: Cross-covariance image transformers. arXiv preprint arXiv:2106.09681 (2021)
18. Ermolov, A., Siarohin, A., Sangineto, E., Sebe, N.: Whitening for self-supervised representation learning. In: ICML. PMLR (2021)
19. Fan, L., Liu, S., Chen, P.Y., Zhang, G., Gan, C.: When does contrastive learning preserve adversarial robustness from pretraining to finetuning? NeurIPS (2021)
20. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations. ICLR (2018)
21. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: ICLR (2015)

22. Goyal, S., Huang, P.S., van den Oord, A., Mann, T., Kohli, P.: Self-supervised adversarial robustness for the low-label, high-data regime. In: ICLR (2021)
23. Goyal, S., Qin, C., Uesato, J., Mann, T., Kohli, P.: Uncovering the limits of adversarial training against norm-bounded adversarial examples. arXiv preprint arXiv:2010.03593 (2020)
24. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al.: Bootstrap your own latent—a new approach to self-supervised learning. *Advances in Neural Information Processing Systems* (2020)
25. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. arXiv preprint arXiv:1911.05722 (2019)
26. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: CVPR (2020)
27. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
28. Henaff, O.: Data-efficient image recognition with contrastive predictive coding. In: ICML (2020)
29. Hjelm, R.D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., Bengio, Y.: Learning deep representations by mutual information estimation and maximization. arXiv preprint arXiv:1808.06670 (2018)
30. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: CVPR (2017)
31. Jiang, Z., Chen, T., Chen, T., Wang, Z.: Robust pre-training by adversarial contrastive learning. *NeurIPS* (2020)
32. Kim, M., Tack, J., Hwang, S.J.: Adversarial self-supervised contrastive learning. arXiv preprint arXiv:2006.07589 (2020)
33. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R.: Albert: A lite bert for self-supervised learning of language representations. In: ICLR (2020)
34. Li, C., Yang, J., Zhang, P., Gao, M., Xiao, B., Dai, X., Yuan, L., Gao, J.: Efficient self-supervised vision transformers for representation learning. arXiv preprint arXiv:2106.09785 (2021)
35. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. *Journal of machine learning research* (2008)
36. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: ICLR (2018)
37. Moosavi-Dezfooli, S.M., Fawzi, A., Fawzi, O., Frossard, P.: Universal adversarial perturbations. In: CVPR (2017)
38. Najafi, A., Maeda, S.i., Koyama, M., Miyato, T.: Robustness to adversarial perturbations in learning from incomplete data. *NeurIPS* (2019)
39. Nie, P., Zhang, Y., Geng, X., Ramamurthy, A., Song, L., Jiang, D.: Dc-bert: Decoupling question and document for efficient contextual encoding. In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (2020)
40. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: ECCV (2016)
41. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)
42. Pang, T., Yang, X., Dong, Y., Su, H., Zhu, J.: Bag of tricks for adversarial training. arXiv preprint arXiv:2010.00467 (2020)
43. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. *OpenAI blog* (2019)



44. Rice, L., Wong, E., Kolter, Z.: Overfitting in adversarially robust deep learning. In: ICML (2020)
45. Richemond, P.H., Grill, J.B., Alth  , F., Tallec, C., Strub, F., Brock, A., Smith, S., De, S., Pascanu, R., Piot, B., et al.: Byol works even without batch statistics. arXiv preprint arXiv:2010.10241 (2020)
46. Schmidt, L., Santurkar, S., Tsipras, D., Talwar, K., Madry, A.: Adversarially robust generalization requires more data. In: NeurIPS (2018)
47. Su, W., Zhu, X., Cao, Y., Li, B., Lu, L., Wei, F., Dai, J.: {VL}-{bert}: Pre-training of generic visual-linguistic representations. In: ICLR (2020)
48. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
49. Tian, Y., Krishnan, D., Isola, P.: Contrastive multiview coding. In: ECCV 2020 (2020)
50. Uesato, J., Alayrac, J.B., Huang, P.S., Stanforth, R., Fawzi, A., Kohli, P.: Are labels required for improving adversarial robustness? NeurIPS (2019)
51. Wang, T., Isola, P.: Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In: ICML (2020)
52. Wang, X., Zhang, R., Shen, C., Kong, T., Li, L.: Dense contrastive learning for self-supervised visual pre-training. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR) (2021)
53. Wu, Z., Xiong, Y., Yu, S.X., Lin, D.: Unsupervised feature learning via non-parametric instance discrimination. In: CVPR (2018)
54. Xie, C., Tan, M., Gong, B., Yuille, A., Le, Q.V.: Smooth adversarial training. arXiv preprint arXiv:2006.14536 (2020)
55. Xie, C., Yuille, A.: Intriguing properties of adversarial training at scale. ICLR (2020)
56. Xu, C., Yang, M.: Adversarial momentum-contrastive pre-training. arXiv preprint arXiv:2012.13154 (2020)
57. Yeh, C.H., Hong, C.Y., Hsu, Y.C., Liu, T.L., Chen, Y., LeCun, Y.: Decoupled contrastive learning. arXiv preprint arXiv:2110.06848 (2021)
58. Zbontar, J., Jing, L., Misra, I., LeCun, Y., Deny, S.: Barlow twins: Self-supervised learning via redundancy reduction. ICML (2021)
59. Zhai, R., Cai, T., He, D., Dan, C., He, K., Hopcroft, J., Wang, L.: Adversarially robust generalization just requires more unlabeled data. arXiv preprint arXiv:1906.00555 (2019)
60. Zhang, C., Benz, P., Argaw, D.M., Lee, S., Kim, J., Rameau, F., Bazin, J.C., Kweon, I.S.: Resnet or densenet? introducing dense shortcuts to resnet. In: WACV (2021)
61. Zhang, C., Benz, P., Imtiaz, T., Kweon, I.S.: Understanding adversarial examples from the mutual influence of images and perturbations. In: CVPR (2020)
62. Zhang, C., Benz, P., Karjauv, A., Kweon, I.S.: Data-free universal adversarial perturbation and black-box attack. In: ICCV (2021)
63. Zhang, C., Benz, P., Karjauv, A., Kweon, I.S.: Universal adversarial perturbations through the lens of deep steganography: Towards a fourier perspective. AAAI (2021)
64. Zhang, C., Rameau, F., Lee, S., Kim, J., Benz, P., Argaw, D.M., Bazin, J.C., Kweon, I.S.: Revisiting residual networks with nonlinear shortcuts. In: BMVC (2019)

- 65. Zhang, C., Zhang, K., Pham, T.X., Yoo, C., Kweon, I.S.: Dual temperature helps contrastive learning without many negative samples: Towards understanding and simplifying moco. In: CVPR (2022)
- 66. Zhang, C., Zhang, K., Zhang, C., Pham, T.X., Yoo, C.D., Kweon, I.S.: How does simsiam avoid collapse without negative samples? a unified understanding with self-supervised contrastive learning. In: ICLR (2022)
- 67. Zhang, H., Yu, Y., Jiao, J., Xing, E.P., Ghaoui, L.E., Jordan, M.I.: Theoretically principled trade-off between robustness and accuracy. In: ICML (2019)
- 68. Zhang, J., Han, B., Niu, G., Liu, T., Sugiyama, M.: Where is the bottleneck of adversarial learning with unlabeled data? arXiv preprint arXiv:1911.08696 (2019)
- 69. Zhuang, C., Zhai, A.L., Yamins, D.: Local aggregation for unsupervised learning of visual embeddings. In: ICCV (2019)