On Higher Adversarial Susceptibility of Contrastive Self-Supervised Learning

Rohit Gupta[1], Naveed Akhtar[2], Ajmal Mian[2], and Mubarak Shah[1]

[1] Center for Research in Computer Vision, UCF, Orlando FL32816, USA
rohitg@knights.ucf.edu, shah@crcv.ucf.edu
[2] Computer Science and Software Engineering, The University of Western Australia, Crawley WA 6009, Australia
{naveed.akhtar, ajmal.mian}@uwa.edu.au

Abstract. Contrastive self-supervised learning (CSL) has managed to match or surpass the performance of supervised learning in image and video classification. However, it is still largely unknown if the nature of the representation induced by the two learning paradigms is similar. We investigate this under the lens of adversarial robustness. Our analytical treatment of the problem reveals intrinsic higher sensitivity of CSL over supervised learning. It identifies the uniform distribution of data representation over a unit hypersphere in the CSL representation space as the key contributor to this phenomenon. We establish that this increases model sensitivity to input perturbations in the presence of false negatives in the training data. Our finding is supported by extensive experiments for image and video classification using adversarial perturbations and other input corruptions. Building on the insights, we devise strategies that are simple, yet effective in improving model robustness with CSL training. We demonstrate up to 68% reduction in the performance gap between adversarially attacked CSL and its supervised counterpart. Finally, we contribute to robust CSL paradigm by incorporating our findings in adversarial self-supervised learning. We demonstrate an average gain of about 5% over two different state-of-the-art methods in this domain.

Keywords: Contrastive Learning, Self-supervised Learning, Adversarial attack, Robustness, Adversarial perturbations.

1 Introduction

Deep Neural Networks (DNNs) are now widely applied for various computer vision tasks [27],[47],[31],[39],[14]. However, supervised training of DNNs requires a large amount of annotated data. Hence, self-supervised learning of high-level semantic representation of images and videos from unlabeled data is an attractive alternative. To that end, contrastive learning [8] is becoming an increasingly popular choice for self-supervised learning. It is common to pre-train DNNs using self-supervised learning on very large unlabelled datasets followed by task-specific finetuning on a smaller labelled dataset. Adversarial vulnerabilities might be introduced during this pre-training phase, which could have a cascading effect on

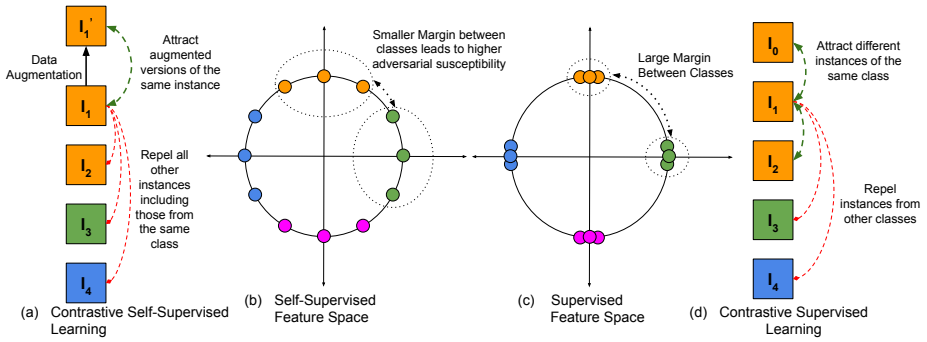


Fig. 1: (a) In Self-Supervised Contrastive Learning (CL) all instances are uniformly repelled from each other, which results in **(b)** a representation space with large class cluster sizes and small inter-class margins. **(d)** In Supervised CL only instances of different classes are repelled, which allows for **(c)** a feature space with larger inter-class margins between class clusters. Lower inter-class margins lead to higher adversarial susceptibility of self-supervised CL. Colors represent class labels for instances $I_0, I_1, ..I_N$.

a multitude of downstream tasks. Since self-supervised and supervised models achieve similar results on benchmark datasets it is a common perception that they exhibit similarities in other properties as well. It has been found that both types of models make similar errors on clean data [22]. In this work, we scrutinize this under-investigated hypothesis from the perspective of adversarial robustness. Surprisingly, our findings do not align well with the prevailing belief that both model types admit representations of similar nature. We find that Contrastive Self-supervised Learning (CSL) models are considerably inferior to supervised models in terms of adversarial robustness for image and video classification (Section 4). Interestingly, we observe that this is true even when the supervised models use contrastive loss (following [35]) instead of the more common cross-entropy loss. We show that this disparity also holds when equivalent augmentations and training schedules are applied to both supervised and self-supervised learning.

Wang et al. [50] highlighted two key properties of CSL representation. (a) *Alignment*: which enforces the closeness of instance features for the positive training pairs, and (b) *uniformity*: a property that induces uniform distribution of instances in the representation space. The latter can be understood as an application of the principle of maximum entropy [32] (colloquially referred to as Occam's Razor). Since class information about the instances is not available for self-supervised learning, we can preserve the maximum amount of information about the data in the representation by inducing a uniform distribution of training data instances. Eventually, when a classifier is trained on the uniformity preserving features, it can better separate the class instances. However, the uniform feature space necessarily has low inter-class margins relative to the supervised case where class instances can be tightly clustered - illustrated in Fig. 1. We establish that this has a negative influence on the intrinsic robustness of CSL models (Section 3).

Our investigation leads to identifying a link between the '*false negative pairs*' in the training of CSL and the higher sensitivity of the resulting model to input alteration. Hence, to improve the robustness of CSL without resorting to the

computationally expensive adversarial training, we explore multiple strategies that can easily reduce the false negative pairs during model training. These strategies are inspired by a theoretical insight that the contrastive loss in CSL still improves model representation for true positive instances despite its exposure to a few false negatives in the training process. We investigate two categories of false negative removal methods following a 'static' and a 'dynamic' strategy, which respectively remove a fixed and dynamically varying number of suspect samples from training data. After establishing the effectiveness of these methods, we also demonstrate enhancement of adversarial contrastive learning techniques [37], [33] with these strategies. Our key contributions can be summarized as:

- We provide the first systematic evidence of higher sensitivity of CSL to input perturbations in the form of rigorous analytical results and extensive experimental verification with image and video classification tasks.
- We establish a connection between CSL model susceptibility and the uniformity of its representation, and theoretically identify the influence of false negative pairs on model sensitivity.
- Leveraging theoretical insights, we devise strategies to improve CSL robustness without adversarial training.
- We also contribute to adversarial CSL by incorporating our findings into RoCL [37] and ACL [33], achieving consistent performance gain against strong adversarial attacks PGD [40] and AutoAttack [15]

2 Related Work

In the context of self-supervised deep learning, denoising auto-encoders [48] are among the first techniques, which were followed by other generative approaches, e.g., inpainting-based Context Encoder [43] and GAN based methods, such as DCGAN [46] and BiGAN [19], [20]. More recently, the literature has also witnessed pre-text task based methods for 2D CNNs [23], [18], [21] and 3D CNNs [53], [1]. However, contrastive learning methods, e.g., SimCLR [8] are currently considered the state-of-the-art. Therefore, we mainly focus on these methods due to their high relevance.

Contrastive learning: SimCLR [8] builds upon prior work of MoCo (Momentum Contrastive learning) [26], Augmented Multiscale Deep InfoMax (AMDIM) [4], and Contrastive Predictive Coding (CPC) [42] to develop its contrastive learning pipeline. The pipeline includes data augmentations and a projection head to align the learned network representation during training. While the performance of SimCLR has been lately matched or exceeded by MoCov2 [11] and SimCLRv2 [9], the fundamental structure of contrastive learning framework remains similar in these works. Contrastive learning has also been successfully extended to action classification in videos [45], [16], and image classification using transformer architectures [13].

Another relevant self-supervised learning method is SwAV [6]. Even though it does not use contrastive loss, it preserve the 'alignment' property of its representation by clustering the augmented versions of instances. Moreover, it is

also able to preserve the 'uniformity' property by enforcing an explicit equipartitioning constraint over its representation space. Other self-supervised learning methods include BYOL [25] and SimSiam [12], which are energy-based methods. These techniques do not use negative contrastive pairs, and rely on a siamese architecture. Since we are mainly concerned with the negative contrastive pairs in our analysis, these methods are not directly related to our core contribution.

Owing to the promising performance of contrastive learning, recent works have also focused on exploring the unique properties of contrastive learning. Geirhos et al. [22] found that such models produce results similar to those learned with supervision. Xiao et al. [51] found that the specific kind of data augmentation which works best for self-supervised training depends on the specific dataset. Purushwalkam et al. [44] claimed that contrastive learning results in superior occlusion-invariant representations. Wang et al. [50] analyzed contrastive learning by studying the alignment and uniformity properties of feature distribution. These properties are claimed to endow more discriminative power to the models. The uniformity property of contrastive learning is also discussed in Chen et al. [10], where it is referred to as the 'distribution' property. Wang and Liu [49] also built a relationship between the uniformity property and the temperature hyper-parameter of the loss function.

Robustness and self-supervision: In prior art on robustification of *supervised* learning, self-supervision has been considered as a helpful tool. Hendrycks et al. [29] found that adversarial robustness of supervised models can be improved by adding an additional self-supervised task in a multi-task approach. Similarly, Carmon et al. [5] also found that using additional unlabeled data improves adversarial robustness of the model. Chen et al. [7] also developed robust versions of pretext-based self-supervised learning tasks and demonstrated that this, along with robust fine-tuning of the model, results in significant increase in the robustness relative to the baseline adversarial training.

Adversarial training for self-supervised models: There has also been work on adversarial training in the context of self-supervised learning. Kim et al. [37] developed an instance-based adversarial attack for contrastive self-supervised training, and later used it during training for model robustness. The concurrent work by Jiang et al. [34] develops an adversarial contrastive learning framework that is claimed to surpass prior self-supervised learning methods in robustness as well as accuracy on clean data. Ho et al. [30] created a generalized formulation of AdvProp training [52] applicable to self-supervised learning, with the goal to increase accuracy on clean data. These methods significantly increase the training cost of the already computationally expensive learning process. Hence, in this work, we directly focus on addressing the root-cause of the issue, and later also transfer the benefits of our findings to adversarial training.

3 Adversarial Susceptibility of CSL

The popular contrastive self-supervised representation learning strategy, e.g., used by SimCLR [8], learns a representation space from unlabeled data. It samples

the so-called 'positive pairs' by applying independent random transformations to an original sample (a.k.a. anchor). The positive pairs are expected to have representations similar to the anchor. The 'negative pairs' are formed by pairing the anchor with other original instances.

Let us denote the distribution of original samples in the training data as $p_{\text{org}}(.) \in \mathbb{R}^m$. The distribution over the positive pairs can then be defined as $p_{\text{pos}}(.,.) \in \mathbb{R}^m \times \mathbb{R}^m$. In general, contrastive loss $\mathcal{L}(.)$ is defined over the sample features computed with an encoder $f(.) : \mathbb{R}^m \to \psi^{q-1}$, where '$q$' is the feature vector dimension and ψ^{q-1} identifies a hypersphere for the ℓ_2 normalized features. Analytically, the contrastive loss takes the form

$$\mathcal{L} = \mathop{\mathbb{E}}_{\substack{(u,v)\sim p_{\text{pos}} \\ \{u_i^-\}_{i=1}^N \sim p_{\text{org}}}} \Big[-\log \frac{e^{f(u)^\intercal f(v)/\tau}}{e^{f(u)^\intercal f(v)/\tau} + \sum_i e^{f(u_i^-)^\intercal f(v)/\tau}} \Big], \tag{1}$$

where $u, v \in \mathbb{R}^m$ are samples forming the positive pairs, $\{u_i^-\}_{i=1}^N \in \mathbb{R}^m$ are the corresponding negative instances, $\tau \in \mathbb{R}_+$ is a scalar (a.k.a. temperature parameter), and $N \in \mathbb{Z}_+$ is the number of negative samples.

It is shown by [50] that the loss in Eq. (1) induces the following two key properties in a representation learned under contrastive self-supervised learning. *(a) Alignment*: Features of samples in the positive pair are close on the representation hypersphere. *(b) Uniformity*: The distribution of all features is roughly uniform on the hypershpere. The property (a) is supposed to promote robustness to unintended noise by encouraging similarity between the features of similar samples. On the other hand, (b) works on the principal of preserving maximum information to improve the overall performance of the learned representation. To build our argument, we first verify these properties with an analytical simplification of Eq. (1). By definition, $p_{\text{pos}}(.,.)$ is symmetric, which lets us write:

$$\mathcal{L} = \mathop{\mathbb{E}}_{(u,v)\sim p_{\text{pos}}} [-f(u)^\intercal f(v)/\tau] + \mathop{\mathbb{E}}_{\substack{(u,v)\sim p_{\text{pos}} \\ \{u_i^-\}_{i=1}^N \sim p_{\text{org}}}} \Big[\log(e^{f(u)^\intercal f(v)/\tau} + \sum_i e^{f(u_i^-)^\intercal f(v)/\tau}) \Big]. \tag{2}$$

In Eq. (2), minimizing the first term promotes 'alignment' of the representation. Since, the summation defined over $\{u_i^-\}_{i=1}^N$ in the second term is always positive, alignment plays a significant role in reducing the contrastive loss. As the alignment improves, we approach $f(u)^\intercal f(v) \to 1$. This simplifies the loss term to

$$\mathop{\mathbb{E}}_{\substack{(u,v)\sim p_{\text{pos}} \\ \{u_i^-\}_{i=1}^N \sim p_{\text{org}}}} \Big[\log \Big(1 + \frac{\sum_i e^{f(u_i^-)^\intercal f(v)/\tau}}{e^{1/\tau}} \Big) \Big]. \tag{3}$$

In Eq. (3), the constant term $1/\tau$ is ignored due to its irrelevance. Clearly, minimizing (3) can be identified as maximizing the difference (in turn, the distance on the hypersphere) between the normalized features of the negative pairs. This promotes uniformity in the representation. Thus, a well-learned representation under contrastive loss *must* exhibit the uniformity property.

A subtle point to note is that the objective of achieving uniformity is, to an extent, contradictory to the goal of representation alignment in contrastive learning. In general, contrastive learning does not assume prior over $p_{\text{org}(.)}$. In the absence of such a prior, positive pair samples are ensured to be positive under the heuristic of 'transformation of the same sample'. However, no such heuristic exists for the negative pairs. This means, the set of negative instances $\{u_i^-\}_{i=1}^N$ can actually contain some samples that form *positive* pairs with the original sample 'v' as seen by the downstream task. To elaborate, assume the downstream task of image classification. The self-supervision mechanism may use minibatches that contain multiple images of `Ostrich`. Although minimizing the first term in Eq. (2) helps the ultimate objective of the downstream task (i.e. achieving similar representation for all `Ostrich` samples), Eq. (3) opposes that objective because it tries to spread apart the (representations of) different `Ostrich` images over the hypersphere. For a finite hypersphere, this can force a subset of `Ostrich` images to be projected close to the images of another category, e.g. `Dog`. This has implications for a downstream task like classification.

Assume we train a downstream classifier $\mathcal{C}(I_c) : I_c \rightarrow \ell$, where I_c is a sample of the c^{th} class that has the correct label ℓ. For training $\mathcal{C}(.)$, we use the representation of contrastive learning as the feature of I_c. However, for simplicity, here we directly use the symbol I_c for the feature. Our analysis above hints towards an easy identification of a transformation $\mathcal{T}(I_c) : \mathcal{C}(\mathcal{T}(I_c)) \rightarrow \tilde{\ell}$, such that $\tilde{\ell} \neq \ell$. We intentionally use an overgeneralized notion of $\mathcal{T}(.)$ here. In Section 4, we will demonstrate how even primitive input transformations can serve as $\mathcal{T}(.)$. At this stage, we are particularly interested in 'adversarial perturbation' [2] as the transformation. For an input $I_c \in \mathbb{R}^m$, an adversarial perturbation is the transformation $\mathcal{T}(I_c) = I_c + \rho, s.t.\ ||\rho||_p < \eta$, where $\rho \in \mathbb{R}^m$ is the perturbation signal whose ℓ_p-norm (denoted by $||.||_p$) is bounded by the threshold $\eta \in \mathbb{R}_+$.

Adversarial perturbations are known to easily fool the supervised models. Hence, for self-supervision, it is imperative to explore the 'weaker' perturbations to establish the higher sensitivity of contrastive self-supervision. To that end, the best available tool is Fast Gradient Sign Method (FGSM) [24]. FGSM performs a single step gradient ascend over the model loss surface w.r.t. the input and calibrates that with the sign function and a scalar multiplier. Formally, it computes ρ as: $\rho = \epsilon\ \text{sign}(\nabla\mathcal{L}(\theta, I_c, \ell))$, where ϵ is the scaling factor, θ denotes the model parameters and $\nabla(.)$ computes the gradient. Like most adversarial attack algorithms, the essence of FGSM is to estimate a direction in the input space along which the model prediction is highly sensitive. Then, it slightly nudges the input sample in that direction to fool the model on the resulting imperceptibaly altered input.

Lemma 3.1 below shows that the presence of false negative pairs in the training process of self-supervised contrastive learning makes the model even more sensitive to the nudge. It happens because under the competitive allocation of classification regions governed by a finite hypersphere ψ^{q-1}, false negatives force the model to place representations of clean samples closer to the decision

boundaries. The same insight is also applicable to the methods like SwAV, which use competitive allocation of their clustering subspace under an equipartitioning constraint. Our empirical results in Section 4 also verify the analogous higher sensitivity of SwAV against input perturbations.

In general, it is common to regularly encounter false negative pairs in self-supervised model training because the actual labels are not available for the data. Frequent occurrence of these pairs in the training process eventually leads to adversarially more sensitive models because it is easier to alter the predictions of samples residing closer to model decision boundaries [41].

Lemma 3.1: *False negatives* u^-_{false} *in* $\{u^-_i\}_{i=1}^N$ *bring the decision boundary closer to the clean samples of their classes.*

Explanation: For a representation hypersphere ψ^{q-1} of the q-dimensional features of p_{org}'s samples forming 'C' classes, the (projection of) classification region Ξ_c of the c^{th} class on the hypersphere has an $\text{Area}(\Xi_c) = \frac{\delta_c}{C}\left(2\pi^{\frac{q-1}{2}}/\Gamma(\frac{q-1}{2})\right)$. Here, $\Gamma(.)$ is the gamma function and $\delta_c \in (0, C)$. The fraction $\frac{\delta_c}{C}$ normalizes the area for 'C' equally likely classes and then scales it for 'c'. Given the fixed values for 'C' and 'q', δ_c needs to be adjusted for the correct classification such that $\{u^-_i\}_{i=1}^N \in \Xi^-_c$. In that case, $u^-_{false} \in \{u^-_i\}_{i=1}^N \in \Xi^-_c$. However, by definition, u^-_{false} is a clean sample of the c^{th} class. This requires Ξ_c to be expanded (with further learning) for an accurate prediction. However, this expansion must satisfy the competitive constraint $\sum_i^C \text{Area}(\Xi_c) = \left(2\pi^{\frac{q-1}{2}}/\Gamma(\frac{q-1}{2})\right)$. Hence, the adjustment can only admit minimal expansion of Ξ_c to enable $u^-_{false} \in \Xi_c$. This places (representation of) u^-_{false}, i.e. a clean sample of the c^{th} class, and similar samples closer to the boundary of Ξ_c.

Whereas our argument on higher sensitivity of contrastive self-supervised learning is best verified using a weaker attack like FGSM, we also investigate the adversarial susceptibility of models to stronger attacks, e.g., Projected Gradient Descent (PGD) [40]. In the following text, we first provide empirical evidence of higher sensitivity of self-supervised models in Section 4. We verify the link between the higher sensitivity of the models and false negative example pairs in Section 5. The tools developed to verify this link in Section 5 are then used to enhance self-supervised adversarial learning in Section 6.

4 Empirical Evidence of Higher Susceptibility

We primarily focus on providing empirical evidence by comparing the robustness of contrastive self-supervised image and action classification models to their supervised counterparts. Along with adversarial attacks, we also study the robustness of models w.r.t. other transformations e.g., adding noise, blurring, simulating effect of adverse weather conditions like fog etc., using ImageNet-C dataset. Similar to adversarial perturbations, these corruptions instantiate the generic transformation function $\mathcal{T}(.)$ discussed in Section 3.

Method		FGSM			PGD-ℓ_∞		PGD-ℓ_2
	$\epsilon = 0$	$\epsilon = .25/255$	$\epsilon = .5/255$	$\epsilon = 1/255$	$\epsilon = .25/255$	$\epsilon = 1/255$	$\epsilon = 0.5$
Supervised	76.71	38.20 ↓50%	22.56 ↓71%	11.53↓85%	28.22 ↓63%	0.65 ↓99%	11.3 ↓85%
SwAV	75.34	23.35 ↓69%	11.47 ↓85%	5.95 ↓92%	11.73 ↓84%	0.20 ↓100%	4.1 ↓95%
SimCLR	68.95	24.33 ↓65%	14.43 ↓79%	8.85 ↓87%	10.89 ↓84%	0.24 ↓100%	5.2 ↓92%

Table 1: Susceptibility of models under weak FGSM and PGD attacks. ImageNet top-1 accuracy is reported. Percentage drop relative to clean input accuracy is given in ↓red. Self-supervised models show higher relative drops.

4.1 Image classification

For the image classification task, we use a pre-trained ImageNet ResNet50 model trained in a supervised manner, and two other ResNet50 models trained with self-supervision techniques. To that end, we use SimCLR [8], which uses contrastive loss, and SwAV [6] methods. SwAV does not use contrastive loss, however, it also preserves the uniformity property of representations, which, according to our analysis in Section 3, is the primary cause of higher adversarial susceptibility of self-supervised learning models. Hence, for a more insightful analysis, we include SwAV in our study as well. We keep architectural similarity between different models to ensure transparent results.

Susceptibility to adversarial perturbations: A comparison of adversarial susceptibility of the models is provided in Table 1. In the table, we use FGSM by varying its perturbation scale ϵ in the range [0, 4], where 0 indicates clean images. The image dynamic range is [0, 255]. For the reported top-1 accuracy, percentage reductions for SwAV and SimCLR are much larger than the supervised model. The difference is particularly large for the weaker perturbations, which indicates the higher sensitivity of the model predictions. The results align well with the theoretical insights in Section 3. The observation also holds for the two popular variants of the stronger PGD attack in the last two columns of the table. We provide results for the standard ℓ_∞ and ℓ_2 variants of the algorithm, performing 40 iterations for the former and 10 for the latter, which is a commonly adopted setting in the literature. Table 1 points to the higher relative adversarial sensitivity of the self-supervision models.

Susceptibility to image corruptions: We also employ ImageNet-C dataset [28] to analyze the robustness of models to more primitive transformations, e.g., blurring and noise addition. ImageNet-C includes these perturbations at 5 increasing distortion levels [28]. However, the lowest level is the most relevant to our analysis because we are concerned with the higher sensitivity of the models. Detailed quantitative results of our experiments are presented in the supplementary material. We summarize those results in Fig. 2, which plots the drop in model accuracy relative to the clean image baseline against the corresponding drop for the supervised model. Best-fit lines are plotted in the figure as '`Self-Supervised Drop` = *slope* $*$ `Supervised Drop`'. Here, *slope* > 1 indicates higher sensitivity of the self-supervised model relative to the supervised model. The larger the *slope*, the higher the sensitivity to the image corruption.

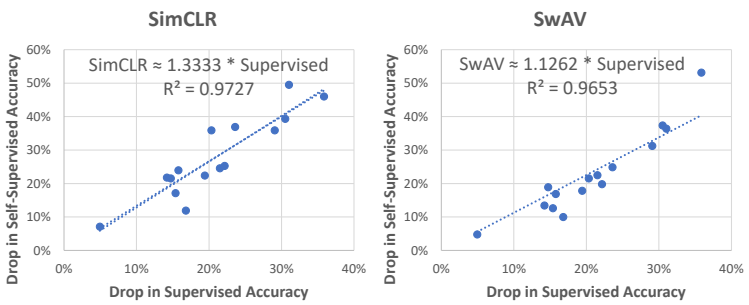


Fig. 2: Susceptibility to ImageNet-C corruptions. A point represents relative accuracy reduction with corruption (i.e. a row entry in Table 1 of supplementary material). The slope of the best fit line identifies the overall robustness relative to the supervised baseline. Larger slope indicates a less robust model. Raw data in Section C of Supplementary.

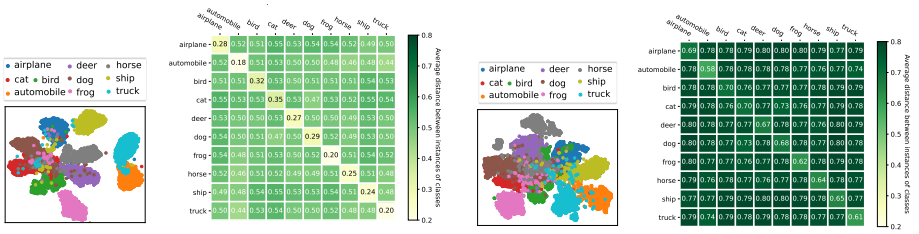
		FGSM-ℓ_∞			**PGD-ℓ_∞**	
Pre-Training	$\epsilon = 0$	$\epsilon = 1/255$	$\epsilon = 2/255$	$\epsilon = 4/255$	$\epsilon = 1/255$	$\epsilon = 2/255$
Supervised	59.4	26.6 ↓55%	12.2 ↓79%	3.9 ↓93%	15.6 ↓74%	5.2 ↓91%
TCLR [16]	75.5	10.8 ↓86%	6.1 ↓92%	3.3 ↓96%	6.30 ↓92%	3.1 ↓96%
CVRL [45]	60.2	6.00 ↓90%	3.0 ↓95%	1.6 ↓97%	4.70 ↓92%	1.9 ↓97%

Table 2: Top-1 accuracy for UCF101 video classification under FGSM and 4-step PGD attacks. Perturbation scaling ϵ varies from 0 to 2 for 8-bit videos. Percentage drop relative to clean data accuracy is given in ↓red.

The results in Fig. 2 establish higher overall sensitivity of the self-supervised models for 15 image corruption types. Interestingly, SimCLR also showed sensitivity to corruptions like Brightness and Contrast jittering, which are used as augmentations in SimCLR training. SwAV is relatively more robust to non-adversarial transformations, which is a natural consequence of its ability to 'cluster' positive samples for a class.

4.2 Video classification

To establish that our observations also hold for different types of models, we perform analysis for action recognition as an example of video classification task. Recently, action recognition techniques have started to exploit contrastive learning [16], [45]. This opens up the avenue of adversarial robustness analysis for the problem. We mainly discuss FGSM-based analysis here due to its higher relevance to the core insight. Results related to PGD are also provided in the supplementary material, which are in-line with the FGSM results. We employ an 18-layer R-(2+1)-D model in our experiments. Its one variant is trained with supervised cross-entropy loss, and other two are trained using contrastive self-supervised learning methods, TCLR [16] and CVRL [45]. These pre-trained models are obtained through communication with the authors of TCLR [16]. We summarize our results in Table 2 which shows that the video classification with self-supervised models also gets affected more strongly by the attack as compared to the supervised models. This is true despite self-supervised models outperforming the supervised model on clean inputs by a considerable margin.



(a) Supervised **(b)** Self-Supervised

Fig. 3: tSNE Visualization of representation space and average inter- and intra- class distances for CIFAR-10 instance pairs obtained with **(a)** Supervised and **(b)** Self-Supervised model trained with contrastive loss. Average ratio of inter-class distances relative to intra-class distances is much lower for the Self-Supervised model (1.19×) than for Supervised (1.98×), which leads to lower adversarial susceptibility.

Pre-Training	**FGSM-ℓ_∞**			**PGD-ℓ_2**		**PGD-ℓ_1**
	$\epsilon = 0$	$\epsilon = 8/255$	$\epsilon = 16/255$	$\epsilon = 0.1$	$\epsilon = 0.25$	$\epsilon = 8.0$
CIFAR-10						
Supervised (Cross entropy)	95.4	26.0 ↓73%	20.1 ↓79%	31.2↓67%	15.7↓84%	16.5↓83%
Supervised (Contrastive)	95.5	38.8 ↓59%	31.8 ↓67%	34.2↓64%	18.4↓81%	20.7↓78%
Self-Supervised (Contrastive)	92.7	26.8 ↓71%	13.4 ↓86%	20.9↓77%	8.3↓91%	11.5↓88%
CIFAR-100						
Supervised (Cross entropy)	74.9	14.3 ↓81%	8.4 ↓89%	23.1 ↓69%	11.5↓85%	12.1↓84%
Supervised (Contrastive)	76.3	12.6 ↓83%	6.7 ↓91%	21.9 ↓71%	9.2 ↓88%	13.4↓82%
Self-Supervised (Contrastive)	68.9	9.40 ↓87%	3.0 ↓96%	13.7 ↓80%	4.4 ↓94%	6.8↓90%

Table 3: Supervised and self-supervised CIFAR models are trained with similar training setups (see the supplementary material for details) and their robustness is compared for FGSM (ℓ_∞) and PGD attack variants (ℓ_1 & ℓ_2). Results averaged over 5 training runs.

4.3 Supervised contrastive learning

In the experiments so far, we established higher sensitivity of pre-trained self-supervised models. These models may be trained with slightly different codebases that use different augmentations. Hence, here we further analyse the scenario of in-house model training where a comparison is conducted with contrastive loss-based supervised counterparts induced under the same codebase. We perform controlled experiments with supervised contrastive learning [35] using CIFAR-10 and CIFAR-100 datasets. Our experiments use the same data augmentation strategy for all models, with a minor difference of using weaker color jittering for the supervised cross-entropy model, as suggested by [9]. Results of our experiments are summarized in Table 3, which suggest that supervised models with contrastive loss are still more resilient to adversarial manipulation as compared to their self-supervised counterparts. In Fig. 3, we provide tSNE visualisation of the representations for supervised and self-supervised CIFAR-10 models learned under the contrastive loss in Table 3. Clearly, the supervised model is able to separate the features better than the self-supervised model. This observation is inline with our analytical analysis that shows uniform representations in self-supervised contrastive learning renders the model more sensitive to input

perturbations. We also provide analysis of inter- and intra-class margins of the two types of models in the supplementary material to further elaborate on this.

5 False Negative Removal Reduces Susceptibility

We have thoroughly established that self-supervised contrastive learning is more sensitive to adversarial inputs than supervised learning. Our analytical analysis in Section 3 points to the presence of false negative instances in the training data as the major cause of this higher sensitivity. Thus, detecting and removing those can potentially improve the model robustness. However, identifying those instances is not straightforward in the absence of label information. Our further analytical treatment of the problem reveals that self-supervised contrastive learning process itself can be helpful to address the issue. According to *Lemma 5.1*, self-supervised contrastive learning forces the model to gradually improve for the true positive samples despite the presence of false negatives in the training data. This observation can be leveraged to identify and remove the suspected false negatives during model training.

Lemma 5.1: *Under contrastive loss of Eq. (1), encoder representation* $f(.)$ *improves for* $(u, v) \sim p_{pos}$ *to converge despite encountering 'm'* $u^-_{false} \in \{u^-_i\}_{i=1}^N$, *where* $m<N/e$.

Proof: Given a reasonable model state for which $f(u) \not\perp f(v)$, $f(u)^\intercal f(v) > 0\ \forall (u, v) \sim p_{\text{pos}}$. Ignoring the temperature parameter, we get $e^{f(u)^\intercal f(v)} \in (1, e]$, because false positives are not possible under self-supervised constrastive learning strategy. In this case, the lower bound on $\mathcal{L}$ for convergence in Eq. (1) is given by $\mathcal{L}_{\text{LB}} = -\log(\frac{1}{1+N/e})$. This is achievable when $f(u) \parallel f(v)$ and $f(u^-_i) \perp f(v), \forall i$. Assuming '$m$' false negatives in data for a practical state where $f(u) \not\parallel f(v)$, the model must assert the following to achieve convergence: $\frac{e}{e+N} = \frac{\rho}{(m+1)\rho+N-m}$, where $\rho = e^{f(u)^\intercal f(v)}$. Simplifying, we get $f(u)^\intercal f(v) = \ln \frac{e(N-m)}{N-em} = \delta_{\text{assert}}$. For $\mathcal{L}_{\text{LB}}$, $\delta_{\text{target}} = \ln e = 1$, resulting in $\delta_{\text{assert}} > \delta_{\text{target}}$ under the validity condition $em < N$. Given 'u' and 'v' are fixed for δ_{assert}, reducing the loss value for convergence is only possible by improving $f(.)$ for p_{pos} even in the presence of $m \in [1, N/e)$ false negatives $u^-_{false} \in \{u^-_i\}_{i=1}^N$. **QED.**

As per *Lemma 5.1*, a converging training process (wherein $\mathcal{L}$ gradually reduces) identifies an improvement of model representation for the true positives despite a few false negatives in the mini-batch. By definition, false negatives must correlate strongly to the true positives. Hence, we can decide on a suspect false negative by measuring the cosine distance between a sample's representation to that of the anchor in our mini-batch. In every epoch, we remove 'k' potential false negatives with the largest distances, where we adjust k dynamically following the intuition from *Lemma 5.1*.

Our dynamic false negative removal methodology, which is inspired by the learning objective itself, must account for two concerns. Namely, if we are too aggressive in instance removal, we may also accidentally remove true negatives. On the other hand, if we are not aggressive enough, we may miss removing actual false negatives. Since it is not possible to know the rate of instance removal *a priori*, we devise two strategies which approach the problem from the

	Attacks →	FGSM-ℓ_∞		AutoAttack-ℓ_∞	
Pre-Training	$\epsilon = 0$	$\epsilon = 8/255$	$\epsilon = 16/255$	$\epsilon = 1/255$	$\epsilon = 2/255$
Supervised Contrastive	95.5	38.8 ↓59%	31.8 ↓67%	24.3 ↓74%	11.5 ↓88%
Self-Supervised Contrastive	92.7	26.8 ↓71%	13.4 ↓86%	13.7 ↓85%	4.3 ↓95%
SimCLR + Static False Negative Removal					
Frequency Prior	93.1	30.1 ↓68% ↑25%	21.3 ↓77% ↑53%	17.4 ↓81% ↑25%	7.5 ↓92% ↑43%
Clustering based Psuedo-Labeling	92.8	28.9 ↓69% ↑18%	18.7 ↓80% ↑32%	18.1 ↓80% ↑18%	7.8 ↓91% ↑57%
SimCLR + Dynamic False Negative Removal					
Precision strategy	93.5	33.6 ↓64% ↑58%	24.9 ↓73% ↑68%	19.3 ↓79% ↑57%	8.7 ↓90% ↑72%
Recall strategy	91.5	31.3 ↓66% ↑42%	21.6 ↓76% ↑53%	19.1 ↓79% ↑55%	8.6 ↓90% ↑70%

Table 4: Robustness improvement with false negative removal. Top-1 accuracy under FGSM attack and `AutoAttack` for CIFAR-10 models. The first two rows provide results without false negative removal. Drop in accuracy under attack is reported in ↓red, percentage of gap closed w.r.t. supervised contrastive learning is indicated in **↑green**.

opposite sides. Based on principles they follow we term them 'precision' and 'recall' strategies.

Precision strategy (dynamic) is conservative, and only starts removing potential false negatives once the quality of model improves considerably. The k nearest instances in feature space are removed as potential false negatives while computing the contrastive loss. After every 'N' epochs, we sets $k = k + 1$, where $k = 0$ at initialization.

Recall strategy (dynamic) starts off by removing half of the samples closest to the anchor and slowly decreases the size of the removed set by Batch-size/R after every 'N' epochs. This strategy prioritizes removing false negatives even at the cost of inadvertently removing true negatives. Hence, we termed it recall strategy.

Both precision and recall are *dynamic* false negative removal strategies. Apart from these, we also analyze two simpler *static* strategies, which provide a baseline for our results in Table 4. These methods are described below.

Frequency prior (static): We remove a fixed fraction of negative instances from the mini-batch. These instances are sorted based on the cosine distance of their representation with the anchor. So, the removed samples are likely to include false negatives. While this improves the results over the SimCLR baseline - see Table 4, here $\frac{B}{C}$ samples per mini-batch are removed, where B is the batch size and C is the number of classes in the dataset. Its key weakness is that it requires *a priori* knowledge of class sample frequencies for optimal performance. This makes the method less pragmatic.

Clustering-based (static): This is a two-stage process where we first learn a model using SimCLR. Then, in the second stage, we perform supervised contrastive learning using pseudo-labels obtained by clustering the training data using features from the first stage model. The cluster labels are used to ensure that we do not encounter false negatives in the second stage.

In the above, 'frequency prior' performs a constant thresholding, disregarding the suspect false negative proportion or model quality. The 'pseudo labelling' method is computationally expensive as it requires an additional round of training. Hence, both these methods are less desirable than our dynamic strategy. Nevertheless, they provide informative experimental baselines as naive techniques. We set hyperparameters using cross-validation. For the precision strategy, we set

$N = 100$. For the recall strategy, $N = 75$ and $R = 16$ give the best results. For the static psuedo-labelling, we set $K = 20$ clusters.

Results for the best set of hyperparameters are averaged over 5 training runs for each strategy and summarized in Table 4. The first two rows are provided for reference, as they do not employ false negative removal. From the table, the dynamic methods are more successful, with precision strategy achieving a considerable gain in the robustness, bringing the self-supervised model performance closer to the supervised model performance. The robustness improvement achieved here is without any adversarial training and minimal additional overhead.

6 Enhancing Adversarial Contrastive Learning

Adversarial learning is a widely adopted paradigm for robustifying models against adversarial attacks in the supervised learning domain [3]. We demonstrate that our technique in Section 5 can readily augment adversarial learning in the CSL domain. To that end, we enhance the popular Robust Contrastive Learning (RoCL) [37] and Adversarial Contrastive Learning (ACL) [33] methods with our technique. Here, it is also pertinent to mention that referring to [37], [5], [7], Hendrycks et al. [29] alluded to the idea that self-supervision can help in adversarial robustness. This proposition has never been tested though. Our findings provide evidence against this idea. This makes our contribution towards the enhancement of adversarial contrastive learning even more relevant. Below, we provide details of enhancing the RoCL method with our technique. Discussion on the ACL enhancement is provided in the supplementary material.

In general, adversarial learning solves the following (non-convex) min-max optimization problem:

$$\underset{\theta}{\operatorname{argmin}} \underset{(v,\ell)\sim p_{\text{org}}}{\mathbb{E}} \Big[\max_{\rho \in B(v,\epsilon)} \mathcal{L}(\theta, v+\rho, \ell) \Big], \tag{4}$$

where ℓ is the label used to generate ρ within the ℓ_∞-ball $B(v, \epsilon)$ of radius ϵ. The requirement of apriori knowledge of ℓ in Eq. (4) makes this formulation inapplicable to the self-supervised learning paradigm. Hence, [37] adopts a different formulation. In our settings, we can re-write the RoCL problem as

$$\underset{\theta}{\operatorname{argmin}} \underset{v\sim p_{\text{org}}}{\mathbb{E}} \Big[\max_{\rho \in B(u,\epsilon)} \underset{\text{con},\theta}{\mathcal{L}} \big(u+\rho, \{\tilde{u}\}, \{u_i^-\}_{i=1}^N\big) \Big]. \tag{5}$$

The notation in the above equation is described as part of the discussion below.

In Eq. (5), $\underset{\text{con},\theta}{\mathcal{L}}$ is a more generalized form of the contrastive loss presented in Eq. (1). We refer to [37] for the exact analytical expression. Here, it is relevant to understand that the inner maximisation objective in Eq. (5) sees a sample as 'adversarial' if it increases the contrastive loss. This removes the need of sample labels in computing the adversarial examples (in contrast to Eq. 4). However, the considered contrastive loss must be defined over a set of N negative samples, along the perturbed positive sample $(u + \rho)$ and a set of other positive samples ($\{\tilde{u}\}$) that are formed by transforming the anchor. Not only that, the outer minimization problem must again be solved with the help of contrastive self-supervised learning, which relies on negative instances, along other factors.

Pre-Training		PGD-ℓ_∞		PGD-ℓ_2	PGD-ℓ_1	`AutoAttack`-ℓ_∞	
	$\epsilon = 0$	$\epsilon = \frac{8}{255}$	$\epsilon = \frac{16}{255}$	$\epsilon = 0.25$	$\epsilon = 12$	$\epsilon = \frac{8}{255}$	$\epsilon = \frac{16}{255}$
Supervised	95.5	0.0	0.0	24.8	25.4	0.0	0.0
Self-Supervised	92.7	0.0	0.0	17.1	21.1	0.0	0.0
RoCL [37]	86.0	43.6	11.4	70.9	80.0	40.8	11.2
Ours *(A-RoCL-static)*	87.5	44.8 ↑**2.8%**	12.9 ↑**13.2%**	72.3 ↑**3.4%**	80.9 ↑**1.1%**	42.3 ↑**3.7%**	11.9 ↑**6.3%**
Ours *(A-RoCL-dynamic)*	87.9	45.9 ↑**5.3%**	13.2 ↑**15.8%**	72.8 ↑**2.7%**	82.1 ↑**2.6%**	43.1 ↑**5.6%**	12.1 ↑**8.0%**
ACL [37]	86.2	41.2	12.1	72.3	80.7	39.8	10.2
Ours *(A-ACL-static)*	87.5	42.1 ↑**2.2%**	13.1 ↑**8.3%**	75.3 ↑**4.1%**	83.4 ↑**3.3%**	41.0 ↑**2.9%**	10.7 ↑**4.7%**
Ours *(A-ACL-dynamic)*	87.9	42.5 ↑**3.1%**	13.2 ↑**9.5%**	75.9 ↑**5.0%**	83.5 ↑**3.5%**	41.3 ↑**3.7%**	10.8 ↑**5.5%**

Table 5: Top-1 accuracy of adversarially trained CIFAR-10 models under PGD attack and `AutoAttack`. Attack strength ϵ is expressed in terms of ℓ_∞, ℓ_2 and ℓ_1 norms. The first two rows provide results without adversarial training. Robust models are trained with PGD ℓ_∞ adversary. Percentage performance gain of our false negative removal augmented methods over adversarially trained RoCL (median gain across attacks **5.5%**) and ACL (median gain **4.4%**) is in ↑**green** .

It does not require sophisticated analytical analysis to conclude that a corrupt set of negative instances $\{u_i^-\}_{i=1}^N$ in Eq. (5) can have pronounced adverse effects on RoCL due to the multi-fold dependence of the optimisation problem on those instances. To mitigate that, we alter the optimisation problem of RoCL to:

$$\underset{\theta}{\operatorname{argmin}} \underset{v\sim p_{\text{org}}}{\mathbb{E}} \Big[\max_{\rho\in B(u,\epsilon)} \mathcal{L}_{\text{con},\theta}\big(u{+}\rho, \{\tilde{u}\}, \{\varphi(u_i^-)\}_{i=1}^N\big)\Big], \tag{6}$$

where $\varphi(.)$ is a false negative replacement function. It removes the suspect false negatives using a strategy discussed in the previous section, and replaces them with other negative samples from the training data. We solve Eq. (6) with a variant of RoCL algorithm [37] that we devise and refer to as Augmented-RoCL (A-RoCL). The key difference between RoCL and A-RoCL is the additional false negative sample replacement step that we introduce to incorporate our findings. We provide complete details of A-RoCL training method in Section D of the supplementary material, where we also discuss the Augmented-ACL (A-ACL).

To evaluate the performance gain achieved by A-RoCL and A-ACL, we mainly followed [37] and performed adversarial training with PGD ℓ_∞-norm bounded adversary. The model robustness is evaluated for ℓ_∞, ℓ_1, ℓ_2 PGD. Additionally, we also evaluate the performance on ℓ_∞ AutoAttack for the CIFAR-10 dataset. Our results are averaged over 5 training runs and are summarized in Table 5. A-RoCL-static and A-ACL-static use 'frequency prior' for negative instance removal, whereas A-RoCL-dynamic and A-ACL-dynamic uses the 'precision' strategy. These are the best performing strategies from their respective categories. As can be seen, A-RoCL and A-ACL consistently improve performance gain over RoCL [37] and AC [33].

7 Conclusion

We presented the first systematic evidence of higher sensitivity of contrastive self-supervised learning models to adversarial attacks. We analytically established the presence of false negative pairs during CSL training as the major contributor to

the adversarial susceptibility of these models due to the property of 'uniformity'. Our analysis is supported by extensive empirical evidence, which we provided for image and video classification tasks. We also devised simple yet effective strategies to intrinsically improve the adversarial robustness of contrastive self-supervised learning. Finally, we showed that these strategies can also help in improving state-of-the-art adversarial contrastive learning approaches.

A Overview

This supplementary material is organized into 8 sections. Firstly, Section B provides necessary implementation details and attribution for existing assets used in our experiments. Section C provides the raw numberical results for experiments with natural corruptions of ImageNet-C data. These results are the same as plotted in Figure 2 of the main paper. Section D.1 and Section D.2 provide the modified algorithm for ARoCL and AACL training respectively. Sections E and F provide additional experimental results which were excluded from the main paper. This includes detailed results for adversarial robustness of video classification models under additional attacks such as PGD and AutoAttack and robustness of our dynamic false negative removal technique under PGD attack (FGSM and AutoAttack results were included in the main paper). Section G provides an empirical comparison of the inter- and intra- class margin between self-supervised and supervised contrastive models. Section H details our hypothesis exploring the similarities between SwAV and Contrastive Pre-Training in terms of feature space uniformity.

B Implementation Details

B.1 Resources Used

All experiments are performed using an internal slurm cluster with 4x Quadro RTX 6000 GPUs (24GB VRAM each) and 16 CPU cores along with 64 GigaBytes of memory. The resource constraints primarily apply to the self-supervised training experiments and other experiments can be done with fewer resources. PyTorch 1.9 was used for all experiments.

B.2 ImageNet Experiments

The pretrained model weights are chosen from the following repositories:
SimCLR: https://github.com/google-research/simclr
SwAV: https://github.com/facebookresearch/swav

FGSM and PGD attacks are used to test the adversarial robustness of ImageNet Models. FGSM attack results are verified through two different implementations (`Foolbox` library and authors' implementation). Attack magnitudes are specified in the respective tables.

ImageNet-C experiments only utilize a subset of the dataset for which the distortion strength was 1 (on a scale of 1-5). The dataset can be obtained from the authors at: https://github.com/hendrycks/robustness

B.3 CIFAR Experiments

This section provides the implementation details necessary for reproducing the results in Table 4 of the main paper. These are controlled experiments on CIFAR-10 and CIFAR-100 datasets performed for the purpose of isolating the effect of Self-Supervised and Supervised training on adversarial robustness.

For implementing our contrastive learning experiments, we build upon the code from the authors of SCL [36] (https://github.com/HobbitLong/SupContrast), and implement the adversarial attacks and our false negative removal strategies. The models are trained with a batch size of 1024. The contrastive models are trained for 1000 epochs, whereas the cross-entropy based model is trained for 500 epochs. Cosine annealing learning rate scheduler with a peak learning rate of 1.0 and warmup is used in both cases. For each type of model, 20 different training runs are carried out and the results are averaged across them in order to reduce the effect of random run-to-run variance.

Data Splits: We use the standard Test (10,000)/Train (40,000) split for the CIFAR datasets as provided by Torchvision datasets submodule.

Image Augmentations: We design our augmentation pipeline using transformation operations from the `torchvision.transforms` library. Both spatial (Random Cropping and Flipping) and Colorimetric transforms (Color jittering and dropping) are used. As suggested by SimCLR [8] we utilize stronger augmentations for training the contrastive models.

The specific augmentation pipeline used in each case are as follows:
Contrastive Models:

```
RandomResizedCrop(size=32,scale=(.2,1.)),
RandomHorizontalFlip(),
RandomApply([ColorJitter(0.4,0.4,0.4,0.1)],p=0.8),
RandomGrayscale(p=0.2),
ToTensor(),
Normalize(mean=mean,std=std)
```

Cross-Entropy Models:

```
RandomResizedCrop(size=32,scale=(.2,1.)),
RandomHorizontalFlip(),
RandomApply([ColorJitter(0.1,0.1,0.1,0.05)],p=0.8),
RandomGrayscale(p=0.2),
ToTensor(),
Normalize(mean=mean,std=std)
```

B.4 Datasets and other assets

CIFAR Datasets by Alex Krizhevsky et al. [38], ImageNet by Jia Deng et al. [17] and ImageNet-C by Dan Hendrycks et al. [28] were originally collected from across the internet, and the copyright for the individual images rests with the

original owner. They are used only for research purposes and are not redistributed by us. They can be obtained from the original authors at:

CIFAR: https://www.cs.toronto.edu/ kriz/cifar.html

ImageNet: https://www.image-net.org/download.php

Pre-Trained models for UCF-101 Action Recognition were obtained from the authors of TCLR [16] can be requested directly from them for reproducing the results.

C Robustness to Natural Image Corruptions

In Section 5.1 of the main paper, we utilize the Imagenet-C [28] dataset to evaluate the robustness of self-supervised and supervised ImageNet classifiers on common natural corruptions and perturbations. ImageNet-C contains full sized images for all 1,000 ImageNet classes, with natural corruptions such as noise, digital transformations, blurring and simulated rough weather. For each corruption, there are 5 different strength levels. Since the models being evaluated are not adversarially trained, we only use corruptions of strength 1. The self-supervised models (with contrastive loss) are significantly less robust over a broad range of distortion types, including noise, digital effects, blurring and simulated weather distortions. Table 6 in this document reports the accuracies and performance drop for the supervised, SimCLR [8] and SwAV [6] models used in the main paper. Each row entry in this table corresponds to a point in Figure 2 of the main paper (pg. 5).

D Augmented Robust Self-Supervised Contrastive Learning Algorithms

In Section 7 of the main paper, we propose two different Augmented Robust Self-Supervised Contrastive Learning Algorithms based on our finding about the effect of False Negative Pairs during training. As compared to the original algorithms [37], [33] that we build on, the proposed algorithms expect an additional input, i.e. the false negative replacement function $\varphi(.)$. As noted in the main paper, we can implement this functions as 'static' or 'dynamic' strategies (Section 6 of the paper). The major difference between RoCL and ARoCL (and ACL and AACL) is the application of $\varphi(.)$ to the negative examples. Since we have established false negatives as the main source of excess adversarial susceptibility, simply improving the set of negative examples improves the performance of robust contrastive learning.

D.1 Augmented-RoCL

Augmented Robust Contrastive Learning (ARoCL) algorithm, which we develop as an enhancement of the RoCL algorithm [37]. ARoCL is presented in Algorithm 1 below. In the presented algorithm, we follow the notational conventions from [37] for the ease of understanding.

Type	Supervised	SimCLR	SwAV
-	76.7	68.9	75.3
Noise			
Impulse	49.2 ↓36%	37.2 ↓46%	35.3 ↓53%
Shot	59.7 ↓22%	51.5 ↓25%	60.4 ↓20%
Gaussian	61.8 ↓19%	53.5 ↓22%	61.9 ↓18%
Digital			
Brightness	72.9 ↓5%	64.0 ↓7%	71.7 ↓5%
Contrast	63.8 ↓17%	60.7 ↓12%	67.8 ↓10%
Pixelate	64.9 ↓15%	57.1 ↓17%	65.8 ↓13%
Elastic	65.8 ↓14%	53.9 ↓22%	65.2 ↓13%
JPEG	65.4 ↓15%	54.1 ↓21%	61.1 ↓19%
Blur			
Defocus	58.6 ↓24%	43.5 ↓37%	56.6 ↓25%
Motion	64.6 ↓16%	52.4 ↓24%	62.6 ↓17%
Glass	54.4 ↓29%	44.2 ↓36%	51.8 ↓31%
Zoom	52.9 ↓31%	34.8 ↓49%	47.9 ↓36%
Weather			
Fog	61.1 ↓20%	44.2 ↓36%	59.1 ↓22%
Frost	60.2 ↓22%	52.0 ↓25%	58.4 ↓22%
Snow	53.3 ↓31%	41.8 ↓39%	47.2 ↓37%

Table 6: Robustness to simulated corruptions on ImageNet-C (top-1 accuracy). Percentage drop relative to clean accuracy is in red. The table corresponds to Figure 2 in the main paper.

D.2 Augmented-ACL

Augmented Adversarial Contrastive Learning (ARoCL) algorithm uses a two stream architecture in which we maintain separate batch-norm layers for the adversarial and clean data. This technique is an enhancement of the ACL algorithm [33]. AACL is detailed in Algorithm 2 below. We closely follow the notational conventions from [33] for the ease of understanding, but some changes have been made for consistency.

E Additional Attacks on Video Classification

In Section 5.2 of the main paper, we have shown that supervised action recognition models are more robust than self-supervised models under the FGSM attack [24]. In this section, we demonstrate that this is also true for the case of stronger PGD attack [40]. Results for 4-step ℓ_∞ PGD and `AutoAttack` attacks are presented in Table 7 of this document. The table corresponds to Table 3 in the main paper (that uses FGSM attack). As can be seen, for small values of ϵ, the reduction

Algorithm 1 Augmented Robust Contrastive Learning (ARoCL)

Require: Dataset $\mathbb{D}$, model parameters θ, model f, parameter of projector π, projector g, constant λ, replacement function $\varphi(.)$

Learn: model parameters θ, parameter of projector π ▷ Replacement function $\varphi(.)$ removes *False Negatives*

1: **for all** iter $\in$ number of training iteration **do**

2: **for all** $x \in$ minibatch $\boldsymbol{x} = \{x_1, \ldots, x_m\}$ **do**

3: Generate adversarial examples from transformed inputs ▷ *RoCL uses instance-wise* attacks

$$t(x)^{i+1} = \Pi_{B(t(x),\epsilon)}(t(x)^i + \alpha \texttt{sign}(\nabla_{t(x)^i} \mathcal{L}_{\texttt{con},\theta,\pi}(t(x)^i, \{t'(x)\}, \varphi(t(x)_{\texttt{neg}}))))$$

4: **end for**

5: Compute total loss:

$$\mathcal{L}_{\texttt{total}} = \frac{1}{N} \sum_{k=1}^{N} [\mathcal{L}_{\texttt{RoCL},\theta,\pi} + \lambda \mathcal{L}_{\texttt{con},\theta,\pi}(t(x)_k^{adv}, \{t'(x)_k\}, \{\varphi(t(x)_{\texttt{neg}}\}))]$$

6: Optimize weights θ, π over $\mathcal{L}_{total}$

7: **end for**

		PGD		AutoAttack
Pre-Training	$\epsilon = \mathbf{0}$	$\epsilon = \mathbf{1/255}$	$\epsilon = \mathbf{2/255}$	$\epsilon = \mathbf{1/255}$
Supervised	59.4	15.6 ↓74%	5.2 ↓91%	12.6 ↓79%
TCLR [16]	75.5	6.30 ↓92%	3.1 ↓96%	4.00 ↓95%
CVRL [45]	60.2	4.70 ↓92%	1.9 ↓97%	3.70 ↓94%

Table 7: Top-1 accuracy for UCF101 video classification under 4-step PGD attack and *AutoAttack*. Perturbation scaling ϵ varies from 0 to 2 for 8-bit videos. Percentage drop relative to clean data accuracy is given in ↓red.

in the accuracy of supervised method is much less than that of self-supervised models. Note that we are interested in relative susceptibility of the models. Since PGD is a very strong attack that can easily fool even supervised models, we operate in the lower range of ϵ values to demonstrate the relative susceptibility of the models.

F Robustness of Dynamic False negative removal under PGD Attack

In the main paper, we demonstrate that false negative removal is, to an extent, successful at mitigating the lack of robustness in self-supervised contrastive models. Those results were based on FGSM attack and `AutoAttack`. Here, we demonstrate that this observation is also true in the case of the stronger PGD attack. Results for 4-step ℓ_∞ PGD attack are presented in Table 8. Note that, here we are only concerned with the dynamic strategies for False Negative removal.

Algorithm 2 Augmented Adversarial Contrastive Learning (AACL)

Require: Dataset $\mathbb{D}$; Transforms $\mathcal{T}$; Network backbone f, projection head g; replacement function $\varphi(.)$;

Learn: Standard BN parameters $\boldsymbol{\theta}_{bn}$; Adversarial branch BN parameters $\boldsymbol{\theta}_{bn^{adv}}$; All parameters $\boldsymbol{\theta}$ in f and g;

1: **for all** iter $\in$ number of training iteration **do**
2: **for all** minibatch $\boldsymbol{x} = \{x_1, \ldots, x_m\}$ **do**
3: Sample augmentations from $\mathcal{T}$ to form $(\tilde{\boldsymbol{x}}_i, \tilde{\boldsymbol{x}}_j)$ from $\boldsymbol{x}$. ▷ Replacement function $\varphi(.)$ removes *False Negatives*
4: Generate the corresponding adversarial mini-batch $(\tilde{\boldsymbol{x}}_i + \boldsymbol{\delta}_i, \tilde{\boldsymbol{x}}_j + \boldsymbol{\delta}_j)$ with

$$\boldsymbol{\delta}_i, \boldsymbol{\delta}_j = \underset{\|\delta_i\|_\infty \leq \epsilon, \|\delta_j\|_\infty \leq \epsilon}{\operatorname{argmax}} \ell_{NT}(f \circ g(\varphi(\tilde{\boldsymbol{x}}_i) + \boldsymbol{\delta}_i, \varphi(\tilde{\boldsymbol{x}}_j) + \boldsymbol{\delta}_j; \boldsymbol{\theta}, \boldsymbol{\theta}_{bn^{adv}}))$$

5: Compute total losses with adversarial and clean examples:

$$\mathcal{L}_{\texttt{clean}} = \mathcal{L}_{NT-XENT}(f \circ g(\varphi(\tilde{\boldsymbol{x}}_i), \varphi(\tilde{\boldsymbol{x}}_j); \boldsymbol{\theta}, \boldsymbol{\theta}_{bn}))$$

$$\mathcal{L}_{\texttt{adversarial}} = \mathcal{L}_{NT-XENT}(f \circ g(\varphi(\tilde{\boldsymbol{x}}_i) + \boldsymbol{\delta}_i, \varphi(\tilde{\boldsymbol{x}}_j) + \boldsymbol{\delta}_j; \boldsymbol{\theta}, \boldsymbol{\theta}_{bn^{adv}}))$$

$$\mathcal{L}_{\texttt{total}} = \mathcal{L}_{\texttt{clean}} + \alpha \mathcal{L}_{\texttt{adversarial}}$$

6: Update parameters $(\boldsymbol{\theta}_{bn}, \boldsymbol{\theta}_{bn^{adv}}, \boldsymbol{\theta})$ to minimize $\mathcal{L}_{\texttt{total}}$.
7: **end for**
8: **end for**

G Discussion of Inter- and intra-class margins

As discussed in Section 5.3 of the main paper, here we provide numerical evidence that inter- and intra-class difference between supervised and self-supervised contrastive models explains the difference in the robustness of these models. We provide heatmaps for SimCLR (self-supervised) and supervised contrastive learning model in Figure 3 of the main paper. The heatmaps show average distances between instances of each class-pair. We provide the heatmaps on the same color scale. The diagonal terms correspond to the intra-class margins, whereas the off-diagonal terms are the inter-class margins. For SimCLR, the inter- and intra-class margins are very similar (avg ratio of inter upon intra class margin is 1.19), whereas for the Supervised Contrastive model, the inter-class margins are much higher than the intra-class margins (avg ratio of inter upon intra class margin is 1.98). As a result the supervised models are much more adversarially robust.

H Discussion of SwAV and feature space uniformity

Even though `SwAV` is not a pure "contrastive" learning method in that it does not utilize a contrastive loss, nevertheless due to its design it possess the uniformity property which makes it vulnerable to adversarial attacks in a similar

Pre-Training	$\epsilon = \mathbf{0}$	$\epsilon = \mathbf{1/255}$	$\epsilon = \mathbf{4/255}$
Supervised Contrastive	95.5	29.5 ↓69%	16.3 ↓83%
Self-Supervised Contrastive (SimCLR)	92.7	20.1 ↓78%	9.60 ↓90%
SimCLR + Dynamic False Negative Removal			
precision strategy	93.5	26.1 ↓72%	11.9 ↓87%
recall strategy	91.5	24.9 ↓73%	11.0 ↓76%

Table 8: Further results for robustness improvement with dynamic false negative removal strategies proposed in the main paper. Top-1 accuracy under 4-step PGD attack (attack strength ϵ expressed in terms of ℓ_∞ norm) is reported for CIFAR-10 models. The first two rows provide baseline results without false negative removal. `FGSM` and `AutoAttack` Results are in the main paper.

way as contrastive attacks. Both `SwAV` and SimCLR rely on generating positive pairs through data augmentation. While SimCLR loss directly operates on the features/representation generated by the CNN, `SwAV` utilizes an online clustering algorithm to generate "codes" using learned prototypes. SimCLR enforces the alignment property by forcing the features for positive pairs to be similar, whereas `SwAV` relies on a "swapped" prediction problem, i.e. predicting the "codes" obtained from one augmented view using the other view. On the other hand, while SimCLR enforces uniformity by simply treating each instance as a negative pair for the anchor in the contrastive loss, `SwAV` requires a different strategy.

The two key components of `SwAV` enforcing uniformity in the feature space are the *equipartition constraint* and *entropy regularization*. Since `SwAV` operates on "codes" assigned using an online clustering algorithm, the *equipartition constraint* enforces that on average each prototype is selected at least $\frac{\texttt{Batch Size}}{\texttt{Number of prototypes}}$ times in each minibatch. This means that in a given minibatch the instances are assigned uniformly across the prototypes on average.

Mathematically the `SwAV` code assignment step is represented as the following optimization:

$$\max_{\mathbf{Q}\in\mathcal{Q}} \operatorname{Tr}\left(\mathbf{Q}^\top \mathbf{C}^\top \mathbf{Z}\right) + \varepsilon H(\mathbf{Q}), \tag{7}$$

Here we have a mini-batch of B feature vectors $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_B]$, and the optimization is mapping them to prototypes $\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_K]$. The mapping is represented by $\mathbf{Q} = [\mathbf{q}_1, \ldots, \mathbf{q}_B]$, where $\mathbf{Q}$ is optimized to maximize the similarity between the features and the prototypes. Here H is the entropy regularization function, $H(\mathbf{Q}) = -\sum_{ij} \mathbf{Q}_{ij} \log \mathbf{Q}_{ij}$, which can be adjusted through ε which controls the smoothness of the mapping. A higher ε means instances are more uniformly assigned to different prototypes.

Mathematically, the *equipartition constraint* can be represented as:

$$\mathcal{Q} = \left\{ \mathbf{Q} \in \mathbb{R}_+^{K\times B} \mid \mathbf{Q}\mathbf{1}_B = \frac{1}{K}\mathbf{1}_K, \mathbf{Q}^\top \mathbf{1}_K = \frac{1}{B}\mathbf{1}_B \right\}, \tag{8}$$

Here $\mathbf{1}_K$ denotes the K-dimensional vector of ones. This constraint enforces that on average each prototype is selected at least $\frac{B}{K}$ times in the batch.

Working in tandem, the *equipartition constraint* and *entropy regularization* ensure that instances are well distributed across the representation hypersphere, which prevents tight clustering of classes which is possible with supervised learning. In the limit where number of prototypes used in `SwAV` is equal to number of instances in the dataset, `SwAV` will be equivalent to `SimCLR`.

References

1. Ahsan, U., Madhok, R., Essa, I.: Video jigsaw: Unsupervised learning of spatiotemporal context for video action recognition. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 179–189. IEEE (2019) 3
2. Akhtar, N., Mian, A.: Threat of adversarial attacks on deep learning in computer vision: A survey. IEEE Access **6**, 14410–14430 (2018) 6
3. Akhtar, N., Mian, A., Kardan, N., Shah, M.: Advances in adversarial attacks and defenses in computer vision: A survey (2021) 13
4. Bachman, P., Hjelm, R.D., Buchwalter, W.: Learning representations by maximizing mutual information across views. In: Wallach, H., Larochelle, H., Beygelzimer, A., d Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 32. Curran Associates, Inc. (2019), https://proceedings.neurips.cc/paper/2019/file/ddf354219aac374f1d40b7e760ee5bb7-Paper.pdf 3
5. Carmon, Y., Raghunathan, A., Schmidt, L., Liang, P., Duchi, J.C.: Unlabeled data improves adversarial robustness. arXiv preprint arXiv:1905.13736 (2019) 4, 13
6. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. In: Advances in Neural Information Processing Systems. vol. 33 (2020), https://proceedings.neurips.cc/paper/2020/file/70feb62b69f16e0238f741fab228fec2-Paper.pdf 3, 8, 17
7. Chen, T., Liu, S., Chang, S., Cheng, Y., Amini, L., Wang, Z.: Adversarial robustness: From self-supervised pre-training to fine-tuning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 699–708 (2020) 4, 13
8. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PMLR (2020) 1, 3, 4, 8, 16, 17
9. Chen, T., Kornblith, S., Swersky, K., Norouzi, M., Hinton, G.E.: Big self-supervised models are strong semi-supervised learners. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 22243–22255. Curran Associates, Inc. (2020), https://proceedings.neurips.cc/paper/2020/file/fcbc95ccdd551da181207c0c1400c655-Paper.pdf 3, 10
10. Chen, T., Li, L.: Intriguing properties of contrastive losses. arXiv preprint arXiv:2011.02803 (2020) 4
11. Chen, X., Fan, H., Girshick, R., He, K.: Improved baselines with momentum contrastive learning (2020) 3
12. Chen, X., He, K.: Exploring simple siamese representation learning (2020) 4
13. Chen, X., Xie, S., He, K.: An empirical study of training self-supervised vision transformers (2021) 3

14. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016) 1
15. Croce, F., Hein, M.: Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In: International conference on machine learning. pp. 2206–2216. PMLR (2020) 3
16. Dave, I., Gupta, R., Rizve, M.N., Shah, M.: Tclr: Temporal contrastive learning for video representation. arXiv preprint arXiv:2101.07974 (2021) 3, 9, 17, 19
17. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. IEEE (2009) 16
18. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: Proceedings of the IEEE international conference on computer vision. pp. 1422–1430 (2015) 3
19. Donahue, J., Krähenbühl, P., Darrell, T.: Adversarial feature learning. arXiv preprint arXiv:1605.09782 (2016) 3
20. Dumoulin, V., Belghazi, I., Poole, B., Mastropietro, O., Lamb, A., Arjovsky, M., Courville, A.: Adversarially learned inference. arXiv preprint arXiv:1606.00704 (2016) 3
21. Feng, Z., Xu, C., Tao, D.: Self-supervised representation learning by rotation feature decoupling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10364–10374 (2019) 3
22. Geirhos, R., Narayanappa, K., Mitzkus, B., Bethge, M., Wichmann, F.A., Brendel, W.: On the surprising similarities between supervised and self-supervised models. NeurIPS workshop on Shared Visual Representations in Human and Machine Intelligence (2020) 2, 4
23. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations. In: International Conference on Learning Representations (2018), https://openreview.net/forum?id=S1v4N2l0- 3
24. Goodfellow, I., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: International Conference on Learning Representations (2015), http://arxiv.org/abs/1412.6572 6, 18
25. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., Piot, B., Kavukcuoglu, K., Munos, R., Valko, M.: Bootstrap your own latent - a new approach to self-supervised learning. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 21271–21284. Curran Associates, Inc. (2020), https://proceedings.neurips.cc/paper/2020/file/f3ada80d5c4ee70142b17b8192b2958e-Paper.pdf 4
26. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9729–9738 (2020) 3
27. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016) 1
28. Hendrycks, D., Dietterich, T.: Benchmarking neural network robustness to common corruptions and perturbations. In: International Conference on Learning Representations (2019), https://openreview.net/forum?id=HJz6tiCqYm 8, 16, 17

29. Hendrycks, D., Mazeika, M., Kadavath, S., Song, D.: Using self-supervised learning can improve model robustness and uncertainty. In: Wallach, H., Larochelle, H., Beygelzimer, A., d Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 32. Curran Associates, Inc. (2019), https://proceedings.neurips.cc/paper/2019/file/a2b15837edac15df90721968986f7f8e-Paper.pdf 4, 13
30. Ho, C.H., Nvasconcelos, N.: Contrastive learning with adversarial examples. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 17081–17093. Curran Associates, Inc. (2020), https://proceedings.neurips.cc/paper/2020/file/c68c9c8258ea7d85472dd6fd0015f047-Paper.pdf 4
31. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017) 1
32. Jaynes, E.T.: Information theory and statistical mechanics. Physical review **106**(4), 620 (1957) 2
33. Jiang, Z., Chen, T., Chen, T., Wang, Z.: Robust pre-training by adversarial contrastive learning. Advances in Neural Information Processing Systems **33**, 16199–16210 (2020) 3, 13, 14, 17, 18
34. Jiang, Z., Chen, T., Chen, T., Wang, Z.: Robust pre-training by adversarial contrastive learning. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 16199–16210. Curran Associates, Inc. (2020), https://proceedings.neurips.cc/paper/2020/file/ba7e36c43aff315c00ec2b8625e3b719-Paper.pdf 4
35. Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D.: Supervised contrastive learning. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 18661–18673. Curran Associates, Inc. (2020), https://proceedings.neurips.cc/paper/2020/file/d89a66c7c80a29b1bdbab0f2a1a94af8-Paper.pdf 2, 10
36. Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D.: Supervised contrastive learning. In: Advances in Neural Information Processing Systems. vol. 33 (2020), https://proceedings.neurips.cc/paper/2020/file/d89a66c7c80a29b1bdbab0f2a1a94af8-Paper.pdf 16
37. Kim, M., Tack, J., Hwang, S.J.: Adversarial self-supervised contrastive learning. In: Thirty-fourth Conference on Neural Information Processing Systems, NeurIPS 2020. NeurIPS (2020) 3, 4, 13, 14, 17
38. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems **25**, 1097–1105 (2012) 16
39. Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., Shi, W.: Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017) 1
40. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: International Conference on Learning Representations (2018) 3, 7, 18
41. Moosavi-Dezfooli, S.M., Fawzi, A., Frossard, P.: Deepfool: a simple and accurate method to fool deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2574–2582 (2016) 7

42. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018) 3
43. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2536–2544 (2016) 3
44. Purushwalkam, S., Gupta, A.: Demystifying contrastive self-supervised learning: Invariances, augmentations and dataset biases. In: Advances in Neural Information Processing Systems. vol. 33 (2020), https://proceedings.neurips.cc/paper/2020/file/22f791da07b0d8a2504c2537c560001c-Paper.pdf 4
45. Qian, R., Meng, T., Gong, B., Yang, M.H., Wang, H., Belongie, S., Cui, Y.: Spatiotemporal contrastive video representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6964–6974 (2021) 3, 9, 19
46. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015) 3
47. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016) 1
48. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A.: Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th international conference on Machine learning. pp. 1096–1103 (2008) 3
49. Wang, F., Liu, H.: Understanding the behaviour of contrastive loss. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2495–2504 (June 2021) 4
50. Wang, T., Isola, P.: Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In: III, H.D., Singh, A. (eds.) Proceedings of the 37th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 119, pp. 9929–9939. PMLR (13–18 Jul 2020), http://proceedings.mlr.press/v119/wang20k.html 2, 4, 5
51. Xiao, T., Wang, X., Efros, A.A., Darrell, T.: What should not be contrastive in contrastive learning. In: International Conference on Learning Representations (2021), https://openreview.net/forum?id=CZ8Y3NzuVzO 4
52. Xie, C., Tan, M., Gong, B., Wang, J., Yuille, A.L., Le, Q.V.: Adversarial examples improve image recognition. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020) 4
53. Xu, D., Xiao, J., Zhao, Z., Shao, J., Xie, D., Zhuang, Y.: Self-supervised spatiotemporal learning via video clip order prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 10334–10343 (2019) 3