NO SHIFTED AUGMENTATIONS (NSA): STRONG BASELINES FOR SELF-SUPERVISED ANOMALY DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Unsupervised Anomaly detection (AD) requires building a notion of normalcy, distinguishing in-distribution (ID) and out-of-distribution (OOD) data, using only available ID samples. Recently, large gains were made on this task for the domain of natural images using self-supervised contrastive feature learning as a first step followed by kNN or traditional one-class classifiers for feature scoring. Learned representations that are non-uniformly distributed on the unit hypersphere have been shown to be beneficial for this task. We go a step further and investigate how the *geometrical compactness* of the ID feature distribution makes isolating and detecting outliers easier, especially in the realistic situation when ID training data is polluted (i.e. ID data contains some OOD data that is used for learning the feature extractor parameters). We propose novel architectural modifications to the self-supervised feature learning step, that enable such compact ID distributions to be learned. We show that the proposed modifications can be effectively applied to most existing self-supervised learning objectives with large gains in performance. Furthermore, this improved OOD performance is obtained without resorting to tricks such as using strongly augmented ID images (e.g. by 90 degree rotations) as proxies for the unseen OOD data, which imposes overly prescriptive assumptions about ID data and its invariances. We perform extensive studies on benchmark datasets for one-class OOD detection and show state-of-the-art performance in the presence of pollution in the ID data, and comparable performance otherwise. We also propose and extensively evaluate a novel feature scoring technique based on the angular Mahalanobis distance, and propose a simple and novel technique for feature ensembling during evaluation that enables a big boost in performance at nearly zero run-time cost compared to the standard use of model ensembling or test time augmentations. Code for all models and experiments will be made open-source.

1 INTRODUCTION

Anomaly detection (AD), novelty detection, or out-of-distribution (OOD) detection requires using only available in distribution (ID) samples for training a classifier to decide upon the relative normalcy of samples at test time, without knowledge of the nature of the OOD data. For natural images, which according to the manifold hypothesis lie in a compact set in a suitable space, this means finding as tight a decision boundary as possible around the normal set, whilst excluding unseen samples from other classes or distributions. This is an important problem with practical applications in, for example, Industrial defect detection, fraud detection, autonomous driving, biometrics, spoofing detection and many other domains (Tack et al., 2020). It relates to issues with generalization of learned Deep Neural Network features, over-confidence of supervised classifiers, and practical cases where the nature of the anomalous images may not be known a-priori, or may change over time.

OOD detection in the visual domain was traditionally done with either generative (Zhai et al., 2016) or discriminative models (Schölkopf et al., 1999) on top of shallow features. As with many other vision applications, progressing from shallow to deep representations provided a large boost in performance. Using the density of deep generative models learned on images however often proved to be ineffective, e.g. (Kingma & Dhariwal, 2018), with poorly calibrated likelhoods away from observed data. There have instead been two recent directions to learn suitable deep features used for ood evaluation: a) supervised pre-training on an external dataset. b) Self-Supervised Learning (SSL)

pre-training on either the normal set only or also on an external dataset. A variety of leaned metrics, scoring functions, or one-class classifiers have then been employed on top of these learned features; this general paradigm has shown to be highly effective in many cases (Sohn et al., 2020).

The best recent results with SSL-based training for OOD has been with contrastive learning (Tack et al., 2020; Sohn et al., 2020; Sehwag et al., 2021; Winkens et al., 2020). Contrastive learning has been shown to distribute ID data uniformly on the hypersphere (Wang & Isola, 2020). While this helps general multi-class SSL training, it hurts OOD detection, as it makes isolating outliers from the single class harder (Sohn et al., 2020). This uniformity also makes OOD detection much more sensitive to pollution in the inlier training data (Han et al., 2021b; Sohn et al., 2020). If some labeled OOD as negatives are available, semi-supervised learning maybe be used (Ruff et al., 2018; Han et al., 2021b).

If no such labeled negatives are available, one way to soften the effect of a contrastively learned uniform representation is to introduce artificial negative samples as proxies for these outliers. Using hard augmentations (such as 90 degree rotations) has been termed *Distributional Shifting* (Tack et al., 2020; Sohn et al., 2020; Li et al., 2021; Tian et al., 2021). This has the intention of making the in-distribution data less uniform, and thus easier to isolate from OOD data. However, this comes at the significant assumption that the data is not (fully or partially) invariant to the used hard augmentation, and that it is a good proxy for the true negative distribution.

Another direction is using features from a model pre-trained on a large external dataset, with the hope of producing universal features that can work in any OOD detection scenario. This can be done in either supervised (Reiss et al., 2020) or self-supervised manner (Xiao et al., 2021). However, the assumption that such representative labeled samples will be available in the latter case, or that learned image features from general datasets such as ImageNet will transfer well in all cases may be restrictive at best.

In this work, we first investigate the training dynamics of contrastive SSL methods, and show that their performance decays significantly on long term training, we found that uniformity or non-compactness of the learned ID distribution is the main reason behind this. We study this effect on positive pairs only SSL using SimSiam (Chen & He, 2020), and show it doesn't suffer from it. We also propose a simple architectural modification and perform extensive analysis to show it always enourages learning a more compact ID distribution.

Finally, compared with methods like CSI that include many bells and whistles in their formulation, we evaluate fair baselines, excluding for example test-time-augmentations / ensembling. We summarize our contributions as follows:

- We show that SimCLR loses most of it is OOD performance if trained longer than usual. We empirically verify and quantify that uniformity of learned ID distribution is the main reason for this, we also show SimSiam to be resilient to this phenomena.
- We propose a simple and novel solution for this behaviour and show it works very well with both SimSiam and SimCLR and produces solid performance improvements.
- We extensively evaluate proposed models under different datasets and setups and show they have comparable performance against more complex methods, more importantly they show state-of-the-art performance with big margin under the presence of polluted data.
- We investigate and solve a gradient problem in SimSiam (and also BYOL) that greatly affect the proper propagation of the norms inside the network, we solve it and notice much higher stability, especially in the low batch size training regime.
- We show unexpected case(s) (e.g. SVHN) where the usual ImageNet-Pretrained ResNet methods fail catastrophically on One-Class Classification Anomaly Detection tasks, even with feature adaptation.
- We achieve robustness across a number of scenarios (pollution, small batch size), and evaluate on a wide variety of datasets, separating the contributions of representation learning, data augmentation, and additional variations like ensembling.

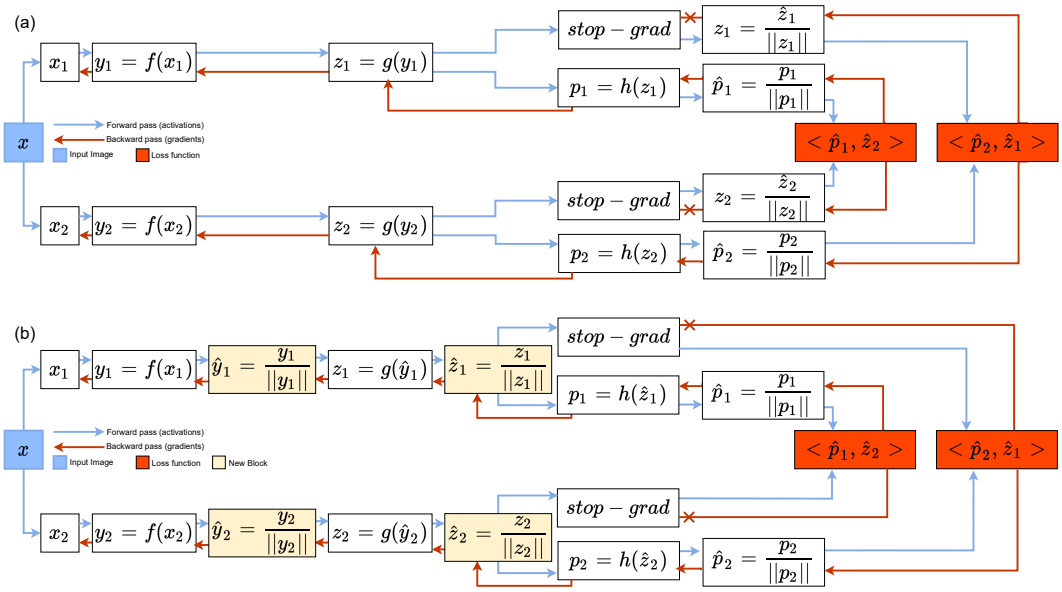

Figure 1: An illustration of the proposed modifications to SimSiam and its gradient flow problem. (a) Original SimSiam. Although all operations contributing to the loss function have gradients flowing back and contributing to learn-able parameters of the whole network, the gradients of $\hat{z}_i = z_i/||z_i||_2$ get blocked in *all possible paths* by the stop-grad operation. Compare this to normalization of the predictor $\hat{p}_i = p_i/||p_i||_2$ whose gradients flow smoothly for both images. (b) The proposed modification, in both Sections 2.3 and 2.4.

2 Methodology

2.1 SSL setup

A general SSL setup resembling SimSiam is depicted in Figure 1. SimCLR is very similar architecturally except that it doesn't have a prediction network, so we illustrate our proposed modifications using this diagram. It shows both the forward pass and backward pass of two random augmentations x_1 and x_2 of an input image x. Both are encoded by the shared encoder f into y_1 and y_2. f is the convolutional backbone network, then both are projected in z_1 and z_2 by a projection network g. A prediction network h transforms z_i into $p_i = h(z_i)$ and the whole network learns to match the output of the prediction network fed with the encoding of one view $p_i = h(z_i)$ to the encoding of the other view $z_j = f(x_j)$ and vice versa, by minimizing their negative cosine similarity:

$$\mathcal{D}(p_i, z_j) = -\frac{p_i}{\|p_i\|_2} \cdot \frac{z_j}{\|z_j\|_2}, \tag{1}$$

Most current self-supervised learning algorithms use a projection head g on top of a convolutional feature extractor, as this was empirically found to help learn a much better final representation (Chen et al., 2020a;b; Grill et al., 2020; Chen & He, 2020). However, it was also found that the learned projection is much worse in downstream tasks (Chen et al., 2020a) this was also shown specifically for OOD (Sohn et al., 2020). So, the output of the encoder backbone f is used during feature evaluation in most OOD methods based on SSL, as it learns a much better representation. We call the outputs of f the learned embedding.

2.2 Is representation quality the only important thing for OOD ?

To illustrate that the quality of learned representation is only one factor for OOD performance, and others exist, we train SimCLR on one class of CIFAR10 and evaluate the learned representation for OOD detection against all other classes; a similar setup is used in (Sehwag et al., 2021; Tack et al., 2020). However, we train for much longer and analyze the learned representation, repeating for 4 different classes. Figure 3 evaluates the learned representations throughout training using 5 different

metrics. It is evident from Figure 3a, 3b, 3c that OOD performance reaches a peak very fast (within few hundred epochs) then deteriorates significantly as training progresses (Due to space limitations, detailed figures are in the appendix A).

One easy way to explain this behavior is to attribute it to the deteriorating quality of learned representation; we carry out two more evaluations to examine such hypothesis. In Figure 3d, 3e we use linear evaluation (Zhang et al., 2016) and weighed k-NN (Wu et al., 2018) to evaluate the quality of learned representation. Figure 3d shows that the quality of the representation barely changes during training, i.e. there is only the expected small decrease after the peak and then just fluctuations, and the linear separability holds. Figure 3e shows in-between performance for k-NN evaluation, while there is a significant decrease for $k = 1$, the situation is better for $k = 20$.

The previous results tell two stories. (a) Methods based on density estimation show performance drops down significantly as training progress. (b) Doing supervised classification on the frozen features show the features are still of high quality and maintain nearly the same linear separability performance of the learned representation. This points to a change on the underlying distribution of the embedding space.

To study what is happening, we use the tool of the von Mises-Fisher (vMF) distribution which is a fundamental probability distribution on the $(n-1)$-dimensional hyper-sphere $S^{d-1} \subset R^d$. Its probability density function is given by $f_n(z, \mu, \kappa) = C_n(\kappa)e^{\kappa\mu^T z}$, where μ and κ are denoted as the mean direction and concentration parameter respectively. The shape of the vMF depends on the concentration parameter κ. For high values of κ the distribution has a mode at the mean direction μ, and is uniform on the hyper-sphere S^{d-1} for $\kappa = 0$. vMF has been successfully used in both analyzing (Wang & Isola, 2020; Lee et al., 2018a; 2021) and learning (Hasnat et al., 2017; Davidson et al., 2018; Kumar & Tsvetkov, 2018) deep neural networks.

We fit a vMF (Banerjee et al., 2005; Sra, 2012) to the learned normalized embeddings of SimCLR, and study how κ (the concentration parameter) changes during training. Figure 4a shows how SimCLR starts with a relatively large κ (high concentration) and it then goes monotonically down (low concentration, more uniform) as training progresses – this is true for both ID and OOD. In Figure 4b we notice the same behaviour using a different tool: the MMD (Gretton et al., 2012) between the learned representation and samples from a uniform distribution on a unit hypersphere, as suggested in (Sohn et al., 2020), where a high MMD means less uniform and more concentrated, and a low MMD means a more uniform and less concentrated distribution.

We can see that this gives a very good explanation of the decaying performance we noticed in Figure 3. It is not the quality of the representation that is essential, but rather its distribution. As the ID distribution gets more uniform the probability p to find an inlier sample $x \in$ ID arbitrary close to an outlier query sample $x' \in$ OOD increases, i.e. ID and OOD get more and more indistinguishable. This is exactly the intuition behind many methods for AD, for example (Ruff et al., 2018; Goyal et al., 2020; Ruff et al., 2020)

Given this analysis, and the fact that contrastive methods are proven to converge to a uniform distribution (Wang & Isola, 2020), we take the natural step and carry the same analysis on a non-contrastive SSL method, SimSiam (which we preferred over BYOL as it is simpler and retains most of the performance). Results are shown in Figure 5, where indeed there is big difference to SimCLR: OOD performance stays nearly constant after reaching the peak. Also in Figure 6 we see that although the representation is also getting more uniform, it is at a much slower rate, and is able to maintain high values for κ and MMD even after many epochs of training.

It is also important to mention that this analysis shows that the number of training epochs (or use of early stopping criteria) is a very important hyper-parameter to choose for contrastive methods used for OOD. This needs to be carefully tuned, and we see that non-contrastive methods have a big advantage in being less sensitive here. Other works that noticed importance of early stopping in this context include (Reiss et al., 2020; Han et al., 2021a; Reiss & Hoshen, 2021)

2.3 Learning a dense representation

(Schmidt et al., 2018) show that the sample complexity of robust learning can be significantly larger than that of standard learning. While some works tried to address this with extra positive or negative data, (Pang et al., 2019) propose the interesting idea of manipulating the local sample distribution of

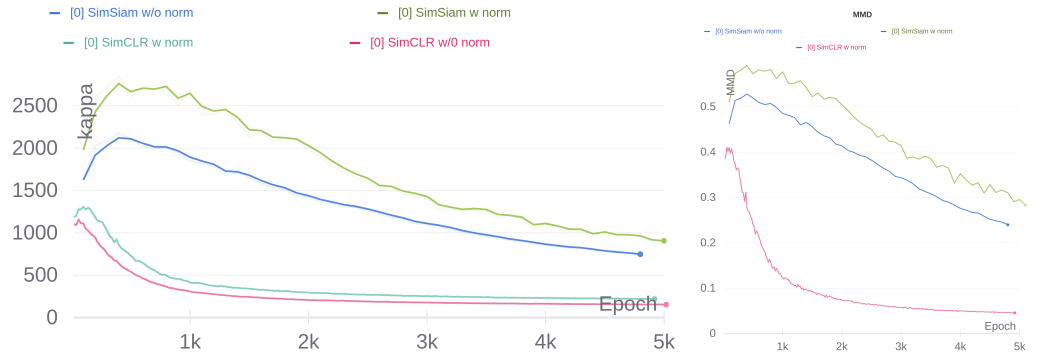

Figure 2: Same setup of Figure 4 but for class 0 only. We compare four cases, SimCLR with and without normalization, and SimSiam with and without normalization.

the training data via appropriate training objectives such that by inducing high-density feature regions, there would be locally sufficient samples to train robust classifiers and return reliable predictions. We propose pursuing the same direction with the OOD problem, and propose a simple architectural modification that can help induce high density feature space.

We propose adding a differentiable l_2-normalization operation after the encoder f and before the projection head g. As such, the output $y_i = f(x_i)$ is transformed into $\hat{y_i} = y_i/||y_i||_2$ (as in Figure 1). The intuition is that adding the normalization step would deprive the model of any gradients when the norm of the embedding is changed, thus enforcing the model to learn directional transformations as those would be the only way to actually to decrease the loss function. Regularizing the model this way should give a finer directional control on learned embedding (the cosine distance makes more sense), and yield a more efficient use of volume and thus a denser representation.

We perform a series of experiments to empirically verify our intuition. Figure 7 shows evaluation logs for SimCLR after adding normalization. Comparing this to Figure 3, it is evident the training is much more regularized now, the decrease in OOD performance after the peak is much smaller and performance is maintained for thousands of epochs. Figure 8 shows a denser learned representation when compared to Figure 4 – one can also notice that κ for ID and OOD is much more homogeneous now.

Figure 2 gives a better one-to-one comparison of all four setups (SimCLR with and without normalization and SimSiam with and without normalization). It gives clear picture of how adding normalization enables the model to learn a denser representation in either case and throughout training.

Lastly, we would like to emphasize that while having a dense ID distribution can be important for OOD detection with clean ID data, it is much more important in the presence of pollution in the ID data. In this polluted setup, some OOD data are mixed in during training and considered ID by the loss. In this case, a compact ID distribution decreases chance of other OOD data to be considered as ID as much as possible. This is empirically verified in the experimental section.

2.4 SimSiam and BYOL gradient flow problem

To avoid the degenerate solution of a collapsed representation, the authors of (Chen & He, 2020) found it crucial to have a gradient blocking operation (`stop-grad`) that blocks gradients starting from $\hat{z}_i$ from flowing back to other parts of the network. This lack of gradients acts as a regularizer and makes it hard for the optimizer to reach the trivial solution of a collapsed representation. Note that the same analysis presented here equally applies to BYOL (Grill et al., 2020) the existence of the momentum encoder enforces an implicit `stop-grad`.

However both (Chen & He, 2020; Grill et al., 2020) does not study the effect `stop-grad` may have on proper gradient flow in the network. Studying Figure 1, we can see that the l_2-norm of the output of the prediction network $\hat{p_i} = p_i/||p_i||_2$ gives proper gradients that flows back to other parts of the network. The exact opposite happens for the l_2-norm of the output of the encoder network, all the

gradients of that operation gets blocked by the `stop-grad`, though it is an integral part of the loss function.

Proposed solution The straight-forward solution of removing the `stop-grad` operation converges very fast to a collapsed representation (Chen & He, 2020). Another straight-forward solution is to minimize the norm of the output representation of the encoder f, but this limits the representation ability of the encoder and also collapses very fast. One last trial would be removing normalization all together from $\mathcal{D}$, however, this solution converges to a sub-optimal representation with an unbounded norm (Grill et al., 2020).

Our proposed solution is based on a simple observation. The missing gradient from $\hat{z}_i$ carries two pieces of information: (a) moving the output of the encoder z_i to be closer to the output of the predictor p_i (which is the information we want to hide from the optimizer). (b) encouraging the projector g to learn a l_2-norm invariant representation, this facilitates training the predictor network and thus also the encoder and is what we want to maintain.

In order to maintain (b) while discarding (a), we propose a simple modification (see Figure 1 (b)). Applying a differentiable l_2-normalization to the output projection z_i, such that the new projection is $\hat{z}_i = z_i/||z_i||_2$. This gives the network proper gradient for learning an l_2-normalized representation satisfying the loss function $\mathcal{D}$ while still having the `stop-grad` operation and avoiding related collapse.

2.5 Feature evaluation for out-of-distribution detection

Many of the recent work on OOD detection is based on scoring a given test sample using its distance to the nearest training sample (Tack et al., 2020; Sehwag et al., 2021; Lee et al., 2018b). This provides a simple evaluation baseline with strong results. Many metrics have been used for this, the most commonly used are the Mahalanobis distance (Lee et al., 2018b; Sehwag et al., 2021) and the Cosine distance (Tack et al., 2020) evaluated at the output of the network after the last convolutional block.

To take the advantages of both of these distances, we go a step further and score features with the arccosine of the Mahalinobis Cosine similarity (i.e. angular Mahalinobis distance) (Beveridge et al., 2005), which have been proposed and studied well previously in the field of face recognition (Beveridge et al., 2005; Vinay et al., 2015). Mahalinobis Cosine is simply the Cosine similarity between vectors after projection into the Mahalinobis space, where the famous Mahalinobis distance is the Euclidean distance computed between vectors after also projection into the Mahalinobis space.

If x_m, y are two random training and test samples respectively, and the projection of their features $f(x_m), f(y)$ into the Mahalinobis space is u, v using the sample covariance matrix Σ_m of the training data $\{x_m\}_{m=1}^{M}$, then the Mahalinobis Cosine distance $\mathcal{D}_{MC}$ and our scoring $\mathcal{S}_{k\text{-}Cos}$ are

$$\mathcal{D}_{MC}(x_m, y) = \frac{u}{\|u\|_2} \cdot \frac{v}{\|v\|_2}, \quad \mathcal{S}_{k\text{-}Cos}(y) = arccos(\max_m \mathcal{D}_{MC}(x_m, y)) \tag{2}$$

$\mathcal{S}_{k\text{-}Cos}$ considers only the distance to the nearest training sample, we also study another variant $S_{c\text{-}Cos}$, that is especially robust in the presence of pollution, that computes distance to u the mean vector of the training data $\{x_m\}$. $\mathcal{S}_{c\text{-}Cos}(y) = arccos(\mathcal{D}_{MC}(u, y))$.

Feature ensembling We also propose another evaluation scheme where all the intermediate feature maps of the network are scored independently, then their scores are all summed together. The idea is to get both high level and low level OOD scores using different feature maps. Note that this is different from (Lee et al., 2018b) that learn a weighted sum of all the feature scores, where the weights are learned on a validation set; our proposed method is a simple sum and doesn't need a validation set. It also attains a huge runtime saving when compared to test-time augmentation (TTA) used in (Tack et al., 2020) or model ensembling used in (Sohn et al., 2020), as it requires a single model forward pass on a single instance. It consists of three steps:

1. Computing a score for each feature map, using either $\mathcal{S}_{k\text{-}Cos}$ or $\mathcal{S}_{c\text{-}Cos}$ or both.
2. Normalizing the range of the scores to be between 0 and 1, simply using the range of training data scores.

3. Summing the normalized scores.

Due to space limitations we show extensive feature summation results in the appendix, along with comparison of different evaluation metrics.

3 EMPIRICAL EVALUATION

3.1 EXPERIMENTAL SETUP

We perform a thorough evaluation of our proposed normalization modification on both SimSiam and SimCLR (with and without negative augmentations). Since the problem discussed in Section 2.4 isn't applicable to SimCLR (it doesn't have a stop-grad) the proposed solution is only applied to SimSiam.

We evaluate in the one-vs-all OOD detection setting for CIFAR-10, CIFAR-100 super-classes (Krizhevsky et al., 2009), Fashion-MNIST (Xiao et al., 2017), and SVHN (Netzer et al., 2011). In this setting one class is treated as the normal/inlier class and all other classes of the same dataset are treated as outliers.

For all results presented we use a ResNet-18, trained with Adam (Kingma & Ba, 2014) with a learning rate of 0.0001 and a cosine learning rate decay (Loshchilov & Hutter, 2016). For SimCLR we used 2-layer MLP as a projection head with an architecure similar two (Sohn et al., 2020). For SimSiam we used a 3-layer MLP for both the projector and the predictor. SimCLR is trained for 500 epochs, and SimSiam for 4000 epochs. All models are trained on Nvidia V100 16GB GPUs and written in Pytorch (Paszke et al., 2019). All our reported results are without test-time augmentation.

3.2 RESULTS

3.2.1 ONE-CLASS CLASSIFICATION

Table B in the appendix summarizes results reported in literature plus some of our results. We can see both SimSiam and SimCLR with our proposed modification achieving strong performance across benchmarks. Even when compared to CSI (Tack et al., 2020) which has much more complex formulation and performs extensive test-time augmentation (160 augmentations) during evaluation. Also while we make no assumptions about the underlying training data, both CSI(Tack et al., 2020) and DROC (Sohn et al., 2020) make the strong assumption that the training data is not (fully or partially) invariant to the hard (negative-like) augmentation used during training, which is not always a practical assumption.

Thanks to our novel evaluation we report the best numbers in literature for pre-trained supervised ImageNet baselines, and without any adaptation like in (Reiss et al., 2020). We also highlight in the table the main drawback with such approach, the fact that it is blind to the invariances of the underlying training data as can seen in the very simple SVHN dataset. Thought most methods that train on SVHN get nearly perfect scores, pre-trained baselines fail miserably. A similar situation happens with the SteelDefects dataset, but at a less severe scale.

3.3 PERFORMANCE UNDER POLLUTION

Table 1 shows a comparison of the performance under the very relistic scenario that some percent of the training data can be polluted with outliers. It can be seen that adding the proposed normalization drastically reduces the effect of polluted data, we can also see that combined with negative augmentations (for the SimCLR case) we get even more benefits. We can see that NSA compares very favourably to state-of-the-art semi-supervised methods with access to 5% labeled outliers, coming only behind ELSA+ which employs heavy TTA during evaluation (which is parallel to our method but incurs big computational cost) and negative augmentations.

One important take-away from the previous table is that, as predicted by our analysis in Section2.3, positive only SSL (e.g. SimSiam or BYOL) is much more suited to real world OOD detection that can possibly contain a small subset of polluted data compared to contrastive SSL (e.g. SimCLR) which suffers big performance drops in this scenario.

	p=0	p=0.05	p=0.10	Δ (p=0.1 - p=0)
SimSiam(w norm)	91.54	88.08	86.21	5.33
SimSiam(w/o norm)	89.56	85.21	81.79	7.77
SimCLR(no neg, w norm)	89.27	83.1	77.49	11.78
SimCLR(no neg, w/o norm)	86.49	71.1	63.5	25.99
SimCLR (neg aug, w norm)	92.91	86.1	83.38	9.53
SimCLR (neg aug, w/o norm)	91.12	81.5	77.94	13.18
Pretrained IN18	92.63	90.9	89.3	3.33
Pretrained IN50	94.45	92.18	89.49	4.96
Pretrained IN152	95.82	94.48	91.39	4.43
DROC (Sohn et al., 2020)	89	76.5	73.0	16.0
DROC (DA) (Sohn et al., 2020)	92.5	85.0	80.5	12.0
CSI (results from (Han et al., 2021b))	94.3	88.2	84.5	9.8
Semi-Supervised (5% labeled) ELSA (Han et al.)	85.7	83.5	81.6	4.1
Semi-Supervised (5% labeled) ELSA+ (Han et al.)	95.2	93.0	91.1	4.1

Table 1: CIFAR10 pollution experiments. p is the ratio of outlier data inside the training set. the last column is the loss in performance between training with clean data and a 10% polluted data. Clearly, our proposed modifications reduces the drop in all cases. With SimSiam we beat standard ELSA which uses 5% labeled data to maintain robustness to pollution, and come close to ELSA+, which uses TTA and other tricks from CSI on top of the baseline SSL approach.

SimSiam(w norm)	SimCLR (w neg aug, norm)	CSI	STOC(Yoon et al., 2021)
88.46	90.59	84.5	89.9

Table 2: Performance of in CIFAR10 under 10% pollution under different ensebling setups.

Table 2 shows CIFAR10 performance under 10% pollution, when some form of ensembling is used. For our proposed methods, feature ensembling proposed in Section 2.5 is used. We would like to highlight the fact that we can match (and even surpass) the performance of STOC (Yoon et al., 2021) with a single model training and a single forward pass! This is significant given that STOC uses very elaborate ensembling where more than 50 different models are trained.

3.4 Effect of normalization

Table 3 Examines the effect of adding or removing the normalization for both SimSiam and SimCLR on Cifar10. Big improvements is demonstrated especially for SimCLR both with and without negative augmentations. Table 5 shows similar situation for Cifar100. The fact that normalization is more effective for SimCLR is aligned with our analysis in Section 2.4.

(Sohn et al., 2020) established the relation between batchsize and OOD performance and that lower batch size can lead to better OOD performance since it helps have a more stochastic and less uniform learned ID. In Table 4 we show that the holds for non-contrastive SSL, but *only* after our proposed normalization. We performed an analysis similar to Section 2.3 and found that without the gradient fixing normalzation proposed in Section 2.4 the learned ID distribution is very concentrated (high κ) and concentration keeps increasing during training for both ID and OOD, which (along with low

	SimSiam	SimCLR	SimCLR(w neg aug)	Pretrained IN18	PT IN50	PT IN152
w norm	91.54	89.27	92.91	n/a	n/a	n/a
w/o norm	89.56	86.49	91.12	93.63	94.45	95.82

Table 3: CIFAR10 results across variants of our method, and our simple Pretrained ImageNet Baselines. We show that our proposed normalization improve both SimSiam and SimCLR results.

	SimSiam		
batch size	32	64	512
w norm	92.9	92.29	91.54
w/o norm	77.97	80.28	89.56

Table 4: SimSiam results on CIFAR10 at different batch sizes. We demonstrate that our proposed normalization architectural changes maintain stability across batch sizes; with the default SimSiam setup, small batches reduce performance.

	SimSiam	SimCLR	SimCLR(w neg aug)	PT IN18	PT IN50	PT IN152
w norm	84.09	83.95	86.68	92.64	94.68	95.21
w/o norm	81.14	80.51	84.97	92.64	94.68	95.21

Table 5: CIFAR100

OOD AUCs) signals a collapsed distribution. Not only does the proposed gradient fix relive collapse, but provides our best performance reported with SimSiam.

3.5 Pretrained backbones on SVHN

Table 6 shows more interesting results for evaluating pre-trained features on SVHN, showing that regardless of the model size or size of pre-training dataset (using PT50 pre-trained on bilion images (Yalniz et al., 2019)), or using fully-supervised or SSL pre-training, or even with state-of-the-art feature adaptation (Reiss & Hoshen, 2021), all catastrophically fail at a very simple dataset like SVHN, just because it requires a different set of features (text has a lot more focus on structure vs texture, which is the opposite case for natural images). This shows that a meta-model for OOD is still far out of reach.

4 Conclusion

We have considered two general approaches to detection of Anomalies in images: using either SSL or Pretrained models as Deep feature extractors; followed by metric learning for scoring these outliers. These methods have both been very successful in other recent work, however we highlight cases where they are prone to fall down. In the case of Pretrained models, it is well known that they do not generalize to certain more distant domains; (Reiss et al., 2020) claimed this did not affect their pretrained baselines, however in our evaluation we demonstrate that while ImageNet Pretraining with Large ResNets is indeed a strong baseline, it does not hold up to even simple SSL baselines on more distant domain shifts.

In the case of SSL, we make arguments that much of the recent performance gains from methods such as CSI (Tack et al., 2020) or DROC (Sohn et al., 2020) are driven by two factors: strongly distributionally shifted augmentations (e.g. 90 degree rotations), as well as tricks such as Test-Time-Augmentation / ensembling, or additional classifiers leading to additional computation, over-engineered methods, and lack of generalization. In particular we see weaknesses with these methods in areas such as polluted data. Whilst semi-supervised methods like ELSA (Han et al., 2021a) can help in these cases, we demonstrate good robustness to pollution with our proposed NSA method.

NSA is built on a modified SimSiam architecture, but these modifications can apply equally to other self-supervised representation learning methods. We demonstrate that with this approach, we do not need the strongly shifted augmentations that CSI, DROC, or ELSA require as part of their headline

PT IN18	PT IN50	PT IN152	PT IN50 (SimCLR)	PT IN50 (Bilion images)	PT IN50 (Adapted)
61	66	64	65	70	68

Table 6: Performance of various pre-trained backbones on SVHN

results. We improve generalization in difficult cases, and report competitive results across the board, validating the benefits of our simple approach to robust Anomaly Detection. We note that further tricks like ensembling, clustering, additional losses could also be applied to improve our results, but wanted to provide a fair and straightforward baseline in order to compare amongst many recent methods.

REFERENCES

Severstal: Steel defect detection kaggle challenge. https://www.kaggle.com/c/severstal-steel-defect-detection/.

Arindam Banerjee, Inderjit S Dhillon, Joydeep Ghosh, Suvrit Sra, and Greg Ridgeway. Clustering on the unit hypersphere using von mises-fisher distributions. *Journal of Machine Learning Research*, 6(9), 2005.

Liron Bergman and Yedid Hoshen. Classification-based anomaly detection for general data. In *International Conference on Learning Representations*, 2020.

J Ross Beveridge, David Bolme, Bruce A Draper, and Marcio Teixeira. The csu face identification evaluation system. *Machine vision and applications*, 16(2):128–138, 2005.

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

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020b.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. *arXiv preprint arXiv:2011.10566*, 2020.

Tim R Davidson, Luca Falorsi, Nicola De Cao, Thomas Kipf, and Jakub M Tomczak. Hyperspherical variational auto-encoders. *arXiv preprint arXiv:1804.00891*, 2018.

Izhak Golan and Ran El-Yaniv. Deep Anomaly Detection Using Geometric Transformations. May 2018a.

Izhak Golan and Ran El-Yaniv. Deep anomaly detection using geometric transformations. In *Advances in Neural Information Processing Systems*, 2018b.

Sachin Goyal, Aditi Raghunathan, Moksh Jain, Harsha Vardhan Simhadri, and Prateek Jain. Drocc: Deep robust one-class classification. In *International Conference on Machine Learning*, pp. 3711–3721. PMLR, 2020.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

Sungwon Han, Hyeonho Song, Seungeon Lee, Sungwon Park, and Meeyoung Cha. Elsa: Energy-based learning for semi-supervised anomaly detection. March 2021a.

Sungwon Han, Hyeonho Song, Seungeon Lee, Sungwon Park, and Meeyoung Cha. Elsa: Energy-based learning for semi-supervised anomaly detection. *arXiv preprint arXiv:2103.15296*, 2021b.

Md Hasnat, Julien Bohné, Jonathan Milgram, Stéphane Gentric, Liming Chen, et al. von mises-fisher mixture model-based deep learning: Application to face verification. *arXiv preprint arXiv:1706.04264*, 2017.

Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. In *Advances in Neural Information Processing Systems*, pp. 15663–15674, 2019.

Chaoqin Huang, Fei Ye, Jinkun Cao, Maosen Li, Ya Zhang, and Cewu Lu. Attribute Restoration Framework for Anomaly Detection. November 2019.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, 2018.

Alex Krizhevsky et al. Learning multiple layers of features from tiny images, 2009.

Sachin Kumar and Yulia Tsvetkov. Von mises-fisher loss for training sequence to sequence models with continuous outputs. *arXiv preprint arXiv:1812.04616*, 2018.

Cheolhyoung Lee, Kyunghyun Cho, and Wanmo Kang. Directional analysis of stochastic gradient descent via von mises-fisher distributions in deep learning. *arXiv preprint arXiv:1810.00150*, 2018a.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, 2018b.

Kuang-Huei Lee, Anurag Arnab, Sergio Guadarrama, John Canny, and Ian Fischer. Compressive visual representations. *arXiv preprint arXiv:2109.12909*, 2021.

Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. Cutpaste: Self-supervised learning for anomaly detection and localization. *arXiv preprint arXiv:2104.04015*, 2021.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *Advances in Neural Information Processing Systems Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

Tianyu Pang, Kun Xu, Yinpeng Dong, Chao Du, Ning Chen, and Jun Zhu. Rethinking softmax cross-entropy loss for adversarial robustness. *arXiv preprint arXiv:1905.10626*, 2019.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.

Tal Reiss and Yedid Hoshen. Mean-shifted contrastive loss for anomaly detection. *arXiv preprint arXiv:2106.03844*, 2021.

Tal Reiss, Niv Cohen, Liron Bergman, and Yedid Hoshen. PANDA: Adapting Pretrained Features for Anomaly Detection and Segmentation. October 2020.

Douglas A Reynolds. Gaussian mixture models. *Encyclopedia of biometrics*, 741:659–663, 2009.

Oliver Rippel, Patrick Mertens, and Dorit Merhof. Modeling the Distribution of Normal Data in Pre-Trained Deep Features for Anomaly Detection. May 2020.

Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *International conference on machine learning*, pp. 4393–4402. PMLR, 2018.

Lukas Ruff, Robert A Vandermeulen, Billy Joe Franks, Klaus-Robert Müller, and Marius Kloft. Rethinking assumptions in deep anomaly detection. *arXiv preprint arXiv:2006.00339*, 2020.

Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. *arXiv preprint arXiv:1804.11285*, 2018.

Bernhard Schölkopf, Robert C Williamson, Alexander J Smola, John Shawe-Taylor, John C Platt, et al. Support vector method for novelty detection. In *NIPS*, volume 12, pp. 582–588. Citeseer, 1999.

Vikash Sehwag, Mung Chiang, and Prateek Mittal. SSD: A Unified Framework for Self-Supervised Outlier Detection. March 2021.

Kihyuk Sohn, Chun-Liang Li, Jinsung Yoon, Minho Jin, and Tomas Pfister. Learning and evaluating representations for deep one-class classification. *arXiv preprint arXiv:2011.02578*, 2020.

Suvrit Sra. A short note on parameter approximation for von mises-fisher distributions: and a fast implementation of i s (x). *Computational Statistics*, 27(1):177–190, 2012.

Jihoon Tack, Sangwoo Mo, Jongheon Jeong, and Jinwoo Shin. Csi: Novelty detection via contrastive learning on distributionally shifted instances. *arXiv preprint arXiv:2007.08176*, 2020.

Yu Tian, Guansong Pang, Fengbei Liu, Seon Ho Shin, Johan W Verjans, Rajvinder Singh, Gustavo Carneiro, et al. Constrained contrastive distribution learning for unsupervised anomaly detection and localisation in medical images. *arXiv preprint arXiv:2103.03423*, 2021.

A Vinay, Vinay S Shekhar, KN Balasubramanya Murthy, and S Natarajan. Performance study of lda and kfa for gabor based face recognition system. *Procedia Computer Science*, 57:960–969, 2015.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pp. 9929–9939. PMLR, 2020.

Jim Winkens, Rudy Bunel, Abhijit Guha Roy, Robert Stanforth, Vivek Natarajan, Joseph R Ledsam, Patricia MacWilliams, Pushmeet Kohli, Alan Karthikesalingam, and Simon Kohl. Contrastive training for improved out-of-distribution detection. *arXiv preprint arXiv:2007.05566*, 2020.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Zhisheng Xiao, Qing Yan, and Yali Amit. Do We Really Need to Learn Representations from In-domain Data for Outlier Detection? May 2021.

I Zeki Yalniz, Hervé Jégou, Kan Chen, Manohar Paluri, and Dhruv Mahajan. Billion-scale semi-supervised learning for image classification. *arXiv preprint arXiv:1905.00546*, 2019.

Jinsung Yoon, Kihyuk Sohn, Chun-Liang Li, Sercan O Arik, Chen-Yu Lee, and Tomas Pfister. Self-trained one-class classification for unsupervised anomaly detection. *arXiv preprint arXiv:2106.06115*, 2021.

Shuangfei Zhai, Yu Cheng, Weining Lu, and Zhongfei Zhang. Deep structured energy based models for anomaly detection. In *International Conference on Machine Learning*, 2016.

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

A PLOTS FOR EVALUATIONS DURING TRAINING

Here we show plots illustrating the characteristics of the learned representation over the course of very long training sessions. They appear in Figures 3, 4, 5, 8

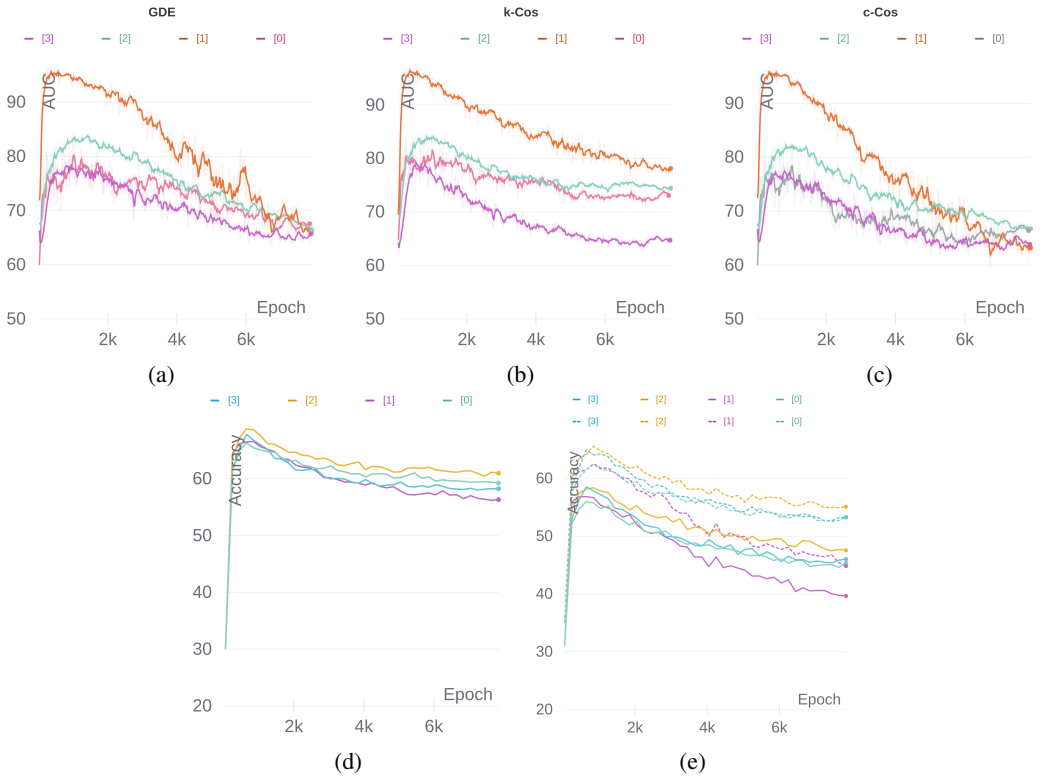

Figure 3: Training SimCLR on classes 0,1,2,3 from CIFAR10 long enough and evaluating the representation during training. (a) Using Gaussian density estimation (GDE) (Reynolds, 2009). (b) k-Cos cosine distance to the closest (k=1) point in ID. (c) c-Cos cosine distance to the mean direction in ID. Both these are defined in more detail in Section 2.5. (d) Linear evaluation (Zhang et al., 2016) of learned embedding. (d) Weighted nearest neighbor classifier (k-NN) (Wu et al., 2018). Solid lines represent $k = 1$ and dashed lines represent $k = 20$

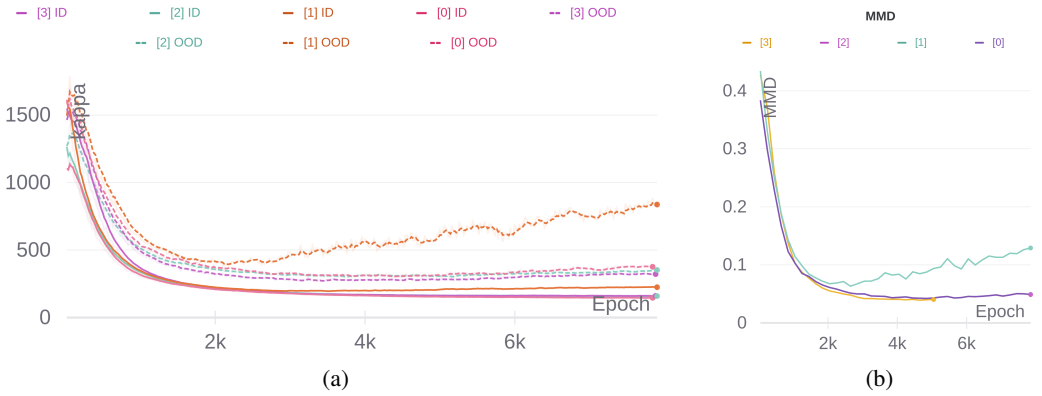

Figure 4: Training SimCLR on classes 0,1,2,3 from CIFAR10 and monitoring the learned vMF ID and OOD distributions during training. (a) κ progress during training, solid lines are ID data and dashed are OOD data. (b) The MMD (Gretton et al., 2012) between the learned representation and samples from uniform distribution on a unit hypersphere

B PREVIOUS RESULTS FROM LITERATURE

Table B gives detailed overview of OOD results from literature.

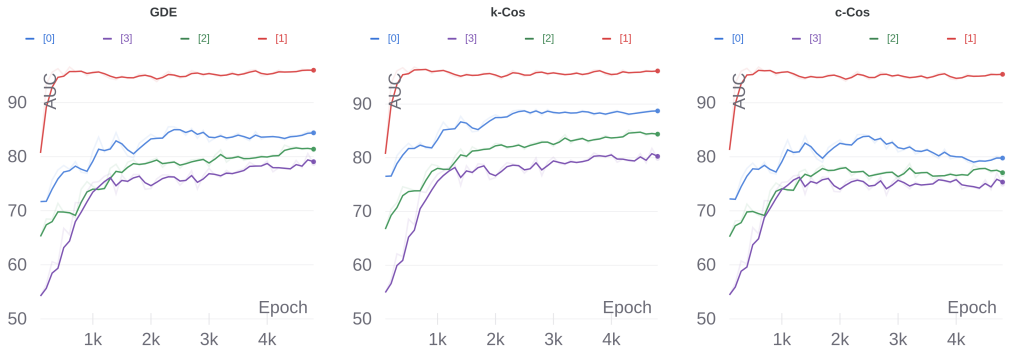

Figure 5: Same experiment as Figure 3 but with SimSiam.

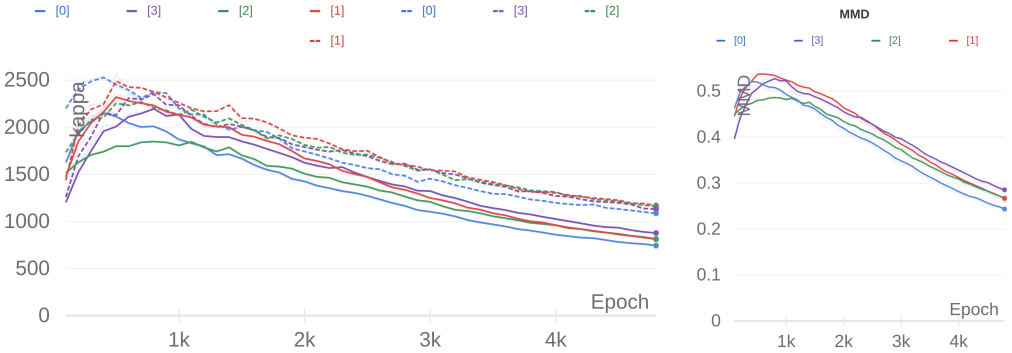

Figure 6: Same experiment as Figure 4 but with SimSiam.

C DATASET DETAILS

The main datasets we test on are: CIFAR 10, CIFAR 100 and ImageNet30, fashion-MNIST following the splits and protocols specified in (Tack et al., 2020) and (Sohn et al., 2020); we add SVHN using the same basic protocol of one class as inlier v.s. the remaining as outliers, and Steel Defects ste using normal v.s. defective samples. We resized images to 32 x 32 for all datasets apart from ImageNet30 and Steel Defects, which use the standard ImageNet ResNet-18 architecture's transformation of a 224-pixel center-cropped region from the 256 x 256 input image.

The Steel Defects dataset ste we used was noted in the appendix of CSI (Tack et al., 2020) as a rotation invariant dataset of images that "commonly appear in real-world scenarios". We note that this is a prime case where we see no real advantage of using 90 degree rotations as shifted augmentations in our baseline results on SimCLR.

The same holds true with CSI, in fact the result was much worse using their full method, giving an AUROC figure of 36.0% for CSI (Rotation), versus 74.0% for their vanilla SimCLR result, roughly in line with our result of 76.35%.[1]

D COMPARISON OF SCORING WITH DIFFERENT FEATURE EVALUATIONS & METRICS

As discussed in section 3.3, we compare the default scoring used in our main experiments (the encoder's last layer features using $\mathcal{S}_{NSA}$) with an evaluation scheme consisting of these scores summed across feature maps from the encoder backbone network, and projection heads from either SimSiam or SimCLR. In the following tables we use following feature maps:

[1]These results were unfortunately only published in the appendix version 1 of their paper, but later removed in v2 on arxiv.

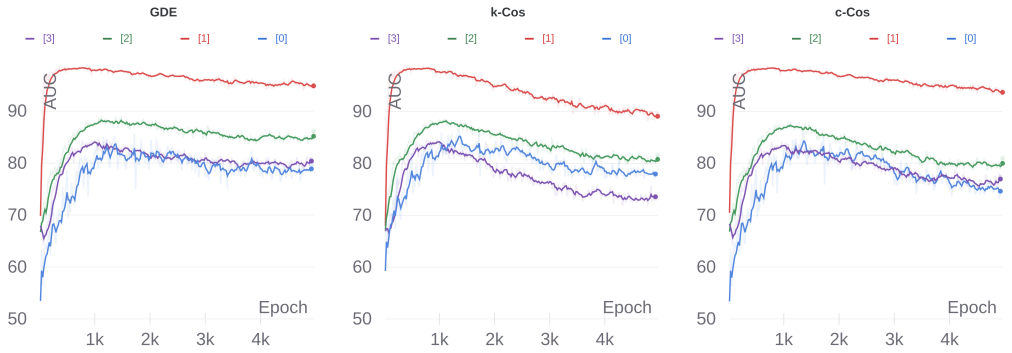

Figure 7: Same experiment as Figure 3 but after adding the normalization proposed in Section 2.3.

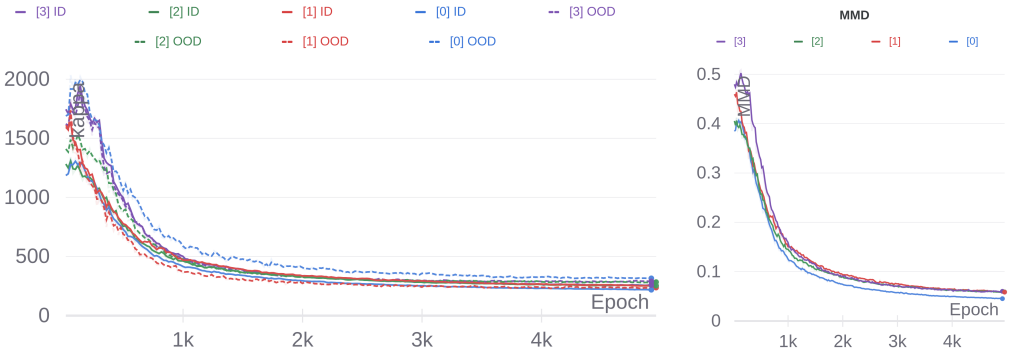

Figure 8: Same experiment as Figure 4 but after adding the normalization proposed in Section 2.3.

- conv_block_n: The output feature map from block$n$ of the convolutional backbone. Which is first directly flattened from 2D to 1D before evaluation.
- conv_block_n (1x1): Same as conv_block_n but after pooling to a size of 1x1.
- head_layer_n: The output feature map from layer n of the projection head.
- All Conv blocks: The sum of the scores of all convolutional blocks, using the distance specified in column, then summed across table columns merged in table.
- All blocks: The sum of the scores of all network internal feature maps, using the distance specified in column, then summed across table columns merged in table.

We also show results with different metric functions, other than the Cosine Mahalanobis distance presented in section 3.3. In the following tables we examine 5 different distance functions to replace $\mathcal{D}_{MC}$:

- *Cos1*: This is same as $\mathcal{D}_{MC}$ but evaluated directly at the original space without projection into the Mahalinobis space.
- *Mah+Cos1*: This is same as $\mathcal{D}_{MC}$
- *Cos2*: Instead of using the cosine distance to closest training vector, we use the distance to the mean of all the training set vectors.
- *Mah+Cos2*: Same as *Cos2*, but evaluted after projection into the Mahalinobis space.
- *GDE*: A Gaussian Kernel density estimator, we use the same configuration as in Sohn et al. (2020)

Tables 8,9,10 show an extensive evaluation of presented models at most internal layers and a variety of scoring metrics. We should highlight the fact that each number presented in those tables is the average across all classes in the dataset, with one experiment per class i.e. 10 experiments for Cifar 10, and 20 experiments for Cifar100. We can notice:

		CIFAR10	CIFAR100	IN30	fMNIST	SteelDefects	SVHN
Ours	1. NSA (SimSiam with norm)	91.54	84.09	80.88	95.15	74.25	94.91
	2. SimCLR(w norm, no neg aug)	89.27	83.95	75.17	94.61	76.35	93.84
	3. SimCLR(w/o norm, no neg aug)	86.49	80.51	75.17	94.61	76.35	93.84
	4. SimCLR(w/o norm, neg aug)	91.12	86.68	76.97	95.47	76.36	92.17
Pretraining / FT	5. Pretrained ResNet18* [Ours]	93.63	92.64	99.78	94.35	62.24	61.06
	6. Pretrained ResNet50* [Ours]	94.45	94.68	99.91	95.16	63.52	66.28
	7. Pretrained ResNet152* [Ours]	95.82	95.21	99.96	94.99	63.35	63.41
	8. PT ResNet50 (DROC (Sohn et al., 2020))*	80.0	83.7	-	91.8	-	-
	10. ResNet152 (DN2 (Reiss et al., 2020))*	92.5	94.1	-	94.5	-	-
	11. ResNet152 (PANDA (Reiss et al., 2020))*	96.2	94.1	-	95.6	-	-
	12. Self-Sup. PT R50. Xiao et al.(Xiao et al., 2021)*	93.8	92.6	-	94.4	-	-
Dist. Shifting	13. CSI (SimCLR loss only)**	87.9	-	-	-	-	-
	13. CSI (SimCLR w neg aug only)**	90.1	86.5	83.1	-	-	-
	14. CSI (Full)**	94.3	89.6†	91.6	-	-	-
	15. DROC, Contrastive	89.0	82.4	-	93.9	-	-
	16. DROC, Contrastive DA	92.5	86.5	-	94.8	-	-
Other Methods	19. DeepSVDD (Ruff et al., 2018)	64.8	67.0	-	84.8	-	-
	20. DROCC (Goyal et al., 2020)	74.2	-	-	-	-	-
	21. Geom (Golan et al.) (Golan & El-Yaniv, 2018b)**	86.0	78.7	-	93.5	-	
	22. GOAD (Bergman) (Bergman & Hoshen, 2020)**	88.2	74.5	-	94.1	-	-
	23. ARNet (Huang) (Huang et al., 2019)**	86.6	78.8	-	93.33	-	
	24. Hendryks et al. (Hendrycks et al., 2019)**	90.1	79.8	85.7	93.2	-	-
	25.SSD (Sehwag et al., 2021)	90.0	-	-	-	-	-

Table 7: One-Class Classification Summary results reported in literature plus some of our results, all results are AUC. * indicates methods trained on external additional data. ** indicates method using test time data augmentation during evaluation.

	Cos1	Mah+Cos1	Cos2	Mah+Cos2	GDE
conv_block_2	80.29	83.58	65.35	83.52	
conv_block_3	85.87	83.59	75.81	83.53	
conv_block_4	92.21	91.20	88.35	91.12	
conv_block_1 (1x1)	71.29	70.85	68.39	67.37	67.39
conv_block_2 (1x1)	73.19	71.93	70.94	69.66	69.39
conv_block_3 (1x1)	81.46	83.47	73.60	79.66	79.32
conv_block_4 (1x1)	91.85	91.54	84.88	90.52	91.04
head_layer_1	82.64	87.84	71.18	87.56	
head_layer_2	80.62	83.08	70.50	82.76	
head_layer_3	79.67	77.41	51.93	54.64	
All Conv blocks	92.37		90.79		
All Conv blocks	92.01				
All blocks	91.62	92.23	80.50	90.86	89.96
All blocks	92.86		88.74		
All blocks	92.86				

Table 8: SimSiam w norm on Cifar10

- The Cosine Mahalanobis distance mostly outperforms most other evaluation metrics, some times by large gap.
- There is a consistent improvement associated with ensembling of feature scores. This is true accross models and accross datasets.
- We note this type of feature ensembling is computationally given essentially for free, compared with the ensembling CSI uses over random augmentations, that slows down the method.

E CIFAR 10 PER-CLASS RESULTS

Here in Table 11 we show the per class results for each of the CIFAR 10 classes trained for one-class classification against the others, and compare our NSA method (SimSIAM with Norm, no ensembling

	Cos1	Mah+Cos1	Cos2	Mah+Cos2	GDE
conv_block_2	73.36	76.42	58.54	76.39	
conv_block_3	79.31	77.83	68.39	77.80	
conv_block_4	85.47	84.46	78.19	84.40	
conv_block_1 (1x1)	66.35	67.51	60.75	64.77	64.76
conv_block_2 (1x1)	68.05	67.99	61.93	65.46	65.37
conv_block_3 (1x1)	74.26	76.49	64.76	72.78	72.75
conv_block_4 (1x1)	84.76	84.09	72.08	82.74	83.15
head_layer_1	76.95	81.46	64.50	81.24	
head_layer_2	74.78	77.46	62.95	77.29	
head_layer_3	75.23	73.86	52.65	68.15	
All Conv blocks	86.45		84.75		
All Conv blocks	85.89				
All blocks	85.19	87.11	72.50	86.17	
All blocks	87.54		85.79		
All blocks	87.25				

Table 9: SimSiam w norm on Cifar100

	Cos1	Mah+Cos1	Cos2	Mah+Cos2	GDE
conv_block_2	79.62	82.40	64.71	82.32	
conv_block_3	81.08	80.22	73.38	80.14	
conv_block_4	88.92	89.47	82.83	89.42	
conv_block_1 (1x1)	71.63	71.53	68.56	68.12	68.17
conv_block_2 (1x1)	73.85	72.67	70.60	70.12	69.79
conv_block_3 (1x1)	74.97	77.00	68.97	74.47	73.50
conv_block_4 (1x1)	86.68	89.27	75.68	88.43	88.39
head_layer_1	73.88	81.05	48.36	80.63	
head_layer_2	65.19	72.33	41.26	46.18	
All Conv blocks	90.65		89.61		
All Conv blocks	90.57				
All blocks	87.37	90.16	75.65	89.12	
All blocks	90.20		89.10		
All blocks	90.35				

Table 10: SimCLR w norm on Cifar10

	Plane	Car	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck	Mean
CSI (Tack et al., 2020), inc. ensembling augmentations	90.0	99.1	93.2	86.4	93.8	93.4	95.2	98.6	97.9	95.5	94.3
DROC, OC-SVM (Sohn et al., 2020)	88.8	97.5	87.7	82.0	82.4	89.2	89.7	95.6	86.0	90.6	89.0
DROC OC-SVM (DA) (Sohn et al., 2020)	91.0	98.9	88.0	83.2	89.4	90.0	93.5	98.1	96.5	95.1	92.5
DROC Gaussian KDE (DA) (Sohn et al., 2020)	91.0	98.9	88.0	83.2	89.4	90.2	93.5	98.1	96.5	95.1	92.4
Rot. Pred. OC-SVM, from (Sohn et al., 2020)	83.6	96.9	87.9	79.0	90.5	89.5	94.1	96.7	95.0	94.9	90.8
Denoising OC-SVM, from (Sohn et al., 2020)	81.6	92.4	75.9	72.3	82.3	83.1	86.7	91.2	78.0	91.0	83.4
RotNet Rot. Cls, (Golan & El-Yaniv, 2018a)	80.3	91.2	85.3	78.1	85.9	86.7	89.6	93.3	91.8	86.0	86.8
NSA (SimSIAM w norm) [Ours]	90.4	98.6	85.2	85.7	84.1	92.9	92.9	94.5	96.3	90.99	91.52
NSA (SimSIAM w norm) All features [Ours]	93.07	98.44	87.16	83.81	90.34	91.79	96.79	96.16	94.80	96.29	92.86

Table 11: CIFAR10 Per class results

augmentations), with (i) last layer features only and (ii) summing all features, with other recent works that also report these results.

F METHOD DETAILS

More details and notes on the methods in Table 1 Follow. In particular, we clarify that "w norm" indicates a method using our proposed modifications from Section 3.1, and "w/o norm" is the standard version of the architecture (SimCLR or SimSiam) without these modifications. "neg aug" means including strong distributionally negative shifted augmentations, using randomly the four 0, 90, 180, and 270 degree rotations, following the approach of CSI (Tack et al., 2020).

We note in particular that CSI's method combines many parts (contrastive+classification losses and scoring functions, plus ensembling) which each contribute something and add up to give good results. Our goal of instead showing baseline SimCLR results with / without the norm, and with/without shifted augmentations is to create a more straightforward baseline to compare with. We note improved results are possible with our method adding these additional features such as ensembling, but also at additional computational cost, as is the case with CSI.

On the other hand, our pretrained baselines obtain very good results equal or exceeding PANDA (Reiss et al., 2020)'s fine-tuned results on datasets that are similar in nature to ImageNet that they are pretrained on; this exemplifies the benefits of our chosen scoring metric $\mathcal{S}_{NSA}$ on good representations in general. However we also show datasets where this approach falls down, compared with self-supervised methods.

For each of the methods, here is a more detailed description of the features, networks, scoring functions etc. used for comparison:

1. NSA [ours]; features = SimSiam (with norm), ResNet18; scoring = KNN + Mahalanobis Cosine, last layer features.
2. features = SimCLR(w norm) + **NO** negative shifting augmentations, ResNet18 ; scoring = KNN + Mahalanobis Cosine, last layer features. [our baseline]
3. features = SimCLR(w/o norm) + **NO** negative shifting augmentations, ResNet18 ; scoring = KNN + Mahalanobis Cosine, last layer features. [our baseline]
4. features = SimCLR(w/o norm) + **With** strong rotation negative shifting augmentations, ResNet18; scoring = KNN + Mahalanobis Cosine, last layer features [our baseline, closest to CSI (**without ensembling**) / DROC+DA]
5. Pretrained ResNet18 (on ImageNet), **no fine-tuning**; scoring = KNN + Mahalanobis Cosine, last layer features [our baseline]
6. Pretrained ResNet50 (on ImageNet), **no fine-tuning**; scoring = KNN + Mahalanobis Cosine, last layer features [our baseline]
7. Pretrained ResNet152 (on ImageNet), **no fine-tuning**; scoring = KNN + Mahalanobis Cosine, last layer features [our baseline]
8. Pretrained ResNet-50 on ImageNet, results from DROC (Sohn et al., 2020)
9. Rippel et al. (Rippel et al., 2020) - Pretrained EfficientNet-B4 on Imagenet + Mahalanobis (no results on our benchmark datasets).

10. Reiss et al (Reiss et al., 2020) Simple baseline "DN2" - Pretrained ResNet-152 on ImageNet, **no fine-tuning** + kNN=2 scoring from last layer, presumably euclidean distance
11. Reiss et al. (Reiss et al., 2020) PANDA-EWC - Pretrained ResNet-152 on ImageNet, **Fine-tuned** last 2 layers on each dataset, with compactness loss + kNN=2 scoring from last layer, presumably euclidean distance.
12. Xiao et al.(Xiao et al., 2021) - Self-Supervised Pretrained ResNet-50. SimCLRv2, Gaussian Mixture Model / Mahalanobis distance .
13. (a) CSI, SimCLR loss only; features = SimCLR (w/o norm), ResNet18 ; scoring = Sim-only contrastive(cosine * norm). [their results Table 15]; (b) same but with full CSI loss, contrastive **With** strong rotation negative shifting augmentations; scoring = contrastive Sim-only (cosine).
14. Full CSI; features = SimCLR(w/o norm) + **With** strong rotation negative shifting augmentations, ResNet18 ; scoring = contrastive(cosine * norm) + rotation prediction, on shifted transforms, **with ensembles**. [their main results, essentially combining many parts]
15. DROC (Sohn et al., 2020) Contrastive (=Deep Representation One-class Classification); features = SimCLR(w/o norm) + **NO** negative shifting augmentations ; scoring = (OC-SVM). [their results]
16. DROC (Sohn et al., 2020) Contrastive DA (=Deep Representation One-class Classification); features = SimCLR(w/o norm) + **With** strong rotation negative shifting augmentations ; scoring = (OC-SVM). [their results]
17. ELSA (Han et al., 2021a) - **NO** negative shifting augmentations - with 1% labeled outliers (Han et al., 2021a)
18. ELSA+ (Han et al., 2021a) - **With** strong rotation negative shifting augmentations (like CSI), and **with ensembles** - also with 1% labeled outliers (Han et al., 2021a)
19. DeepSVDD, Ruff et al. (Ruff et al., 2018), with LeNet architecture Autoencoder + adaptated features. [All results apart from CIFAR10 from Reiss(Reiss et al., 2020)]
20. DROCC, Goyal et al. (Goyal et al., 2020) - LeNet architecture
21. Geom - Golan et al. (Golan & El-Yaniv, 2018b) - WRN-16-8 Arch.
22. GOAD, Bergman et al. (Bergman & Hoshen, 2020) - WRN-16-4 architecture [CIFAR 100 results from CSI with ResNet18] (Bergman & Hoshen, 2020)
23. ARNet (formerly called Inv. Trans AE) - Huang et al Huang et al. (2019)
24. Rot + Trans, Hendryks et al. (Hendrycks et al., 2019) - WRN-16-4 architecture (CIFAR 100 results from CSI with ResNet18; IN30 are ResNet18 Rot+Trans+Attn+Resize; fMNIST results from Reiss (Reiss et al., 2020))
25. SSD (Sehwag et al., 2021); features = SimCLR(w/o norm) + **NO** negative shifting augmentations ; scoring = Mahalanobis [their results]