Supplementary Material for: No Shifted Augmentations (NSA): compact distributions for robust self-supervised Anomaly Detection

Anonymous WACV 2023 Algorithms Track submission

Paper ID 577

A. Plots for evaluations during training

Here we show the detailed version of the summary (mean) plots presented in Figure 2 in the main text. They show the same curves for 4 distinct classes of CIFAR10, and also under 3 different metrics. They appear in Figure 1 (SimCLR AUC and Linear probes during training), Figure 2 (SimCLR Kappa and MMD during training), Figure 3 (SimSiam AUC during training), Figure 4 (SimSiam Kappa and MMD during training), and Figure 5 (SimCLR with norm AUC). The same findings as in the main text are seen to hold across these detailed variants.

B. Dataset details

The main datasets we test on are: CIFAR 10, CIFAR 100 and ImageNet30, fashion-MNIST following the splits and protocols specified in [32] and [31]; we add SVHN using the same basic protocol of one class as inlier v.s. the remaining as outliers.We resized images to 32 x 32 for all datasets apart from ImageNet30 which uses the standard ImageNet ResNet-18 architecture's transformation of a 224-pixel center-cropped region from the 256 x 256 input image.

C. Comparison of scoring with different feature evaluations, metrics & ensembling

As discussed in Section 2.5, we compare the default scoring used in our main experiments (the encoder's last layer features using $\mathcal{S}_{k\text{-}Cos}$) with a feature ensembling 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:

- 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.
- Ens.: The sum of *k-Cos* and *k-Cos (Mah)* for 2D feature maps (e.g. convolutional feature maps) and *c-Cos* and *c-Cos (Mah)* for 1D feature maps (e.g. projection heads).

We also investigate 5 different metric functions for computing the OOD score, namely:

- *k-Cos*: Cosine distance to closest (k=1) training vector, introduced in Section 2.5

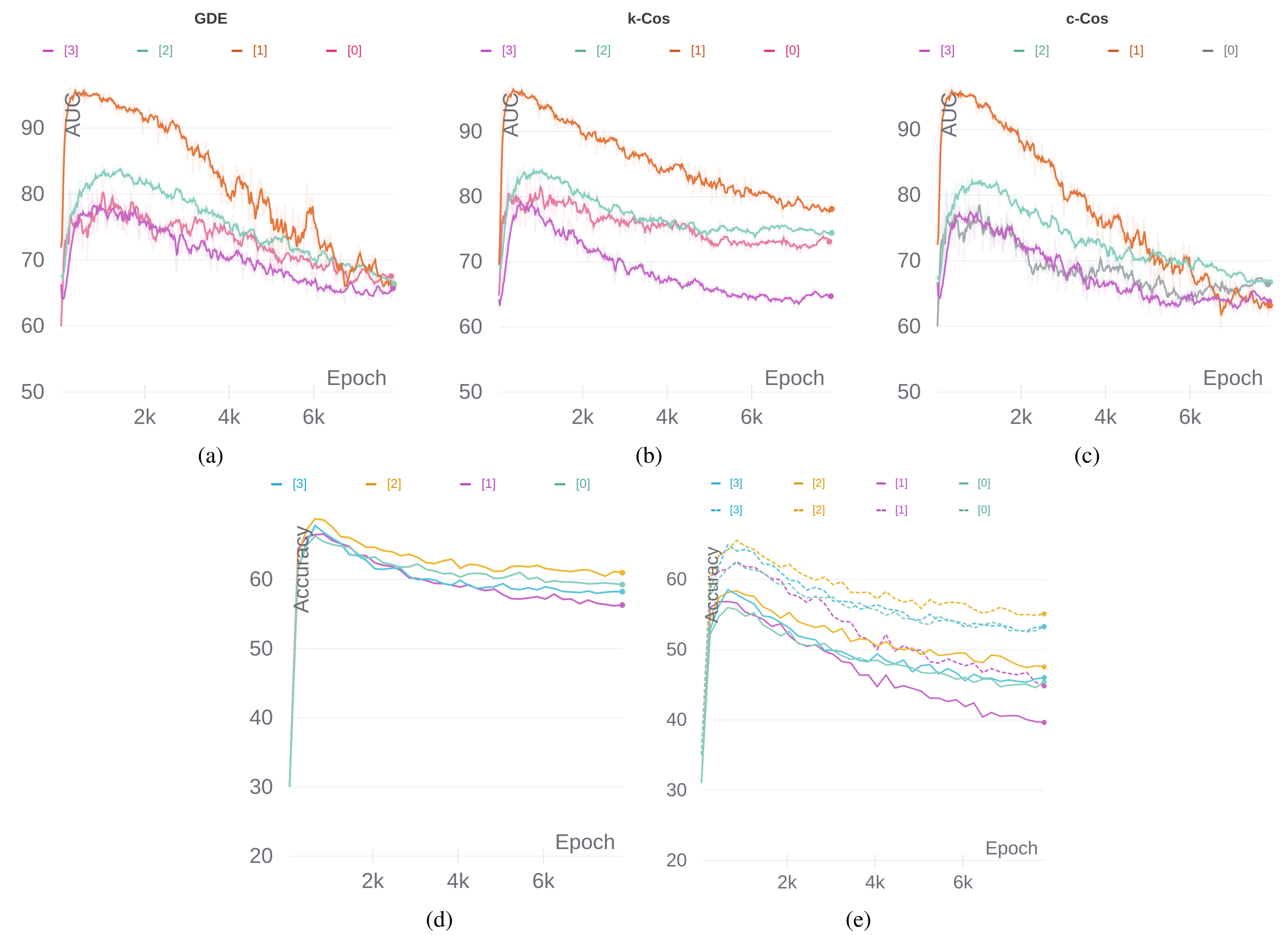

Figure 1: Training SimCLR on classes 0,1,2,3 from CIFAR10 and evaluating the representation during training. (a) Using Gaussian density estimation (GDE) [25]. (b) k-Cos cosine distance to the closest (k=1) point in ID data. (c) c-Cos cosine distance to the mean direction in ID data. Both these are defined in more detail in Section 2.5. (d) Linear evaluation [35] of learned embedding. (d) Weighted nearest neighbor classifier (k-NN) [33]. Solid lines represent $k = 1$ and dashed lines represent $k = 20$

- *k-Cos (Mah)*: Same but evaluated in Mahalanobis space.
- *c-Cos*: Cosine distance to the mean of all the training set vectors, introduced in Section 2.5
- *c-Cos (Mah)*: Same but evaluated in Mahalanobis space.
- *GDE*: A Gaussian Kernel density estimator, we use the same configuration as in [31].

Tables 2,3,4 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 CIFAR10, and 20 experiments for CIFAR100. We can notice:

- The Cosine Mahalanobis distance mostly outperforms most other evaluation metrics, sometimes a by large gap.
- There is a consistent improvement associated with ensembling of feature scores. This is true across models and across datasets.
- We note this type of feature ensembling can be considered free, computation-wise, compared with the ensembling used in e.g. CSI, that slows down the method significantly at inference.

(a) (b)

Figure 2: Training SimCLR on classes 0,1,2,3 from CIFAR10 and monitoring the learned vMF ID and OOD representations during training. (a) κ progress during training; solid lines are ID data and dashed are OOD data. (b) The MMD [11] between the learned representation and samples from uniform distribution on a unit hypersphere (lower is more uniform).

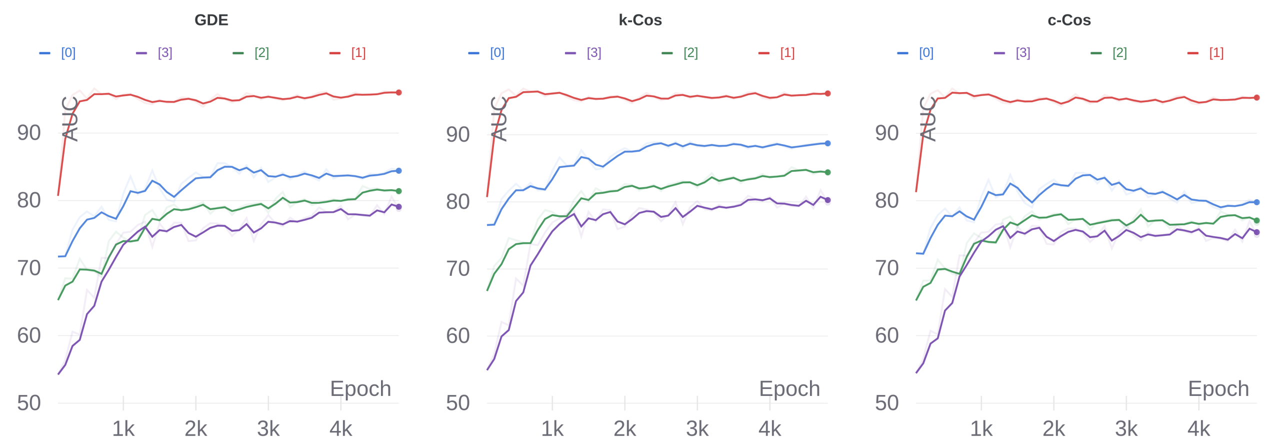

Figure 3: Same experiment as Figure 1 but with SimSiam.

D. CIFAR 10 per-class results

Here in Table 5 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 augmentations), with (i) last layer features only and (ii) summing all features, with other recent works that also report these results.

E. Additional pollution results

Table 6 shows additional CIFAR10 results in the presence of pollution, also more comparisons.

F. Additional comparison with previous results from literature

Table 1 gives a more detailed overview of related OOD / One Class Classification results from the literature, and comparison with variants of our baseline (ensemble free) variants of SimSiam, SimCLR, along with our implementation of pretrained ResNets (as an additional basline for comparison to other pretrained methods.

The pretrained results also serve to indicate where these types of approaches can fall down, and the issues in a with fairly comparing these pretrained methods with from-scratch trained approaches in the Anomaly/OOD detection context (It is unsurprising that pretraining a representation on ImageNet does well at distinguishing “unseen” ImageNet30 classes in a

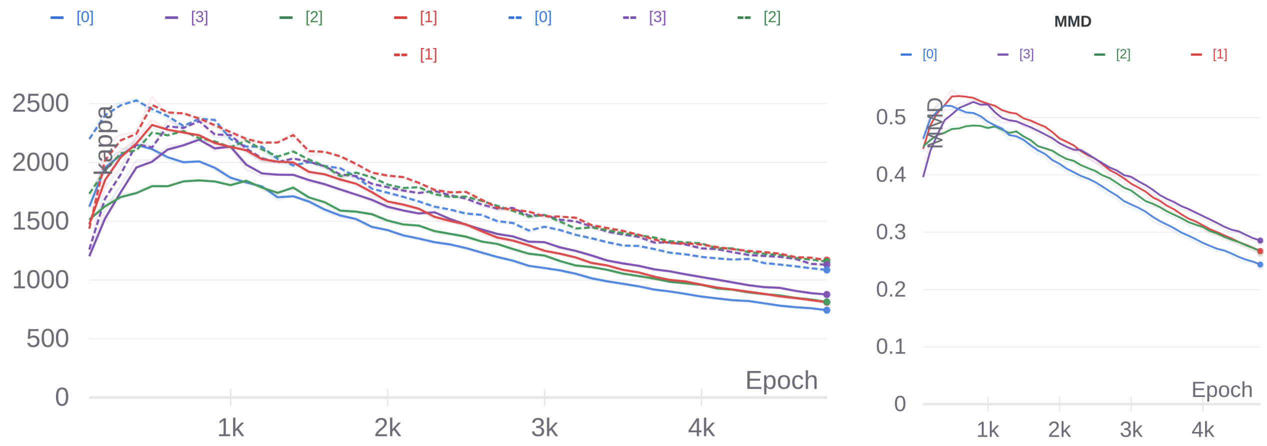

Figure 4: Same experiment as Figure 2 but with SimSiam.

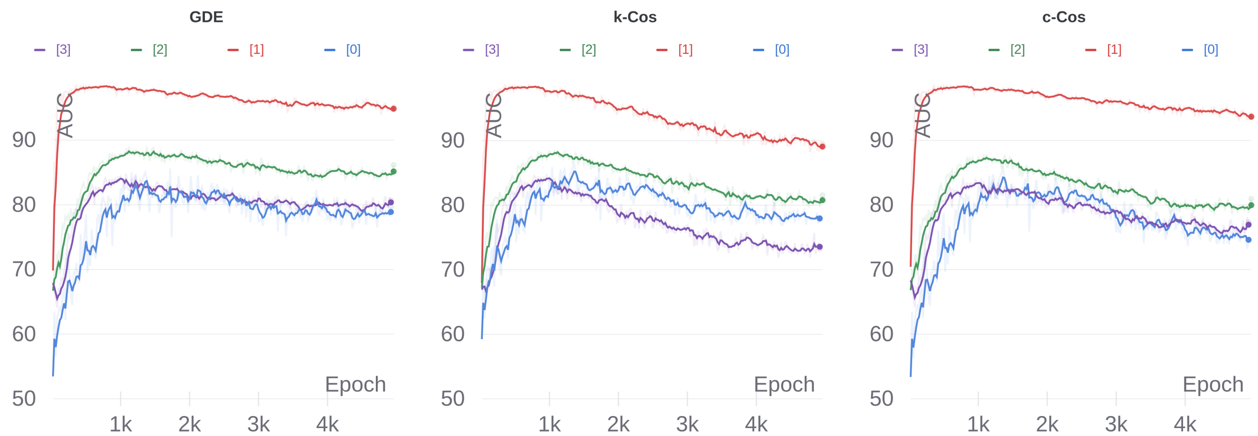

Figure 5: Same experiment as Figure 1 but after adding the normalization proposed in Section 2.3.

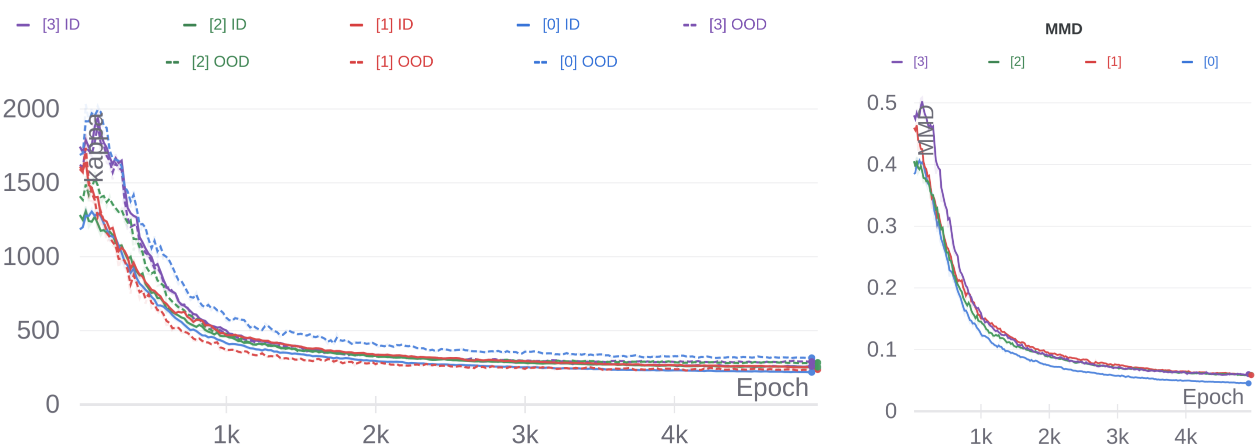

Figure 6: Same experiment as Figure 2 but after adding the normalization proposed in Section 2.3.

One-Class classifier scenario, and similarly for e.g. CIFAR, where the same types of class are present; however SVHN is less similar and therefore the pretrained represenation is less useful).

More details and notes on these competing methods follow. In particular, we clarify that "**(n)**" indicates a method using

		CIFAR10	CIFAR100	IN30	fMNIST	SVHN
Ours	1. **SS(n)** = SimSiam with norm, aka “NSA”	91.54	84.09	80.88	95.15	94.91
	2. **SC(n)** = SimCLR(w norm, no neg aug)	89.27	83.95	75.17	94.61	93.84
	3. **SC)** = SimCLR(w/o norm, no neg aug)	86.49	80.51	75.17	94.61	93.84
	4. **SS(-)** = SimCLR(w/o norm, neg aug)	91.12	86.68	76.97	95.47	92.17
Pretraining / FT	5. Pretrained ResNet18* [Ours]	93.63	92.64	99.78	94.35	61.06
	6. Pretrained ResNet50* [Ours]	94.45	94.68	99.91	95.16	66.28
	7. Pretrained ResNet152* [Ours]	95.82	95.21	99.96	94.99	63.41
	8. PT ResNet50 (DROC [31])*	80.0	83.7	-	91.8	-
	10. ResNet152 (DN2 [24])*	92.5	94.1	-	94.5	-
	11. ResNet152 (PANDA [24])*	96.2	94.1	-	95.6	-
	12. Self-Sup. PT R50. Xiao et al.[34]*	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 [27]	64.8	67.0	-	84.8	-
	20. DROCC [9]	74.2	-	-	-	-
	21. Geom (Golan et al.) [8]**	86.0	78.7	-	93.5	
	22. GOAD (Bergman) [1]**	88.2	74.5	-	94.1	-
	23. ARNet (Huang) [17]**	86.6	78.8	-	93.33	
	24. Hendryks et al. [16]**	90.1	79.8	85.7	93.2	-
	25.SSD [30]	90.0	-	-	-	-

Table 1: One-Class Classification Summary results reported in the literature on various datasets, plus some of our results; all figures are AUC. * indicates methods trained on external additional data, which may overlap in scope with the “unseen” OOD data. ** indicates method using test-time data augmentation / ensembling during evaluation, which can involve drastically slower inference.

our proposed modifications, and “w/o norm” is the standard version of the architecture (SimCLR or SimSiam) without these modifications. “‘**(-)**” means including strong distributionally negative shifted augmentations, using randomly the four 0, 90, 180, and 270 degree rotations, following the approach of CSI [32].

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 [24]’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 $\mathcal{S}_{NSA} = \mathcal{S}_{k\text{-}Cos}$ 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,

	k-Cos	k-Cos (Mah)	c-Cos	c-Cos (Mah)	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				
Ens.	92.5				

Table 2: Different feature ensembling methods: SimSiam w norm on Cifar10. Note that items spanning multiple columns imply summation of the corresponding features.

	k-Cos	k-Cos (Mah)	c-Cos	c-Cos (Mah)	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				
Ens.	86.6				

Table 3: Different feature ensembling methods: SimSiam w norm on Cifar100

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]

	k-Cos	k-Cos (Mah)	c-Cos	c-Cos (Mah)	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				
Ens.	90.3				

Table 4: Different feature ensembling methods: SimCLR w norm on Cifar10

	Plane	Car	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck	Mean
CSI [32], 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 [31]	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) [31]	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) [31]	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 [31]	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 [31]	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, [8]	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 5: CIFAR10 Per class results

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 [31]

9. Rippel et al. [26] - Pretrained EfficientNet-B4 on Imagenet + Mahalanobis (no results on our benchmark datasets).

10. Reiss et al [24] Simple baseline "DN2" - Pretrained ResNet-152 on ImageNet, **no fine-tuning** + kNN=2 scoring from last layer, presumably euclidean distance

11. Reiss et al. [24] 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.[34] - 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]

	p=0	p=0.05	p=0.10	Δ (p=0.1 - p=0)
SimSiam(w norm)	91.54	88.08 pjk17e23	86.21 6e8arxn0	5.33
SimSiam(w/o norm)	89.56	85.21 suo8biik	81.79 olj3ysx3	7.77
SimCLR(no neg, w norm)	89.27	83.174n3hybb	77.49ypprggnv	11.78
SimCLR(no neg, w/o norm)	86.49	71.1z0mshamg	63.5 kjok11ok	25.99
SimCLR (neg aug, w norm)	92.91	86.1 dzr6vm0n	83.38 nafeskrq	9.53
SimCLR (neg aug, w/o norm)	91.12	81.5 lyaevmo5	77.94 4j3xb30j	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 [31]	89	76.5	73.0	16.0
DROC (DA) [31]	92.5	85.0	80.5	12.0
CSI (results from [14])	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 6: 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.

15. DROC [31] Contrastive (=Deep Representation One-class Classification); features = SimCLR(w/o norm) + **NO** negative shifting augmentations ; scoring = (OC-SVM). [their results]
16. DROC [31] 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 [14] - **NO** negative shifting augmentations - with 1% labeled outliers [14]
18. ELSA+ [14] - **With** strong rotation negative shifting augmentations (like CSI), and **with ensembles** - also with 1% labeled outliers [14]
19. DeepSVDD, Ruff et al. [27], with LeNet architecture Autoencoder + adaptated features. [All results apart from CIFAR10 from Reiss[24]]
20. DROCC, Goyal et al. [9] - LeNet architecture
21. Geom - Golan et al. [8] - WRN-16-8 Arch.
22. GOAD, Bergman et al. [1] - WRN-16-4 architecture [CIFAR 100 results from CSI with ResNet18] [1]
23. ARNet (formerly called Inv. Trans AE) - Huang et al [17]
24. Rot + Trans, Hendryks et al. [16] - WRN-16-4 architecture (CIFAR 100 results from CSI with ResNet18; IN30 are ResNet18 Rot+Trans+Attn+Resize; fMNIST results from Reiss [24])
25. SSD [30]; features = SimCLR(w/o norm) + **NO** negative shifting augmentations ; scoring = Mahalanobis [their results]

G. Detailed ablation study

In Table 7 we show an extensive ablation study demonstrating results on CIFAR10, CIFAR100 and fMNIST of each variant of the methods we study, with and without our normalization enhancements, for different pollution settings, and under 5 different feature evaluations metrics (same as those in Appendix C) and the feature ensemble (Ens.) proposed in Section 2.5. We would like to stress that this study includes training 640 different models and evaluating each model using 6 different metrics.

We can take a few important notes:

- For all examined situations, the proposed normalization always brings a noticeable improvement. The highest improvement is for SimCLR in the presence of pollution; this is consistent with our analysis in the main text.
- For different datasets, different algorithms, and different pollution ratios, the proposed ensembling has the best performance most of the time, and the second best otherwise, which shows how generic our proposed ensembling scheme is.
- k-Cos (Mah) is the metric most often getting second best results among all evaluated, and as such was chosen for our simple baseline (ensemble-free comparison).
- Although in many times it is not the best or second best metric, c-Cos is the metric that gets the largest boost in performance after applying normalization, which is expected because as the ID representation gets more compact, the center is much more representative of the distribution.

H. Background and related work

H.1. Anomaly and OOD detection

Outlier detection is important in a variety of practical tasks, such as detecting problems in a production process, detecting security events, and acting upon novelties. The most general case is unsupervised novelty detection (or poisoned data): there are outliers in the training set, and we have no information about them. Then there are degrees of supervision, semi-supervised (a few outliers are labeled) and fully supervised (all outliers are labeled). Furthermore, in the sub-field of anomaly detection it is assumed that there are no outliers in the training data. A challenge in OOD detection is that the notion of in- and out-of-distribution is not well defined, and task-dependent. A good OOD method would generalize to different notions of out-of-distribution and datasets, e.g. with respect to color, style, perspective and content. Recent approaches in Anomaly/OOD detection can be categorized in four groups:

- **Density-based methods** are based on the assumption that models trained to fit the in-distribution data will be less confident on out-of-distribution data in terms of likelihood of the outputs. Using the likelihood as a detection score has been shown to be a weak metric [19, 5], and modifications such as entropy, energy [7, 10] and WAIC [5] have been proposed.
- **One-class classifiers** are a classic approach for outlier detection and have been adopted to deep learning settings. They find a decision boundary that separates ID and OOD samples. A margin is introduced to allow generalization [29, 27].
- **Reconstruction-based methods** model the ID training data by training an encoder and decoder network to reconstruct the in-distribution data. The reconstruction will generalize less for OOD data such that the reconstruction loss can be used a the detection metric. Auto-encoders [36, 23] and GANs [28, 6, 22].
- **Self-supervised methods** leverage the representations learned from self-supervision, combined with different detection scores. The current state-of-the-art in OOD detection is CSI [32], using representations learned by SimCLR [2] and the distance to the closest training point in latent space as a detection score. Other approaches train networks with predefined tasks such as permutations of image patches or rotations [8, 16, 1]].

H.2. Self-supervised learning (SSL)

SSL is a form of unsupervised learning, tackling it through means of supervised learning from pseudo-labels that can easily be generated. One line of computer vision research uses augmentations as pseudo-labels. These augmentations can be generated at no additional human cost, for example 90 degree rotations results in four labels. In jigsaw tasks [20] the image is split in grids, for example 2x2 or 3x3, and shuffled, the resulting position is the prediction target.

Another more recent direction is constrastive learning [21, 15, 18, 13]. In SimCLR [2, 3] every image in a batch is augmented twice, and the objective is to minimize the distance of the latent representations of the same origin image, while maximizing the distance to other images in the batch.

Another recent SSL direction is non-contrastive or positive samples only SSL. Bootstrap Your Own Latent (BYOL) [12] was the first example of this class of algorithms to achieve very competitive results, that even surpasses SimCLR. BYOL gets away from the problem of representation collapse (first enemy of SSL, usually handled by negative samples) by introducing

Table 7: Detailed ablation study. Here **p** is the ratio of outlier data inside the training set. **Norm** is whether normalization is applied or not. **Ens.** is our proposed feature ensembling. Best results are in **bold**. Second best results are underlined.

Data	Algo	p	Norm	k-Cos	k-Cos (MAH)	c-Cos	c-Cos (MAH)	GDE	Ens.
C10	BYOL	0		84.9	88.5	51.3	83.8	86.2	**88.9**
		0		89.9	90.5	79.5	89.0	89.5	**91.9**
		0.1		62.7	82.9	63.4	82.0	80.5	**85.3**
		0.1		77.5	85.3	78.5	83.6	83.0	**88.3**
	SimSiam	0		86.5	89.5	59.3	85.8	87.9	**90.1**
		0		91.6	91.7	84.9	90.5	91.0	**92.5**
		0.1		65.8	83.2	67.9	81.5	80.7	**84.3**
		0.1		80.6	86.3	82.5	84.9	84.3	**88.4**
	SimCLR	0		79.3	86.3	45.6	84.9	84.7	**87.8**
		0		86.0	88.9	75.2	87.9	88.0	**90.3**
		0.1		53.1	65.6	62.6	65.9	64.7	**80.3**
		0.1		68.8	79.8	80.0	78.7	78.2	**86.7**
	SimCLR(-)	0		88.6	90.7	79.1	90.0	90.1	**91.1**
		0		91.2	92.9	87.3	92.5	92.6	**93.0**
		0.1		71.4	79.6	83.4	80.8	80.2	**86.3**
		0.1		77.9	83.9	87.2	83.6	83.1	**87.8**
C100	BYOL	0		78.6	79.6	49.4	74.8	76.8	**81.3**
		0		81.0	80.2	59.5	77.5	78.2	**83.4**
		0.1		74.1	76.2	50.8	71.4	73.4	**77.7**
		0.1		78.0	77.8	59.5	75.0	75.7	**80.7**
	SimSiam	0		79.7	81.4	55.6	77.1	78.8	**83.3**
		0		84.5	84.3	72.1	82.7	83.1	**86.6**
		0.1		73.1	78.9	54.7	75.4	75.4	**79.7**
		0.1		79.7	80.3	67.2	78.4	78.4	**82.5**
	SimCLR	0		76.3	77.0	35.5	76.3	78.2	**84.2**
		0		80.1	82.0	68.4	82.6	82.7	**86.9**
		0.1		69.2	75.9	46.5	75.0	73.8	**79.9**
		0.1		75.8	80.0	65.6	79.2	78.6	**83.0**
	SimCLR(-)	0		83.4	84.7	68.5	84.9	85.9	**87.9**
		0		85.8	87.0	80.3	87.4	87.8	**89.4**
		0.1		78.1	80.5	67.3	81.1	81.0	**84.3**
		0.1		81.2	82.8	79.5	83.9	83.6	**85.8**
fMNIST	BYOL	0		90.5	95.3	84.7	95.0	95.4	**95.9**
		0		93.2	95.1	91.2	94.8	95.0	**96.2**
		0.1		38.1	61.3	84.9	75.5	75.2	**86.7**
		0.1		48.0	73.2	86.9	80.9	80.9	**87.9**
	SimSiam	0		92.7	**95.9**	84.7	95.7	95.8	95.8
		0		93.9	95.0	90.7	94.8	94.9	**96.1**
		0.1		40.3	63.0	**88.4**	73.3	72.6	86.1
		0.1		52.7	75.3	**90.3**	80.0	79.8	87.8
	SimCLR	0		87.6	94.6	70.4	95.0	95.1	**96.1**
		0		91.3	94.9	86.8	94.9	95.0	**96.3**
		0.1		30.9	46.5	**88.4**	55.5	55.0	86.3
		0.1		34.9	53.1	**90.6**	61.1	60.6	87.5
	SimCLR(-)	0		92.7	94.7	86.3	94.5	94.5	**95.6**
		0		94.2	95.7	90.4	95.6	95.6	**95.9**
		0.1		60.9	78.7	**90.5**	83.7	83.5	90.4
		0.1		65.3	80.9	**92.1**	84.4	84.3	90.9

assymetry in the network architecute through the idea of a prediction network after the project head, it also uses an exponential moving average of the weights of the network as a target representation. SimSiam [4] made a significant analysis on BYOL and found that using a moving average of the weights wasnot necessary and just a simple `stop-grad` operation was enough.

References

[1] Liron Bergman and Yedid Hoshen. Classification-based anomaly detection for general data. In *International Conference on Learning Representations*, 2020. 5, 8, 9

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, 2020. 9

[3] 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*, 2020. 9

[4] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. *arXiv preprint arXiv:2011.10566*, 2020. 11

[5] Hyunsun Choi, Eric Jang, and Alexander A Alemi. WAIC, but Why? Generative Ensembles for Robust Anomaly Detection. Oct. 2018. 9

[6] Lucas Deecke, Robert A Vandermeulen, Lukas Ruff, Stephan Mandt, and Marius Kloft. Image Anomaly Detection with Generative Adversarial Networks. *ECML/PKDD*, 11051(3):3–17, 2018. 9

[7] Yilun Du and Igor Mordatch. Implicit generation and generalization in energy-based models. *CoRR*, 2019. 9

[8] Izhak Golan and Ran El-Yaniv. Deep anomaly detection using geometric transformations. In *Advances in Neural Information Processing Systems*, 2018. 5, 7, 8, 9

[9] Sachin Goyal, Aditi Raghunathan, Moksh Jain, Harsha Vardhan Simhadri, and Prateek Jain. Drocc: Deep robust one-class classification. In *International Conference on Machine Learning*, pages 3711–3721. PMLR, 2020. 5, 8

[10] Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. *CoRR*, 2019. 9

[11] 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. 3

[12] 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. 9

[13] 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, Bilal Piot, Koray Kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap Your Own Latent: A New Approach to Self-Supervised Learning. June 2020. 9

[14] 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*, 2021. 8

[15] K He, Kaiming He, Haoqi Fan, H Fan, Yuxin Wu, Y Wu, S Xie, Saining Xie, R Girshick arXiv preprint arXiv 1911.05722, Ross Girshick, and 2019. Momentum Contrast for Unsupervised Visual Representation Learning. *arXiv.org*, Nov. 2019. 9

[16] 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*, pages 15663–15674, 2019. 5, 8, 9

[17] Chaoqin Huang, Fei Ye, Jinkun Cao, Maosen Li, Ya Zhang, and Cewu Lu. Attribute Restoration Framework for Anomaly Detection. Nov. 2019. 5, 8

[18] Ishan Misra and Laurens van der Maaten. Self-Supervised Learning of Pretext-Invariant Representations. Dec. 2019. 9

[19] Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do Deep Generative Models Know What They Don't Know? Oct. 2018. 9

[20] Mehdi Noroozi and Paolo Favaro. Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles. Mar. 2016. 9

[21] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, 2018. 9

[22] Pramuditha Perera, Ramesh Nallapati, and Bing Xiang. OCGAN: One-class Novelty Detection Using GANs with Constrained Latent Representations. Mar. 2019. 9

[23] Stanislav Pidhorskyi, Ranya Almohsen, Donald A Adjeroh, and Gianfranco Doretto. Generative Probabilistic Novelty Detection with Adversarial Autoencoders. July 2018. 9

[24] Tal Reiss, Niv Cohen, Liron Bergman, and Yedid Hoshen. PANDA: Adapting Pretrained Features for Anomaly Detection and Segmentation. Oct. 2020. 5, 7, 8

[25] Douglas A Reynolds. Gaussian mixture models. *Encyclopedia of biometrics*, 741:659–663, 2009. 2

[26] Oliver Rippel, Patrick Mertens, and Dorit Merhof. Modeling the Distribution of Normal Data in Pre-Trained Deep Features for Anomaly Detection. May 2020. 7

[27] 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*, pages 4393–4402. PMLR, 2018. 5, 8, 9

[28] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery. Mar. 2017. 9

[29] Bernhard Schölkopf, Robert C Williamson, Alex J Smola, John Shawe-Taylor, and John C Platt. Support vector method for novelty detection. In *Advances in Neural Information Processing Systems*, 2000. 9

[30] Vikash Sehwag, Mung Chiang, and Prateek Mittal. {SSD}: A unified framework for self-supervised outlier detection. In *International Conference on Learning Representations*, 2021. 5, 8

[31] 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. 1, 2, 5, 7, 8

[32] Jihoon Tack, Sangwoo Mo, Jongheon Jeong, and Jinwoo Shin. CSI: Novelty detection via contrastive learning on distributionally shifted instances. *Advances in Neural Information Processing Systems*, 33:11839–11852, 2020. 1, 5, 7, 9

[33] 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. 2

[34] Zhisheng Xiao, Qing Yan, and Yali Amit. Do We Really Need to Learn Representations from In-domain Data for Outlier Detection? May 2021. 5, 7

[35] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European Conference on Computer Vision*, 2016. 2

[36] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection. Feb. 2018. 9