SSD: A Unified Framework for Self-Supervised Outlier Detection

Vikash Sehwag[✣], Mung Chiang[⁕], Prateek Mittal[✣]
[✣]Princeton University, USA, [⁕]Purdue University, USA

Abstract

We ask the following question: what training information is required to design an effective outlier / out-of-distribution (OOD) detector, i.e, detecting samples that lie far away from training distribution? Since unlabeled data is easily accessible for many applications, the most compelling approach is to develop detectors based on only unlabeled in-distribution data. However, we observe that existing detectors based on unlabeled data perform poorly, often equivalent to a random prediction. In contrast, existing state-of-the-art OOD detectors achieve impressive performance but require access to fine-grained data labels for supervised training. We propose SSD, an outlier detector based on only unlabeled training data. We use self-supervised representation learning followed by a Mahalanobis distance based detection in the feature space. We demonstrate that SSD outperforms existing detectors based on unlabeled data by a large margin. Additionally, SSD even achieves performance on par, and sometimes even better, with supervised training based detectors. Finally, we expand our detection framework with two key extensions. First, we formulate *few-shot OOD detection*, in which the detector has access to only one to five samples from each class of the targeted OOD dataset. Second, we extend our framework to incorporate training data labels, if available. We find that our novel detection framework based on SSD displays enhanced performance with these extensions, and achieves *state-of-the-art* performance [1].

1 Introduction

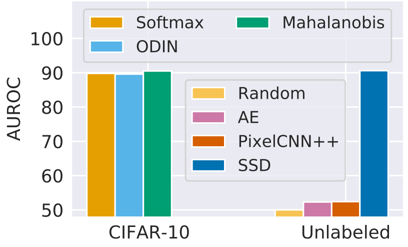


Figure 1: Detection performance with CIFAR-10 as in-distribution and CIFAR-100 as OOD dataset. SSD performs significantly better than existing detectors for unlabeled data and on par with supervised learning based detectors.

Deep neural networks are at the cornerstone of multiple safety-critical applications, ranging from autonomous driving [39] to biometric authentication [32, 16]. When trained on a particular data distribution, referred to as in-distribution data, deep neural networks are known to fail against test inputs that lie far away from training distribution, commonly referred to as outliers or out-of-distribution (OOD) samples [15, 19]. This vulnerability motivates the use of an outlier detector before feeding the input samples to the downstream network modules. However, a key question is to understand what training information is *crucial* for effective outlier detection? Will the detector require fine-grained training data labels or even access to a set of outliers in the training process?

Since neither data labels nor outliers are ubiquitous, the most compelling option is to design outlier detectors based on only *unlabeled* in-distribution data. However, we observe that existing outlier detectors based on unlabeled data fail to scale up to complex data modalities, such as images

[1] Our code is publicaly available at `https://github.com/inspire-group/SSD`

Accepted at NeurIPS 2020 Workshop: Self-Supervised Learning - Theory and Practice.

(Figure 1). For example, autoencoder (AE) [17] based outlier detectors have achieved success in applications such as intrusion detection [33], and fraud detection [43]. However, this approach achieves close to chance performance on image datasets. Similarly, density modeling based methods, such as PixelCNN++ [42], and Glow [24] are known to assign even a higher likelihood to outliers in comparison to in-distribution data [36].

In contrast, existing state-of-the-art detectors achieve high success on image datasets but assume the availability of fine-grained labels for in-distribution samples [19, 3, 30, 10, 47]. This is a strong assumption since labels, in-particular fine-grained labels, can be very costly to collect in some applications [14], which further motivates the use of unlabeled data. The inability of supervised detectors to use unlabeled data and poor performance of existing unsupervised approaches naturally give rise to the following question.

> *Can we design an effective outlier detector with access to only unlabeled data from training distribution?*

A framework for outlier detection with unlabeled data involves two key steps: 1) Learning a good feature representation with unsupervised training methods 2) Modeling features of in-distribution data without requiring class labels. For example, autoencoders attempt to learn the representation with a bottleneck layer, under the expectation that successful reconstruction requires learning an good set of representations. Though useful for tasks such as dimensionality reduction, we find that these representations are not good enough to sufficiently distinguish in-distribution data and outliers. We argue that if unsupervised training can develop a rich understanding of key semantics in in-distribution data then absence of such semantics in outliers can cause them to lie far in the feature space, thus making it easy to detect them. Recently, self-supervised representation learning methods have made large progress, commonly measured by accuracy achieved on a downstream classification task [5, 18, 38, 34, 45]. We leverage these representations in our proposed cluster-conditioned framework based on the Mahalanobis distance [31]. Our key result is that self-supervised representations are highly effective in our self-supervised outlier detection (SSD) framework where they not only perform far better than previous unsupervised representation learning methods but also perform on par, and sometimes even better, than supervised representations (Figure 1).

What if additional information, such as class labels or few outlier samples, is available? We extend SSD to take advantage of it where we first show that access to even one to five outliers can at training time can bring large improvement in SSD performance. Next, we extend SSD to also incorporate data labels (referred to as $SSD+$) leading to a state-of-the-performance.

Key Contributions. We make the following key contributions.

- *SSD for unlabeled data.* We propose SSD, an unsupervised framework for outlier detection based on unlabeled in-distribution data. We demonstrate that SSD outperforms existing unsupervised outlier detectors by a large margin while also performing on par, and sometimes even better than supervised training based detection methods. We validate our observation across four different datasets: CIFAR-10, CIFAR-100, STL-10, and ImageNet.
- *Extensions of SSD.* We provide two extensions of SSD to further improve its performance. First, we formulate *few-shot OOD detection* and propose detection methods which can achieve a significantly large gain in performance with access to only a few targeted OOD samples. Next, we extend SSD, without using any tuning parameter, to also incorporate training data labels and achieve *state-of-the-art* performance.

2 SSD: Self-supervised OOD detection

Notation. We represent the input space by $\mathcal{X}$ and corresponding label space as $\mathcal{Y}$. We assume in-distribution data is sample from $\mathbb{P}^{in}_{X \times Y}$. In the absence of data labels, it is sampled from marginal distribution $\mathbb{P}^{in}_{X}$. We sample out-of-distribution data from $\mathbb{P}^{ood}_{X}$. We denote the feature extractor by $f : \mathcal{X} \rightarrow \mathcal{Z}$, where $\mathcal{Z} \subset \mathbb{R}^d$, a function which is often parameterized by a deep neural network.

Background: Contrastive self-supervised representation learning [5, 18, 45, 38, 34]. Given unlabeled training data, it aims to train a feature extractor, by discriminating between individual instances from data, to learn a good set of representations. Using image transformations, it first generates multiple transformations/views of an image, commonly referred to as positives. Next, it optimizes to pull each instance close to the other positives while pushing away from other negatives.

Assuming that (x_i, x_j) are positive pairs for an image and $h(.)$ is a projection header, τ is the temperature, contrastive training minimizes the following loss, referred to as Normalized temperature-scaled cross-entropy (*NT-Xent*), over each batch.

$$\mathcal{L}_{batch} = \frac{1}{2N} \sum_{i=1}^{2N} -log \frac{e^{u_i^T u_j / \tau}}{\sum_{k=1}^{2N} \mathbb{1}(k \neq i) e^{u_i^T u_k / \tau}} \quad ; \qquad u_i = \frac{h(f(x_i))}{\|h(f(x_i))\|_2} \tag{1}$$

Problem Formulation: Outlier / Out-of-distribution (OOD) detection. Given a collection of samples from $\mathbb{P}_X^{in} \times \mathbb{P}_X^{ood}$, the objective is to correctly identify the source distribution, i.e., $\mathbb{P}_X^{in}$ or $\mathbb{P}_X^{ood}$, for each sample. We use the term *supervised OOD detectors* for detectors which use in-distribution data labels, i.e., train the neural network $(g \circ f)$ on $\mathbb{P}_{\mathcal{X} \times \mathcal{Y}}$ using supervised training techniques. *Unsupervised OOD detectors* aim to solve the aforementioned OOD detection tasks, with access to only $\mathbb{P}_{\mathcal{X}}^{in}$. In this work, we focus on developing effective unsupervised OOD detectors.

Outlier detection with unlabeled data using SSD. In absence of data labels, SSD consists of two steps: 1) Training a feature extractor using self-supervised representation learning methods. 2) Developing an effective OOD detector based on hidden features which isn't conditioned on data labels. We achieve the former by using recent self-supervised training methods [5, 13] and achieve latter by developing a *cluster-conditioned* detection method in the feature space.

We first partition the features for in-distribution training data in *m* clusters. We represent features for each cluster as $\mathcal{Z}_m$. We use k-means clustering method, due to its effectiveness and low computation cost. Next, we model features in each cluster independently, and calculate the following *membership score* (s_x) = $\min_m \mathcal{D}(x, \mathcal{Z}_m)$ for each test input x, where $\mathcal{D}(.,.)$ is a distance metric in the feature space. We use the membership scores for the test set of the in-distribution dataset and OOD dataset to discriminate between them. We use Mahalanobis distance for membership score as follows:

$$s_x = \min_m (z_x - \mu_m)^T \Sigma_m^{-1} (z_x - \mu_m) \tag{2}$$

where μ_m and Σ_m are the sample mean and sample covariance of features $(\mathcal{Z})$ of the in-distribution training data. We justify this choice with quantitative results in Appendix C.

Extension 1: Few-Shot OOD detection (SSD_k). In this extension of the SSD framework, we consider the scenario where access to a k samples, often $k = 1, 5$, from the OOD dataset of interest is available at the time of training. Our hypothesis is that in-distribution samples and OOD samples will be closer to other inputs from their respective distribution *in the feature space*, while lying further away from each other. We realize it by using the following formulation for membership score.

$$s_x = (z_x - \mu_{in})^T \Sigma_{in}^{-1} (z_x - \mu_{in}) - (z_x - \mu_U)^T S_U^{-1} (z_x - \mu_U) \tag{3}$$

where we use shrunk covariance estimators, instead of samples covariance, since they are a better estimator when number of samples is much lower than feature dimension [27, 44]. We also use data-augmentation to transform a set k OOD samples $\{u_1, u_2, \ldots, u_k\}$ to a set of $k \times n$ samples using data augmentation, $\mathcal{U} = \{u_1^1, \ldots, u_1^n, \ldots u_k^1, \ldots, u_k^n\}$.

Extension 2: Using data labels, when available ($SSD+$). A common theme in earlier work ([21, 47]) is to add self-supervised (L_{ssl}) and supervised (L_{sup}) training loss functions, i.e., $L_{training} = L_{sup} + \alpha L_{ssl}$, where the hyper-parameter α is chosen for best performance on OOD detection. We argue for a instance-based contrastive loss function in which labels can also be incorporated to further improve the learned representations. To this end, we use the recently proposed supervised contrastive training loss function [23], which uses labels for a more effective selection of positive and negative instances for each image. We minimize the following loss function.

$$\mathcal{L}_{batch} = \frac{1}{2N} \sum_{i=1}^{2N} -log \frac{\frac{1}{2N_{y_i}-1} \sum_{k=1}^{2N} \mathbb{1}(k \neq i) \mathbb{1}(y_k = y_i) e^{u_i^T u_k / \tau}}{\sum_{k=1}^{2N} \mathbb{1}(k \neq i) e^{u_i^T u_k / \tau}} \tag{4}$$

Not only this approach doesn't require tuning parameters (such as α) but also able to achieve *state-of-the-art* detection performance.

3 Experimental results

Experimental setup. We use ResNet50 network in major experiments and ResNet18 in ablations studies. We train each for 500 epochs, 0.5 starting learning rate with cosine decay, weight decay, and

Table 1: Comparison of SSD with different outlier detectors using only unlabeled training data.

In-distribution (Out-of-distribution)	CIFAR-10 (SVHN)	CIFAR-10 (CIFAR-100)	CIFAR-100 (SVHN)	CIFAR-100 (CIFAR-10)
Autoencoder [17]	2.5	51.3	3.0	51.4
VAE [25]	2.4	52.8	2.6	47.1
PixelCNN++ [42]	15.8	52.4	–	–
Deep-SVDD [41]	14.5	52.1	16.3	51.4
Rotation-loss [13]	97.9	81.2	94.4	50.1
SSD	**99.6**	**90.6**	**94.9**	**69.6**
SSD_k (k = 5)	**99.7**	**93.1**	**99.1**	**78.2**

Table 2: Comparison of $SSD+$ with state-of-the-art detectors based on supervised training.

In-distribution (Out-of-distribution)	CIFAR-10 (CIFAR-100)	CIFAR-10 (SVHN)	CIFAR-100 (CIFAR-10)	CIFAR-100 (SVHN)
Softmax-probs [19]	89.8	95.9	78.0	78.9
ODIN[30]†	89.6	96.4	77.9	60.9
Mahalnobis [29]†	90.5	99.4	55.3	94.5
Residual Flows [49]†	89.4	99.1	77.1	97.5
Outlier exposure [20]	93.3	98.4	75.7	86.9
Rotation-loss + Supervised [21]	90.9	98.9	–	–
Contrastive + Supervised [47]*	92.9	99.5	**78.3**	95.6
$SSD+$	**93.4**	**99.9**	**78.3**	**98.2**
SSD_k+ (k = 5)	**94.1**	**99.6**	**84.1**	**97.4**

* Uses 4× wider network, † Requires additional OOD data for tuning.

batch size set to 1e-4, and 512, respectively. We use NT-Xent loss [5] for self-supervised training and measure performance with three performance metrics, namely FPR (at TPR=95%), AUROC, and AUPR. We find the choice of the number of clusters dependent on which layer we extract the features from in the Residual neural networks. We achieve best results when modeling features from last layer using a single cluster. We present detailed setup and results in Appendix B, C.

Performance of SSD

- *Comparison with unsupervised detectors.* We find that SSD improves average AUROC by up to 55, compared to standard outlier detectors based on Density modeling (PixelCNN++ [42]), input reconstruction (Auto-encoder [17], Variational Auto-encoder [25]), and One-class classification (Deep-SVDD [41]) (Table 1). We also experiment with Rotation-loss [13], a non-contrastive self-supervised training objective. We find that SSD with SimCLR [5] achieves 9.6% higher average AUROC in comparison to using Rotation-loss [13]. We also ablate along individual parameters in self-supervised training and report results in Appendix C.4.
- *Comparison with supervised representations.* We earlier asked the question *whether data labels are even necessary to learn representations crucial for OOD detection?* To answer it, we compare SSD with a supervised network, trained with an identical budget as SSD while also using Mahalanobis distance in the feature space, across sixteen different pairs of in-distribution and out-of-distribution datasets, and report our results in Table 4, 5 in Appendix C.9. We observe that self-supervised representation even achieve better performance than supervised representation for 56% of the tasks.

Few-shot OOD detection (SSD_k). Compared to baseline SSD detection, one-shot and five-shot settings improves the average AUROC, across all OOD datasets, by 1.6 and 2.1, respectively (Table 1, 5). In particular, we observe the large gain with CIFAR100 as in-distribution and CIFAR-10 as OOD where five-shot detection improves the AUROC from 69.6 to 78.3. We find shrunk covariance estimator most critical in the success of our approach. Use of shrunk covariance estimation itself improves the AUROC from 69.6 to 77.1. Then data augmentation further improves it 78.3 for the five-shot detection. With access access to 1000 outliers, similar to earlier works [30, 29], it achieves the 89.4 AUROC, which is 14.2% higher than the current state-of-the-art [47].

Success when using data labels ($SSD+$). Now we integrate labels of training data in our framework and compare it with existing state-of-the-art detectors (Table 2). Our approach improves the average AUROC by 0.8 over the previous state-of-the-art detector. For example, using labels in our framework improves the AUROC of Mahalanobis detector from 55.5 to 72.1 for CIFAR-100 as in-distribution and CIFAR-10 as the OOD dataset. Using the simple softmax probabilities, training a two-layer MLP on learned representations further improves the AUROC to 78.3. Combining $SSD+$ with a five-shot OOD detection method further brings a gain of 1.4 in the average AUROC.

4 Discussion and Conclusion

To understand why contrastive self-supervised learning is effective for outlier detection, we vary the temperature (τ) parameter in $NT-Xent$ loss. A smaller value of temperature quickly saturates the loss, discouraging it to improve the feature representations. We also find that SSD performance monotonically decreases with decrease in temperature (Appendix C.6). A compelling advantage of unsupervised learning is to learn from unlabeled data, which can be easily collected. We conduct another experiment with the STL-10 dataset, where in addition to the 5k training images, we also use additional 10k images from the unlabeled set. This improves the AUROC from 94.7 to 99.4 for CIFAR-100 as the OOD dataset, further demonstrating the success of SSD in leveraging unlabeled data (Appendix C.8). In conclusion, our framework provides an effective & flexible approach for outlier detection using unlabeled data.

References

[1] D. Abati, A. Porrello, S. Calderara, and R. Cucchiara. Latent space autoregression for novelty detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 481–490, 2019.

[2] J. An and S. Cho. Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE*, 2:1–18, 2015.

[3] A. Bendale and T. E. Boult. Towards open set deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1563–1572, 2016.

[4] R. Chalapathy, A. K. Menon, and S. Chawla. Anomaly detection using one-class neural networks. *arXiv preprint arXiv:1802.06360*, 2018.

[5] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, 2020.

[6] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[7] A. Coates, A. Ng, and H. Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223, 2011.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. Ieee, 2009.

[9] T. DeVries and G. W. Taylor. Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint arXiv:1802.04865*, 2018.

[10] A. R. Dhamija, M. Günther, and T. Boult. Reducing network agnostophobia. In *Advances in Neural Information Processing Systems*, pages 9175–9186, 2018.

[11] R. El-Yaniv and Y. Wiener. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11(May):1605–1641, 2010.

[12] Y. Geifman and R. El-Yaniv. Selective classification for deep neural networks. In *Advances in neural information processing systems*, pages 4878–4887, 2017.

[13] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018.

[14] A. Google AI Pricing. *Data Labelling Pricing - Google AI Platform*, 2020.

[15] F. E. Grubbs. Procedures for detecting outlying observations in samples. *Technometrics*, 11(1):1–21, 1969.

[16] M. Günther, S. Cruz, E. M. Rudd, and T. E. Boult. Toward open-set face recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. IEEE*, 2017.

[17] S. Hawkins, H. He, G. Williams, and R. Baxter. Outlier detection using replicator neural networks. In *International Conference on Data Warehousing and Knowledge Discovery*, pages 170–180. Springer, 2002.

[18] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[19] D. Hendrycks and K. Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2017.

[20] D. Hendrycks, M. Mazeika, and T. Dietterich. Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations*, 2019.

[21] D. Hendrycks, M. Mazeika, S. Kadavath, and D. Song. Using self-supervised learning can improve model robustness and uncertainty. In *Advances in Neural Information Processing Systems*, pages 15663–15674, 2019.

[22] H. Jiang, B. Kim, M. Guan, and M. Gupta. To trust or not to trust a classifier. In *Advances in Neural Information Processing Systems*, pages 5546–5557, 2018.

[23] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, 2020.

[24] D. P. Kingma and P. Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in neural information processing systems*, pages 10215–10224, 2018.

[25] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In Y. Bengio and Y. LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[26] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[27] O. Ledoit and M. Wolf. Honey, i shrunk the sample covariance matrix. *The Journal of Portfolio Management*, 30(4):110–119, 2004.

[28] K. Lee, H. Lee, K. Lee, and J. Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *International Conference on Learning Representations*, 2018.

[29] K. Lee, K. Lee, H. Lee, and J. Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, pages 7167–7177, 2018.

[30] S. Liang, Y. Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*, 2018.

[31] P. C. Mahalanobis. On the generalized distance in statistics. National Institute of Science of India, 1936.

[32] I. Masi, Y. Wu, T. Hassner, and P. Natarajan. Deep face recognition: A survey. In *2018 31st SIBGRAPI conference on graphics, patterns and images (SIBGRAPI)*, pages 471–478. IEEE, 2018.

[33] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai. Kitsune: an ensemble of autoencoders for online network intrusion detection. *arXiv preprint arXiv:1802.09089*, 2018.

[34] I. Misra and L. v. d. Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717, 2020.

[35] S. Mohseni, M. Pitale, J. Yadawa, and Z. Wang. Self-supervised learning for generalizable out-of-distribution detection. In *AAAI*, pages 5216–5223, 2020.

[36] E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan. Do deep generative models know what they don't know? In *International Conference on Learning Representations*, 2019.

[37] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 5, 2011.

[38] A. v. d. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[39] M. S. Ramanagopal, C. Anderson, R. Vasudevan, and M. Johnson-Roberson. Failing to learn: autonomously identifying perception failures for self-driving cars. *IEEE Robotics and Automation Letters*, 3(4):3860–3867, 2018.

[40] J. Ren, P. J. Liu, E. Fertig, J. Snoek, R. Poplin, M. Depristo, J. Dillon, and B. Lakshminarayanan. Likelihood ratios for out-of-distribution detection. In *Advances in Neural Information Processing Systems*, pages 14707–14718, 2019.

[41] L. Ruff, R. A. Vandermeulen, N. Görnitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft. Deep one-class classification. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 4393–4402, 2018.

[42] T. Salimans, A. Karpathy, X. Chen, and D. P. Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. In *International Conference on Learning Representations*, 2017.

[43] M. Schreyer, T. Sattarov, D. Borth, A. Dengel, and B. Reimer. Detection of anomalies in large scale accounting data using deep autoencoder networks. *arXiv preprint arXiv:1709.05254*, 2017.

[44] C. Stein. Estimation of a covariance matrix. *39th Annual Meeting IMS, Atlanta, GA, 1975*, 1975.

[45] Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, and P. Isola. What makes for good views for contrastive learning. *arXiv preprint arXiv:2005.10243*, 2020.

[46] A. Vyas, N. Jammalamadaka, X. Zhu, D. Das, B. Kaul, and T. L. Willke. Out-of-distribution detection using an ensemble of self supervised leave-out classifiers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 550–564, 2018.

[47] J. Winkens, R. Bunel, A. G. Roy, R. Stanforth, V. Natarajan, J. R. Ledsam, P. MacWilliams, P. Kohli, A. Karthikesalingam, S. Kohl, et al. Contrastive training for improved out-of-distribution detection. *arXiv preprint arXiv:2007.05566*, 2020.

[48] R. Yoshihashi, W. Shao, R. Kawakami, S. You, M. Iida, and T. Naemura. Classification-reconstruction learning for open-set recognition. *arXiv preprint arXiv:1812.04246*, 2018.

[49] E. Zisselman and A. Tamar. Deep residual flow for novelty detection. *arXiv preprint arXiv:2001.05419*, 2020.

A Related work

OOD detection with unsupervised detectors. Interest in unsupervised outlier detection goes back to [15]. We categorize these approaches in three groups 1) Reconstruction-error-based detection using Auto-encoders [17, 33, 43], Variational auto-encoders [1, 2] 2) Classification based, such as Deep-SVDD [41, 11, 12] and 3) Probabilistic detectors, such as density models like Glow, PixelCNN++ [40, 36, 42, 24]. We compare with detectors from each category and find that SSD outperforms them by a wide margin.

OOD detection with supervised learning. Supervised detectors have been most successful with complex input modalities, such as images and language [4, 9, 10, 22, 48, 28]. Most of these approaches model features of in-distribution data at output [30, 19, 10] or in the feature space [29, 47] for detection. We show that SSD can achieve performance on par with these supervised detectors, without using data labels. A subset of these detectors also leverages generic OOD data to boost performance [20, 35].

Access to OOD data at training time. Some recent detectors also require OOD samples for hyperparameter tuning [30, 29, 49]. We extend SSD to this setting but assume access to only a few OOD samples, referred to as few-shot OOD detection, which our frameworks can efficiently utilize to bring large gains in performance.

In conjunction with supervised training. [46] use ensemble of leave-one-out classifier, [47] uses contrastive self-supervised training, [21] uses rotation based self-supervised loss, in conjunction with supervised cross-entropy loss to achieve state-of-the-art performance in OOD detection. Here we extend SSD, to incorporate data labels, when available, and achieve *better* performance than existing state-of-the-art.

B Additional details on experimental setup

B.1 Training and evaluation setup for deep neural networks.

We use ResNet-50 architecture for all our major experiments and ResNet-18 for ablation studies. We also provide results with ResNet-34 and ResNet-101 architecture. We use a two-layer fully connected network as the projection header ($h(.)$). To contrast with a large number of negatives, $NT-Xent$ loss requires a much larger batch size compared to the supervised cross-entropy loss function. We train it using a batch size of 512. When evaluating self-supervised models, even when we incorporate labels in SSD, we achieve the best performance when modeling in-distribution features with only a single cluster. However, for supervised training, which refers to the supervised baseline in the paper, we find that increasing the number of clusters helps. For it, we report the best of the results obtained from cluster indexes or using true labels of the data. For each dataset, we use the test set partition, if it exists, as the OOD dataset. For consistent comparison, we re-implement Softmax-probabilities [19], ODIN [30], and Mahalanobis detector [29] and evaluate their performance on the identical network, trained with supervised training for 500 epochs. We set the perturbation budget to 0.0014 and temperature to 1000 for ODIN, since these are the most successful set of parameters reported in the original paper [30]. We primarily focus on one-shot and five-shot OOD detection, i.e, set k to 1 or 5. It implies access to one and five images, respectively, from each class of the targeted OOD dataset. We create 10 randomly transformed samples from each available OOD image in the SSD_k detector. To avoid any hyperparameter selection, we use the image augmentation pipeline to be the same as the one used in training. Finally, we use the Ledoit-Wolf method [27] to estimate the covariance of OOD data.

B.2 Performance metrics for outlier detectors

We use the following performance metrics to evaluate the outlier detectors.

- **FPR at TPR=95%.** It refers to the False positive rate (= FP / (FP+TN)), when True positive rate (= TP / (TP + FN)) is equal to 95%. Effectively, its goal is to measure what fraction of outliers go undetected when it is desirable to have a True positive rate of 95%.

- **AUROC.** It refers to the area under the receiver operating characteristic curve. We measure it by calculating the area under the curve when we plot TPR against FPR.

- **AUPR.** It refers to the area under the precision-recall curve, where precision = TP / (TP+FP) and recall = TP / (TP+FN). Similar to AUROC, AUPR is also a threshold independent metric.

Algorithm 1: Self-supervised outlier detection framework (SSD)

```
Input  : $\mathcal{X}_{in}$, $\mathcal{X}_{test}$, feature extractor ($f$), projection head ($h$), Required True-positive rate ($T$), Optional:
         $\mathcal{X}_{ood}, \mathcal{Y}_{in}$ # $\mathcal{X}_{in} \in \mathbb{P}_X^{in}, \mathcal{X}_{test} \in \mathbb{P}_X^{in}, \mathcal{X}_{ood} \in \mathbb{P}_X^{ood}$
Output : Is outlier or not? $\forall x \in \mathcal{X}_{test}$
Function getFeatures($\mathcal{X}$): return $\{f(x_i)/\|f(x_i)\|_2, \forall\, x_i \in \mathcal{X}\}$;
Function SSDScore($\mathcal{Z}, \mu, \Sigma$): return $\{(z-\mu)^T \Sigma^{-1} (z-\mu), \forall\, z \in \mathcal{Z}\}$;
Function SSDkScore($\mathcal{Z}, \mu_{in}, \Sigma_{in}, \mu_{ood}, \Sigma_{ood}$):
    return $\{(z-\mu_{in})^T \Sigma_{in}^{-1} (z-\mu_{in}) - (z-\mu_{ood})^T \Sigma_{ood}^{-1} (z-\mu_{ood})\}, \forall\, z \in \mathcal{Z}$;
end
Parition $\mathcal{X}_{in}$ in training set ($\mathcal{X}_{train}$) and calibration set ($\mathcal{X}_{cal}$);
if $\mathcal{Y}_{in}$ is not available then
    $\mathcal{L}_{batch} = \frac{1}{2N} \sum_{i=1}^{2N} -log \frac{e^{u_i^T u_j / \tau}}{\sum_{k=1}^{2N} \mathbb{1}(k \neq i) e^{u_i^T u_k / \tau}}; u_i = \frac{h(f(x_i))}{\|h(f(x_i))\|_2}$; # Train feature extractor
else
    $\mathcal{L}_{batch} = \frac{1}{2N} \sum_{i=1}^{2N} -log \frac{\frac{1}{2N_{y_i}-1} \sum_{k=1}^{2N} \mathbb{1}(k \neq i) \mathbb{1}(y_k = y_i) e^{u_i^T u_k / \tau}}{\sum_{k=1}^{2N} \mathbb{1}(k \neq i) e^{u_i^T u_k / \tau}}$;
end
Train feature extractor ($f$) by minimizing $\mathcal{L}_{batch}$ over $\mathcal{X}_{train}$;
$\mathcal{Z}_{train} = \texttt{getFeatures}(\mathcal{X}_{train})$, $\mathcal{Z}_{cal} = \texttt{getFeatures}(\mathcal{X}_{cal})$
$\mathcal{Z}_{test} = \texttt{getFeatures}(\mathcal{X}_{test})$, if $\mathcal{X}_{ood}$ is available: $\mathcal{Z}_{ood} = \texttt{getFeatures}(\mathcal{X}_{ood})$;
if $\mathcal{X}_{ood}$ is not available then
    $s_{cal} = \texttt{SSDScore}(\mathcal{Z}_{cal}, \mu_{train}, \Sigma_{train})$; # Sample mean and convariance of $\mathcal{Z}_{train}$
    $s_{test} = \texttt{SSDScore}(\mathcal{Z}_{test}, \mu_{train}, \Sigma_{train})$ # membership score;
else
    # Using estimation techniques from Section 2.2 for $\mathcal{Z}_{ood}$
    $s_{cal} = \texttt{SSDkScore}(\mathcal{Z}_{cal}, \mu_{train}, \Sigma_{train}, \mu_{ood}, \Sigma_{ood})$;
    $s_{test} = \texttt{SSDkScore}(\mathcal{Z}_{test}, \mu_{train}, \Sigma_{train}, \mu_{ood}, \Sigma_{ood})$;
end
$x_i \in \mathcal{X}_{test}$ is an outlier if $s_{test}^i > (s_{cal}$ threshold at TPR = $T$);
```

B.3 Datasets used in this work

We use the following datasets in this work. Whenever there is a mismatch between the resolution of images in in-distribution and out-of-distribution (OOD) data, we appropriately scale the OOD images with bilinear scaling. When there is an overlap between the classes of the in-distribution and OOD dataset, we remove the common classes from the OOD dataset.

- **CIFAR-10 [26]**. It consists of 50,000 training images and 10,000 test images from 10 different classes. Each image size is 32×32 pixels.
- **CIFAR-100 [26]**. CIFAR-100 also has 50,000 training images and 10,000 test images. However, it has 100 classes which are further organized in 20 sub-classes. Note that its classes aren't identical to the CIFAR-10 dataset, with a slight exception with class *truck* in CIFAR-10 and *pickup truck* in CIFAR-100. However, their classes share multiple similar semantics, making it hard to catch outliers from the other dataset.
- **SVHN [37]**. SVHN is a real-world street-view housing number dataset. It has 73,257 digits available for training, and 26,032 digits for testing. Similar to the CIFAR-10/100 dataset, the size of its images is also 32×32 pixels.
- **STL-10 [7]**. STL-10 has identical classes as the CIFAR-10 dataset but focuses on the unsupervised learning. It has 5,000 training images, 8,000 test images, and a set of 100,000 unlabeled images. Unlike the previous three datasets, the size of its images is 96×96 pixels.
- **DTD [6]**. Describable Textures Dataset (DTD) is a collection of textural images in the wild. It includes a total of 5,640 images, split equally between 47 categories where the size of images range between 300×300 and 640×640 pixels.
- **ImageNet[2] [8]**. ImageNet is a large scale dataset of 1,000 categories with 1.2 Million training images and 50,000 validation images. It has high diversity in both inter- and intra-class images and is known to have strong generalization properties to other datasets.
- **Blobs.** Similar to [20], we algorithmically generate these amorphous shapes with definite edges.

[2] We refer to the commonly used ILSVRC 2012 release of ImageNet dataset.

- **Gaussian Noise.** We generate images with Gaussian noise using a mean of 0.5 and a standard deviation of 0.25. We clip the pixel value to the valid pixel range of [0, 1].
- **Uniform Noise.** It refers to images where each pixel value is uniformly sampled from the [0, 1] range.

B.4 Outline of SSD framework

We provide a pseudocode of SSD framework in Algorithm 1. While the baseline detector requires only unlabeled training data, it can also utilize data labels and outliers, if available.

C Additional experimental results

C.1 Limitations of outlier detectors based on supervised training

Existing supervised training based detector assumes that fine-grained data labels are available. What happens to the performance of current detectors if we relax this assumption by assuming that only coarse labels are present. We simulate this setup by combining consecutive classes from the CIFAR-10 dataset into two groups, referred to as CIFAR-2, or five groups referred to as CIFAR-5. We use CIFAR-100 as the out-of-distribution dataset. We find that the performance of existing detectors degrades significantly when only coarse labels are present (Figure 2).

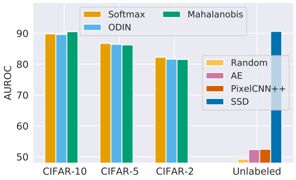


Figure 2: Existing supervised detector requires fine-grained labels. In contrast, SSD can achieve similar performance with only unlabelled data.

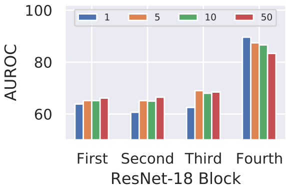


Figure 3: Relationship of AUROC with clusters depends on which layer we use as feature extractor.

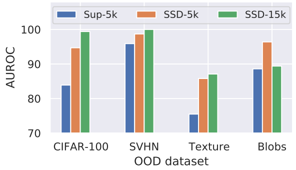


Figure 4: Using extra unlabelled training data can help to further improve the performance of SSD.

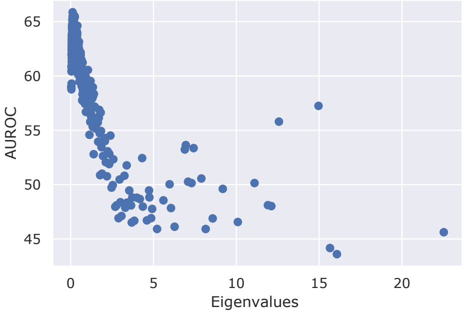


Figure 5: AUROC along every individual principle eigenvector with CIFAR-10 as in-distribution and CIFAR-100 as OOD.

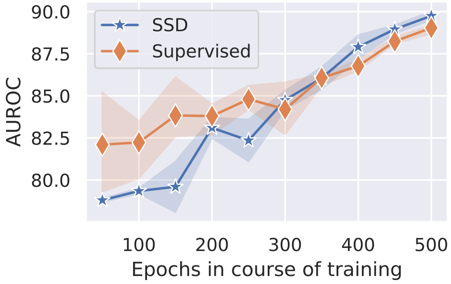


Figure 6: AUROC over the course of training with CIFAR-10 as in-distribution and CIFAR-100 as OOD set.

C.2 Choice of distance metric: Mahalanobis distance.

We use Mahalanobis distance to calculate the membership score as follows:

$$s_x = \min_m (z_x - \mu_m)^T \Sigma_m^{-1} (z_x - \mu_m) \tag{5}$$

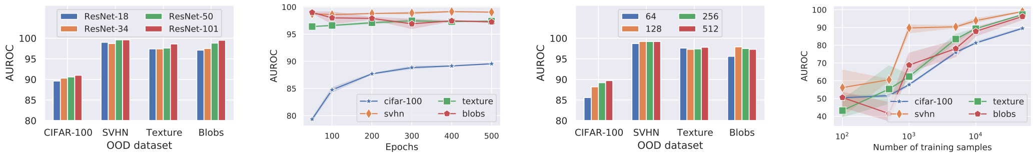
where μ_m and Σ_m are the estimated mean and covariance for features ($\mathcal{Z}$) of the in-distribution training data. We justify this choice with quantitative results in Figure 5. With eigendecomposition of sample covariance ($\Sigma_m = Q_m \Lambda_m Q_m^{-1}$), $s_x = \min_m \left(Q^T(z_x - \mu_m)\right)^T \Lambda_m^{-1} \left(Q^T(z_x - \mu_m)\right)$ which is equivalent to euclidean distance scaled with eigenvalues. We discriminate between in-distribution (CIFAR10) and OOD (CIFAR100) data along each principal eigenvector (using AUROC, higher the better). With euclidean distance, i.e, in absence of scaling, component with higher eigenvlaues weight most but provide least discriminative power. Scaling with eigenvalues remove the bias, making Mahalnobis distance effective for outlier detection in the feature space.

C.3 On choice of number of clusters

We find the choice of optimal number of clusters dependent on which layer we extract the features from in a Residual Neural networks. We demonstrate this trend in Figure 3, with CIFAR-10 as in-distribution dataset and CIFAR-100 as out-of-distribution dataset. While for the first three blocks, we find an increase in AUROC with number of clusters, the trend is reversed for the last block (Figure 3). Since last block features achieve highest detection performance, we model in the in-distribution features as a single cluster.

C.4 Ablation studies for SSD

We ablate along individual parameters in self-supervised training with CIFAR-10 as in-distribution data (Figure 7). While architecture doesn't have a large effect on AUROC for any OOD dataset, we find that number of epochs and batch-size plays a key role in detecting outliers from the CIFAR-100 dataset, which is hardest to detect among the four OOD datasets. We find an increase in the size of training data helpful in the detection of all four OOD datasets.



(a) Model size. (b) Number of epochs. (c) Batch-size. (d) Size of training set.

Figure 7: SSD performance when ablating across different training parameters under following setup: In-distribution dataset = CIFAR-10, OOD dataset = CIFAR100, Epoch = 500, Batch-size = 512.

C.5 Ablation study for few-shot OOD detection with SSD

For a few shot OOD detection, we ablate along the number of transformations used for each sample. We choose CIFAR100 as in-distribution and CIFAR-10 as OOD dataset with SSD_k, k set to five and Resnet-18 network. When increasing number of transformations from 1, 5, 10, 20, 50 the AUROC of detector is 74.3, 75.7, 76.1, 76.3, 76.7. To achieve a balance between performance and computational cost, we use 10 transformations for each sample in our final experiments.

C.6 Why contrastive self-supervised learning is effective in the SSD framework?

We focus on the NT-Xent loss function, which is parameterized by a temperature variable (τ). Its objective to pull positive instances, i.e., different transformations of an image, together while pushing away from other instances. Earlier works have shown that such contrastive training forces the network to learn a good set of feature representations. However, a smaller value of temperature quickly saturates the loss, discouraging it to improve the feature representation. We find that the performance of SSD also degrades with lower temperature, suggesting the necessity of learning a good set of feature representation for effective outlier detection (Table 3).

Table 3: Test Accuracy and AUROC with different temperature values in *NT-Xent* (Equation 1) loss. Using CIFAR-10 as in-distribution and CIFAR-100 as OOD dataset.

Temperature	0.001	0.01	0.1	0.5
Test -Accuracy	70.8	76.7	86.9	90.5
AUROC	66.7	71.6	85.5	89.5

C.7 How discriminative ability of feature representations evolves over the course of training.

We analyze this effect in Figure 6 where we compare both SSD and supervised training based detector over the course of training. While discriminative ability of self-supervised training in SSD is lower at the start, it quickly catches up with supervised representation after half of the training epochs.

C.8 Performance of SSD improves with amount of unlabeled data

With easy access to unlabeled data, it is compelling to develop detectors that can benefit from the increasing amount of such data. We earlier demonstrated this ability of SSD for the CIFAR-10 dataset in Figure 7. Now we present similar results with the STL-10 dataset. We first train a self-supervised network, and an equivalent supervised network with 5,000 training images from the STL-10 dataset. We refer to these networks by SSD-5k and Sup-5k, respectively. Next, we include additional 10,000 images from the available 100k unlabeled images in the dataset. As we show in Figure 4, SSD is able to achieve large gains in performance with access to the additional unlabeled training data.

C.9 Results with different performance metrics

We provide our experimental results for each component in SSD framework with three different performance metrics in Table 4, 5.

Table 4: Experimental results of SSD detector with multiple metrics for ImageNet dataset.

In-distribution	OOD	FPR (TPR = 95%) ↓				AUROC ↑				AUPR ↑			
		SSD	Supervised	SSD_k		SSD	Superivsed	SSD_k		SSD	Supervised	SSD_k	
				k=1	k=5			k=1	k=5			k=1	k=5
ImageNet	SVHN	1.3	0.6	0.0	0.0	99.4	99.1	100.0	100.0	98.4	96.6	100.0	100.0
	Texture	57.2	23.2	20.1	11.4	85.4	95.4	95.4	97.4	41.7	75.8	78.6	84.2
	Blobs	0.0	0.0	0.0	0.0	98.4	99.5	100.0	100.0	81.1	91.6	100.0	100.0
	Gaussian noise	0.0	0.0	0.0	0.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
	Uniform noise	0.0	0.0	0.0	0.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0

Table 5: Experimental results of SSD detector with multiple metrics for CIFAR-10, CIFAR100, and STL-10 dataset.

In-distribution	OOD	FPR (TPR = 95%) ↓							AUROC ↑							AUPR ↑						
		SSD	Supervised	SSD_k		$SSD+$	SSD_k+		SSD	Superivsed	SSD_k		$SSD+$	SSD_k+		SSD	Supervised	SSD_k		$SSD+$	SSD_k+	
				k=1	k=5		k=1	k=5			k=1	k=5		k=1	k=5			k=1	k=5		k=1	k=5
CIFAR-10	CIFAR-100	50.7	47.4	44.7	39.4	38.5	36.3	34.6	90.6	90.6	91.7	93.0	93.4	93.4	94.0	89.2	89.5	90.5	91.9	92.3	92.5	92.9
	SVHN	2.0	1.6	0.2	1.0	0.2	0.5	1.9	99.6	99.6	99.9	99.7	99.9	99.9	99.6	99.8	99.8	100.0	100.0	99.9	100.0	99.8
	Texture	14.6	12.4	5.1	2.7	7.7	6.4	3.6	97.6	97.8	98.9	99.4	98.5	98.6	99.2	95.6	96.7	98.4	99.0	97.3	98.1	98.9
	Blobs	4.3	0.0	4.4	0.0	0.0	0.0	0.0	98.8	99.9	99.7	100.0	100.0	100.0	100.0	98.4	99.9	99.4	100.0	100.0	100.0	100.0
CIFAR-100	CIFAR-10	89.4	96.4	85.3	69.4	89.5	72.1	65.2	69.6	55.3	74.8	78.3	71.0	78.2	84.0	64.5	51.8	69.3	77.2	65.3	76.2	81.7
	SVHN	20.9	28.4	2.7	4.5	7.9	11.3	11.9	94.9	94.5	99.5	99.1	98.2	97.3	97.4	98.1	97.5	99.8	99.6	99.3	99.1	99.0
	Texture	65.8	2.3	16.4	25.1	68.1	24.3	23.8	82.9	98.8	96.8	94.2	81.2	93.8	94.5	72.9	97.9	95.0	91.9	70.7	91.6	92.2
	Blobs	1.2	95.6	1.1	0.0	3.6	0.0	0.0	98.1	57.3	98.1	100.0	98.8	100.0	100.0	97.8	47.6	97.8	100.0	98.2	100.0	100.0
STL-10	CIFAR-100	29.9	73.2	32.9	32.3	40.0	8.6	14.1	94.8	84.0	90.1	90.0	92.4	98.4	97.8	95.1	82.5	92.6	92.3	93.3	98.7	98.0
	SVHN	6.6	26.2	5.8	2.4	18.6	0.4	0.3	98.7	95.7	98.7	99.4	96.9	99.8	99.9	99.5	97.6	99.5	99.7	98.9	100.0	100.0
	Texture	53.0	69.4	50.0	39.5	51.8	46.2	43.8	85.8	75.5	85.7	84.5	85.8	90.4	91.0	82.6	69.3	84.1	84.3	83.7	87.1	87.7
	Blobs	16.3	88.9	14.6	0.0	67.8	0.1	0.0	96.4	88.6	96.5	99.9	92.9	99.7	99.8	92.1	81.3	92.2	99.9	86.7	99.5	99.7