# Two Is Better Than One: Aligned Clusters Improve Anomaly Detection

**Alain Ryser**[1], **Thomas M. Sutter**[1], **Alexander Marx**[2] **& Julia E. Vogt**[1]
[1] Department of Computer Science, ETH Zurich
[2] Research Center Trustworthy Data Science and Security of the University Alliance Ruhr, Department of Statistics, TU Dortmund University

## Abstract

Anomaly detection focuses on identifying samples that deviate from the norm. When working with high-dimensional data such as images, a crucial requirement for detecting anomalous patterns is learning lower-dimensional representations that capture concepts of normality. Recent advances in self-supervised learning have shown great promise in this regard. However, many successful self-supervised anomaly detection methods assume prior knowledge about anomalies to create synthetic outliers during training. Yet, in real-world applications, we often do not know what to expect from unseen data, and we can solely leverage knowledge about normal data. In this work, we propose CON$_2$, which learns representations through context augmentations that model invariances of normal data while letting us observe samples from two distinct perspectives. At test time, representations of anomalies that do not adhere to these invariances deviate from the representation structure learned during training, allowing us to detect anomalies without relying on prior knowledge about them.

## 1 Introduction

Reliably detecting anomalies is essential in many safety-critical fields such as healthcare [Schlegl et al., 2017, Ryser et al., 2022], finance [Golmohammadi and Zaiane, 2015], industrial fault detection [Atha and Jahanshahi, 2018, Zhao et al., 2019], or cyber-security [Xin et al., 2018]. In healthcare, a common real-world example of anomaly detection (AD) is standard screenings, i.e., data from doctors who regularly examine the general population for anomalies that would indicate a health risk. These datasets predominantly comprise samples from healthy people since most screened individuals do not exhibit any diseases. Detecting anomalies in this setting is challenging, as anomalies can arise from an arbitrary set of potentially rare diseases while we predominantly encounter normal samples in the dataset. The field of AD tackles this problem by learning representations that reflect normality during training and, at test time, detecting anomalies as deviations from the learned normal structure [Ruff et al., 2021].

Recent works have demonstrated that learning a representation space containing features that tightly represent normality is essential for AD [Ruff et al., 2018, Oza and Patel, 2018, Sabokrou et al., 2020]. Current state-of-the-art methods further carefully design synthetic anomalies and explicitly encourage anomalous representations to be different from normal ones [Tack et al., 2020, Wang et al., 2023]. However, anomalies can be diverse and unexpected, which can make it difficult to simulate them in real-world settings.

This work presents a novel AD objective, CON$_2$, which learns informative, tightly clustered representations of normal samples without assuming prior knowledge about anomalies. CON$_2$ leverages *context augmentations* that let us observe samples in different contexts while preserving their normal information. Our new CON$_2$ objective clusters representations according to these new contexts while

Self-Supervised Learning - Theory and Practice (NeurIPS 2024 Workshop)

encouraging similar representations within each cluster. Our approach ensures a highly informative structure within each cluster by preserving the relative normality of samples independent of their context.

In the following, we will provide some intuition behind context augmentations and introduce the two properties that define them. We then demonstrate how CON$_2$ uses context augmentations to learn highly informative, tightly clustered representations of normal data. We further define two anomaly score functions that measure the anomalousness of new samples. Finally, we demonstrate the advantage of modeling invariances of normal data by detecting anomalies on two medical datasets.

## 2 Methods

Here, we first introduce the notion of context augmentations and demonstrate how to use them for context contrasting with CON$_2$. We then present how to use these representations to detect anomalies at test time.

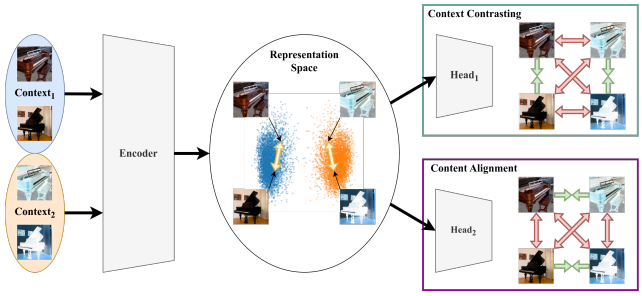



Figure 1: Overview of CON$_2$. Samples get context augmented and passed through an encoder. The *context contrasting* loss ensures context-specific representations (■ and ■ clusters) while the *content alignment* loss encourages a context independent structure (⟺) within each context cluster. We learn representations in a contrastive fashion, matching corresponding positive (⇒⇐) and discriminating between negative (⟺) pairs of representations separately for *context contrasting* and *content alignment*.

### 2.1 Context Augmentation

Our approach leverages the fact that certain transformations can transform a sample into another context, creating a distinct new view without altering the information content of the sample. Our goal is to use such transformations to learn context-specific representation clusters that align in content, letting us detect anomalies as samples that deviate from the learned structure. Here, let $X$ be our dataset, let $t_{\mathcal{C}} : \mathcal{X} \to \mathcal{X}$ be a data augmentation, and let $X' = \{t_{\mathcal{C}}(\boldsymbol{x}) \mid \boldsymbol{x} \in X\}$ be the dataset transformed by $t_{\mathcal{C}}$. The function $t_{\mathcal{C}}$ is a *context augmentation* if it fulfills the following two properties:

**Distinctiveness** There are no two samples $\boldsymbol{x} \in X$, $\boldsymbol{x}' \in X'$ such that $\boldsymbol{x} \approx \boldsymbol{x}'$, i.e., there is a clear distinction between the original and the context augmented distribution after applying $t_{\mathcal{C}}$. For instance, if our normal class consists of images of melanoma, flipping the image violates distinctiveness, as melanoma can be photographed from any angle. Conversely, histogram equalizing or color inversion of the image satisfies distinctiveness, as the resulting color distribution is clearly distinct from the original samples of such a dataset.

**Alignment** Let $\boldsymbol{x}, \boldsymbol{x}' \in X$, and let $\mathrm{d}(\boldsymbol{x}, \boldsymbol{x}')$ denote an appropriate similarity measure for samples in the input space. Then, we require that $\mathrm{d}(\boldsymbol{x}, \boldsymbol{x}') \approx \mathrm{d}(t_{\mathcal{C}}(\boldsymbol{x}), t_{\mathcal{C}}(\boldsymbol{x}'))$, i.e., originally similar normal samples should stay just as similar in the new context, meaning that the original and the context-augmented normal distributions should align. For instance, masking part of a torso x-ray image would violate alignment, as we could potentially remove important regions, such as the lungs, from the image altogether. On the other hand, two vertically flipped x-rays are as similar to each other as their original counterparts.

While these conditions may be dataset-dependent, some examples of context augmentation that often fulfill distinctiveness and alignment are vertical flipping (*Flip*), color inversion (*Invert*), or histogram equalization (*Equalize*). We present some examples of these augmentations in Figure 2.

### 2.2 Context Contrasting

In the following, we demonstrate how to use context augmentations to learn aligned, context-specific representation clusters with our new CON$_2$ loss. More background and preliminaries on contrastive learning can be found in Appendix B.1.

Assume a set of normal samples $X_{\text{train}}$, a context augmentation $t_{\mathcal{C}}$, a set of content-preserving augmentations $\mathcal{T}$ like in Chen et al. [2020], and let $X_{\mathcal{C}} = \{(\boldsymbol{x}, 0) \mid \boldsymbol{x} \in X_{\text{train}}\} \cup \{(t_{\mathcal{C}}(\boldsymbol{x}), 1) \mid \boldsymbol{x} \in X_{\text{train}}\}$ denote the context-augmented dataset, labeling each sample with its context.

We then apply augmentations from $\mathcal{T}$ to our dataset to create $\tilde{X}_\mathcal{C}$, where $(\tilde{\boldsymbol{x}}_{2i}, y_i), (\tilde{\boldsymbol{x}}_{2i+1}, y_i) \in \tilde{X}_\mathcal{C}$ denote two transformations of the same context-augmented sample using random augmentations from $\mathcal{T}$. More specifically, for $t, t' \in \mathcal{T}$, $\tilde{\boldsymbol{x}}_{2i} = t(\boldsymbol{x}_i^\mathcal{C})$ and $\tilde{\boldsymbol{x}}_{2i+1} = t'(\boldsymbol{x}_i^\mathcal{C})$, where $\boldsymbol{x}_i^\mathcal{C} \in X_\mathcal{C}$. Further, let $f(\tilde{X}_\mathcal{C}) := \{(f(\boldsymbol{x}), y) | \boldsymbol{x} \in \tilde{X}_\mathcal{C}\}$ for any function $f$. CON$_2$ then consists of two parts.

First, by leveraging the distinctiveness property of context augmentations, we can learn tightly concentrated, context-specific representation clusters with our *context contrasting* loss. Similar to Chen et al. [2020], let $f_\Phi(\boldsymbol{x}) = h_\phi(g_\theta(\boldsymbol{x}))$ project representations $\boldsymbol{z} = g_\theta(\boldsymbol{x})$ with a projection head $h_\phi(\boldsymbol{z})$ that gets discarded after training and $\ell$ is the instance discrimination loss as defined in Appendix B.1. We then define the *context contrasting* loss as

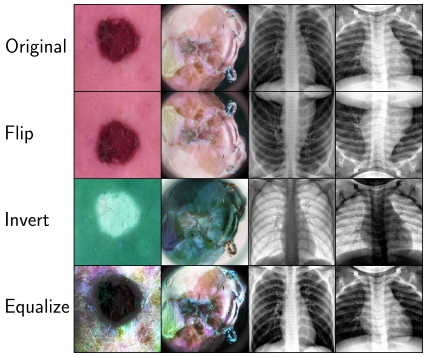



Figure 2: Examples of context augmentations for four samples from our experiments.

$$\mathcal{L}_{\text{Context}}(\tilde{X}_\mathcal{C}) = \sum_{(\tilde{\boldsymbol{x}}_i, y_i) \in \tilde{X}_\mathcal{C}} \frac{1}{2N-1} \sum_{\substack{(\tilde{\boldsymbol{x}}_j, y_j) \in \tilde{X}_\mathcal{C} \\ \tilde{\boldsymbol{x}}_j \neq \tilde{\boldsymbol{x}}_i \wedge y_i = y_j}} \ell(f_\Phi(\tilde{\boldsymbol{x}}_i), f_\Phi(\tilde{\boldsymbol{x}}_j), f_\Phi(\tilde{X}_\mathcal{C})).$$

Intuitively, context contrasting encourages representations of the same context to be clustered together while pushing other context clusters away, similar to class representations of supervised contrastive learning [Khosla et al., 2020].

While $\mathcal{L}_{\text{Context}}$ allows us to learn context-dependent representation clusters, it does not enforce a specific structure within each cluster. To make the cluster structure more informative, CON$_2$ leverages the alignment property of context augmentations to align representations across clusters through context-independent instance discrimination. More specifically, let $\Lambda(i) = \{2i, 2i+1, 4i, 4i+1\}$, i.e., $\Lambda(i)$ corresponds to all indices of samples in $\tilde{X}_\mathcal{C}$ which are augmentations of the original sample $\boldsymbol{x}_i \in X$. We then define the *content alignment* loss as

$$\mathcal{L}_{\text{Content}}(\tilde{X}_\mathcal{C}) = \sum_{k=1}^{N} \frac{1}{12} \sum_{i \in \Lambda(k)} \sum_{j \in \Lambda(k) \setminus i} \ell(f_\Psi(\tilde{\boldsymbol{x}}_i), f_\Psi(\tilde{\boldsymbol{x}}_j), f_\Psi(\tilde{X}_\mathcal{C})),$$

where $f_\Psi(\boldsymbol{x}) = h_\psi(g_\theta(\boldsymbol{x}))$, and $h_\psi$ denotes another projection head that is different from $h_\phi$. Content alignment ensures that all representations of the same normal sample can be matched across different contexts, encouraging alignment of the representations within each context cluster.

Finally, we combine context contrasting and content alignment to our loss function CON$_2$, which enables us to learn *context-specific, content-aligned* representations of normality:

$$\mathcal{L}_{\text{Con}_2}(\tilde{X}_\mathcal{C}) = \mathcal{L}_{\text{Context}}(\tilde{X}_\mathcal{C}) + \alpha \mathcal{L}_{\text{Content}}(\tilde{X}_\mathcal{C})$$

To account for the different scaling of $\mathcal{L}_{\text{Context}}$ and $\mathcal{L}_{\text{Content}}$, we need to introduce a weighting factor $\alpha \in \mathbb{R}^+$. We discuss our specific choice for $\alpha$ in Appendix E.

### 2.3 Anomaly Detection

In the AD setting, we typically assume an unlabeled training set containing predominantly normal samples, whereas we want to discriminate between normal and anomalous samples at test time [Ruff et al., 2021]. We provide some additional background on the setting in Appendix B.2.

To detect anomalies, we define two anomaly score functions that measure how well a test sample adheres to the context representation clusters. The simplest way to achieve this is a simple non-parametric nearest neighbor approach using the cosine similarity similar to [Bergman et al., 2020, Sun et al., 2022]. Specifically, we define the cosine distance between the training set $X_{\text{train}}$ and a given test sample $\boldsymbol{x}$ with transformation $t$ as

$$s_{\text{NND}}(\boldsymbol{x}; t) = - \max_{\boldsymbol{x}' \in X_{\text{train}}} \frac{\langle g_\theta(t(\boldsymbol{x})), g_\theta(t(\boldsymbol{x}')) \rangle}{\|g_\theta(t(\boldsymbol{x}))\| \|g_\theta(t(\boldsymbol{x}'))\|}$$

While this approach works well in practice, it is rather memory-inefficient, as we need to store the representations of all samples in $X_{\text{train}}$.

We address this problem by introducing a likelihood-based score function $s_{\text{LH}}$ assuming Gaussian context clusters. More specifically, let $Z^{(t)}_{train} = \left\{ \frac{g_\theta(t(\boldsymbol{x}))}{\|g_\theta(t(\boldsymbol{x}))\|} \mid \boldsymbol{x} \in X_{train} \right\}$, we then compute the density of a multivariate normal distribution based on the empirical mean $\overline{\mu}\left(Z^{(t)}_{train}\right)$ and covariance $\overline{\Sigma}\left(Z^{(t)}_{train}\right)$ given a transformation $t$. We thus define

$$s_{\text{LH}}(\boldsymbol{x}; t) = -\log\left(\mathcal{N}\left(\frac{g_\theta(t(\boldsymbol{x}))}{\|g_\theta(t(\boldsymbol{x}))\|} \,\middle|\, \overline{\mu}\left(Z^{(t)}_{train}\right), \overline{\Sigma}\left(Z^{(t)}_{train}\right)\right)\right).$$

Similar to previous works [Tack et al., 2020, Wang et al., 2023], we leverage test-time augmentations to improve the anomaly detection performance. More formally, let $\mathcal{T}_{\text{test}} = \{t_1, \ldots, t_A\}$ be a set of $A$ test time augmentations. We define our final anomaly score functions $\mathcal{S}_D : \mathcal{X} \to \mathbb{R}$ as

$$\mathcal{S}_D(\boldsymbol{x}) = \frac{1}{A}\left(\sum_{i=1}^{A/2} s_D(\boldsymbol{x}; t_i) + \sum_{i=A/2}^{A} s_D(\boldsymbol{x}; t_i \circ t_C)\right), \text{ where } D \in \{\text{NND}, \text{LH}\}.$$

## 3 Medical Anomaly Detection Experiment

We compare the performance of CON$_2$ with recent unsupervised AD methods on two challenging medical imaging datasets. We train CON$_2$ on the healthy samples of a real-world medical chest x-ray dataset [Kermany et al., 2018] and a melanoma imaging dataset [Javid, 2022], discriminating between unseen healthy and anomalous samples at test time. Here, we model invariances of normal samples with the three context augmentations *Flip*, *Invert*, and *Equalize* described in Section 2.1. Note that *Flip* violates distinctiveness on melanoma images as they could be taken from any angle. See Appendices D and E for more details on the datasets, the experimental setup, and baselines.

Table 1: Anomaly detection results on two real-world medical imaging datasets. We train each model with three different seeds and report mean ± standard deviation.

| Method | Score $\mathcal{S}$ | Pneumonia | Melanoma |
|---|---|---|---|
| SimCLR | $\mathcal{S}_{\text{NND}}$ | 91.0 ±0.9 | 72.9 ±2.8 |
| SSD | $\mathcal{S}_{\text{Mahalanobis}}$ | 90.9 ±0.2 | 79.0 ±2.2 |
| CSI | $\mathcal{S}_{\text{CSI}}$ | 73.9 ±1.6 | 92.3±0.02 |
| UniCon-HA | $\mathcal{S}_{\text{UniCon}}$ | 86.4 ±0.1 | 91.1 ±0.8 |
| CON$_2$ (Equalize) | | 93.3 ±0.6 | **93.1** ±0.04 |
| CON$_2$ (Invert) | $\mathcal{S}_{\text{LH}}$ | 90.6 ±1.0 | 91.7 ±0.2 |
| CON$_2$ (Flip) | | 91.5 ±0.6 | 80.5 ±3.0 |
| CON$_2$ (Equalize) | | **93.9** ±3.1 | 90.5 ±0.9 |
| CON$_2$ (Invert) | $\mathcal{S}_{\text{NND}}$ | 91.1 ±0.7 | 91.8 ±0.2 |
| CON$_2$ (Flip) | | 92.8 ±1.1 | 83.3 ±1.3 |

Table 1 contains the results of this experiment, including a comparison to our baselines. While most of our runs perform similarly, we indeed see that CON$_2$ with *Flip* performs drastically worse on melanoma, demonstrating that fulfilling distinctiveness and alignment is indeed crucial for context augmentations. Further, we observe that our method outperforms our baselines, confirming that modeling invariances of normal data offers an advantage over simulating anomalies for learning normal representations. We provide additional ablations on more traditional AD benchmark datasets in Appendix F.

## 4 Conclusion

In this work, we focused on anomaly detection by learning representations that capture normality. We identified that although methods based on self-supervised representation learning show promising results in this area, their reliance on prior knowledge of the structure of anomalies is a limitation. As such knowledge might not be available in real-world settings, we proposed CON$_2$ instead. Our CON$_2$ approach lets us learn representations of normal data by leveraging context augmentations. These transformations set the normal space into a new context, allowing us to observe normal data from different perspectives and thus learn context-specific representation clusters that are aligned according to the properties of the normal samples in the dataset. We demonstrated how our new representation learning method allows us to detect anomalies by introducing two anomaly scores that measure sample anomalousness by how much a representation deviates from the learned context cluster. Finally, we presented the applicability of our method in two experiments where we performed anomaly detection on real-world medical datasets. In conclusion, CON$_2$ is a reliable approach to learning highly informative representations of normality across various settings without making any assumption about anomalies, which is especially useful in safety-critical domains such as healthcare.

## References


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

J. Ansel, E. Yang, H. He, N. Gimelshein, A. Jain, M. Voznesensky, B. Bao, P. Bell, D. Berard, E. Burovski, G. Chauhan, A. Chourdia, W. Constable, A. Desmaison, Z. DeVito, E. Ellison, W. Feng, J. Gong, M. Gschwind, B. Hirsh, S. Huang, K. Kalambarkar, L. Kirsch, M. Lazos, M. Lezcano, Y. Liang, J. Liang, Y. Lu, C. K. Luk, B. Maher, Y. Pan, C. Puhrsch, M. Reso, M. Saroufim, M. Y. Siraichi, H. Suk, S. Zhang, M. Suo, P. Tillet, X. Zhao, E. Wang, K. Zhou, R. Zou, X. Wang, A. Mathews, W. Wen, G. Chanan, P. Wu, and S. Chintala. PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, volume 2 of *ASPLOS '24*, pages 929–947, New York, NY, USA, Apr. 2024. Association for Computing Machinery. ISBN 9798400703850. doi: 10.1145/3620665.3640366.

D. J. Atha and M. R. Jahanshahi. Evaluation of deep learning approaches based on convolutional neural networks for corrosion detection. *Structural Health Monitoring*, 17(5):1110–1128, Sept. 2018. ISSN 1475-9217. doi: 10.1177/1475921717737051.

L. Bergman and Y. Hoshen. Classification-Based Anomaly Detection for General Data. In *International Conference on Learning Representations*, 2019.

L. Bergman, N. Cohen, and Y. Hoshen. Deep nearest neighbor anomaly detection. *arXiv preprint arXiv:2002.10445*, 2020.

C. M. Bishop. Novelty detection and neural network validation. *IEE Proceedings - Vision, Image and Signal Processing*, 141(4):217–222, Aug. 1994. ISSN 1359-7108. doi: 10.1049/ip-vis:19941330.

M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. LOF: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, SIGMOD '00, pages 93–104, New York, NY, USA, May 2000. Association for Computing Machinery. ISBN 978-1-58113-217-5. doi: 10.1145/342009.335388.

J. Chen, S. Sathe, C. Aggarwal, and D. Turaga. Outlier Detection with Autoencoder Ensembles. In *Proceedings of the 2017 SIAM International Conference on Data Mining (SDM)*, Proceedings, pages 90–98. Society for Industrial and Applied Mathematics, June 2017.

T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A Simple Framework for Contrastive Learning of Visual Representations. *37th International Conference on Machine Learning, ICML 2020*, PartF168147-3:1575–1585, Feb. 2020. doi: 10.48550/arxiv.2002.05709.

M. J. Cohen and S. Avidan. Transformaly-two (feature spaces) are better than one. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4060–4069, 2022.

S. Cortinhas. Muffin vs. Chihuahua, 2023.

W. Cukierski. Dogs vs. Cats, 2013.

I. Daunhawer, A. Bizeul, E. Palumbo, A. Marx, and J. E. Vogt. Identifiability results for multimodal contrastive learning. In *The Eleventh International Conference on Learning Representations*, 2023.

W. Falcon and The PyTorch Lightning team. PyTorch Lightning, Mar. 2019.

I. Golan and R. El-Yaniv. Deep Anomaly Detection Using Geometric Transformations. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

K. Golmohammadi and O. R. Zaiane. Time series contextual anomaly detection for detecting market manipulation in stock market. In *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–10, Oct. 2015. doi: 10.1109/DSAA.2015.7344856.

C. R. Harris, K. J. Millman, S. J. v. d. Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. v. Kerkwijk, M. Brett, A. Haldane, J. F. d. Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant. Array programming with NumPy. *Nature*, 585(7825): 357–362, Sept. 2020. doi: 10.1038/s41586-020-2649-2.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

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.

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

D. Hendrycks, M. Mazeika, S. Kadavath, and D. Song. Using self-supervised learning can improve model robustness and uncertainty. *Advances in neural information processing systems*, 32, 2019b.

M. H. Javid. Melanoma skin cancer dataset of 10000 images, 2022.

D. S. Kermany, M. Goldbaum, W. Cai, C. C. S. Valentim, H. Liang, S. L. Baxter, A. McKeown, G. Yang, X. Wu, F. Yan, J. Dong, M. K. Prasadha, J. Pei, M. Y. L. Ting, J. Zhu, C. Li, S. Hewett, J. Dong, I. Ziyar, A. Shi, R. Zhang, L. Zheng, R. Hou, W. Shi, X. Fu, Y. Duan, V. A. N. Huu, C. Wen, E. D. Zhang, C. L. Zhang, O. Li, X. Wang, M. A. Singer, X. Sun, J. Xu, A. Tafreshi, M. A. Lewis, H. Xia, and K. Zhang. Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning. *Cell*, 172(5):1122–1131.e9, Feb. 2018. ISSN 0092-8674, 1097-4172. doi: 10.1016/j.cell.2018.02.010.

P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

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

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.

F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation Forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422, Dec. 2008. doi: 10.1109/ICDM.2008.17.

P. Liznerski, L. Ruff, R. A. Vandermeulen, B. J. Franks, K. R. Muller, and M. Kloft. Exposing Outlier Exposure: What Can Be Learned From Few, One, and Zero Outlier Images. *Transactions on Machine Learning Research*, Aug. 2022. ISSN 2835-8856.

I. Loshchilov and F. Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017.

I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

W. McKinney. Data Structures for Statistical Computing in Python. In S. v. d. Walt and J. Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 56 – 61, 2010. doi: 10.25080/Majora-92bf1922-00a.

H. Mirzaei, M. Salehi, S. Shahabi, E. Gavves, C. G. Snoek, M. Sabokrou, and M. H. Rohban. Fake It Until You Make It: Towards Accurate Near-Distribution Novelty Detection. In *The Eleventh International Conference on Learning Representations*, 2022.

B. Nachman and D. Shih. Anomaly detection with density estimation. *Physical Review D*, 101(7): 075042, Apr. 2020. doi: 10.1103/PhysRevD.101.075042.

E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan. Do Deep Generative Models Know What They Don't Know? *7th International Conference on Learning Representations, ICLR 2019*, Oct. 2018. doi: 10.48550/arxiv.1810.09136.

P. Oza and V. M. Patel. One-class convolutional neural network. *IEEE Signal Processing Letters*, 26 (2):277–281, 2018.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, and others. Scikit-learn: Machine learning in Python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

C. Qiu, A. Li, M. Kloft, M. Rudolph, and S. Mandt. Latent outlier exposure for anomaly detection with contaminated data. In *International Conference on Machine Learning*, pages 18153–18167. PMLR, 2022.

T. Reiss and Y. Hoshen. Mean-Shifted Contrastive Loss for Anomaly Detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(2):2155–2162, June 2023. ISSN 2374-3468. doi: 10.1609/aaai.v37i2.25309.

L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft. Deep one-class classification. In *International conference on machine learning*, pages 4393–4402. PMLR, 2018.

L. Ruff, R. A. Vandermeulen, N. Görnitz, A. Binder, E. Müller, K.-R. Müller, and M. Kloft. Deep semi-supervised anomaly detection. In *International Conference on Learning Representations*, 2020.

L. Ruff, J. R. Kauffmann, R. A. Vandermeulen, G. Montavon, W. Samek, M. Kloft, T. G. Dietterich, and K.-R. Müller. A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE*, 109(5):756–795, 2021.

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, Dec. 2015. ISSN 0920-5691, 1573-1405. doi: 10.1007/s11263-015-0816-y.

A. Ryser, L. Manduchi, F. Laumer, H. Michel, S. Wellmann, and J. E. Vogt. Anomaly Detection in Echocardiograms with Dynamic Variational Trajectory Models. In *Proceedings of the 7th Machine Learning for Healthcare Conference*, pages 425–458. PMLR, Dec. 2022.

M. Sabokrou, M. Fathy, G. Zhao, and E. Adeli. Deep end-to-end one-class classifier. *IEEE transactions on neural networks and learning systems*, 32(2):675–684, 2020.

T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs. Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery. In M. Niethammer, M. Styner, S. Aylward, H. Zhu, I. Oguz, P.-T. Yap, and D. Shen, editors, *Information Processing in Medical Imaging*, pages 146–157, Cham, 2017. Springer International Publishing. ISBN 978-3-319-59050-9. doi: 10.1007/978-3-319-59050-9_12.

T. Schlegl, P. Seeböck, S. M. Waldstein, G. Langs, and U. Schmidt-Erfurth. f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks. *Medical Image Analysis*, 54:30–44, May 2019. ISSN 1361-8415. doi: 10.1016/j.media.2019.01.010.

B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.

V. Sehwag, M. Chiang, and P. Mittal. SSD: A unified framework for self-supervised outlier detection. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

K. Sohn. Improved Deep Metric Learning with Multi-class N-pair Loss Objective. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Y. Sun, Y. Ming, X. Zhu, and Y. Li. Out-of-Distribution Detection with Deep Nearest Neighbors. In *Proceedings of the 39th International Conference on Machine Learning*, pages 20827–20840. PMLR, June 2022.

J. Tack, S. Mo, J. Jeong, and J. Shin. CSI: Novelty Detection via Contrastive Learning on Distributionally Shifted Instances. In *Advances in Neural Information Processing Systems*, volume 33, pages 11839–11852. Curran Associates, Inc., 2020.

D. M. Tax and R. P. Duin. Support Vector Data Description. *Machine Learning*, 54(1):45–66, Jan. 2004. ISSN 1573-0565. doi: 10.1023/B:MACH.0000008084.60811.49.

T. p. d. team. pandas-dev/pandas: Pandas, Feb. 2020.

A. van den Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding, 2019.

P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, I. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17: 261–272, 2020. doi: 10.1038/s41592-019-0686-2.

J. von Kügelgen, Y. Sharma, L. Gresele, W. Brendel, B. Schölkopf, M. Besserve, and F. Locatello. Self-Supervised Learning with Data Augmentations Provably Isolates Content from Style. In *Advances in Neural Information Processing Systems*, volume 34, pages 16451–16467. Curran Associates, Inc., 2021.

G. Wang, Y. Wang, J. Qin, D. Zhang, X. Bao, and D. Huang. Unilaterally aggregated contrastive learning with hierarchical augmentation for anomaly detection. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 6865–6874. IEEE, 2023. doi: 10.1109/ICCV51070.2023.00634.

H. Wang, Z. Li, L. Feng, and W. Zhang. Vim: Out-of-distribution with virtual-logit matching. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 4911–4920. IEEE, 2022. doi: 10.1109/CVPR52688.2022.00487.

Z. Wu, Y. Xiong, S. X. Yu, and D. Lin. Unsupervised feature learning via non-parametric instance discrimination. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 3733–3742. Computer Vision Foundation / IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00393.

Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, and C. Wang. Machine Learning and Deep Learning Methods for Cybersecurity. *IEEE Access*, 6:35365–35381, 2018. ISSN 2169-3536. doi: 10.1109/ACCESS.2018.2836950.

S. You, K. C. Tezcan, X. Chen, and E. Konukoglu. Unsupervised Lesion Detection via Image Restoration with a Normative Prior. In *Proceedings of The 2nd International Conference on Medical Imaging with Deep Learning*, pages 540–556. PMLR, May 2019.

R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, and R. X. Gao. Deep learning and its applications to machine health monitoring. *Mechanical Systems and Signal Processing*, 115:213–237, Jan. 2019. ISSN 0888-3270. doi: 10.1016/j.ymssp.2018.05.050.

B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

# A Related Work

Recently, learning useful normal representations of high-dimensional data to perform anomaly detection has become a popular line of research. Prior work has tackled the problem from various angles, for instance, using hypersphere compression [Ruff et al., 2018]. Other popular methods define pretext tasks such as learning reconstruction models [Chen et al., 2017, Zong et al., 2018, You et al., 2019] or predicting data transformations [Golan and El-Yaniv, 2018, Hendrycks et al., 2019b, Bergman and Hoshen, 2019]. While these approaches had some success in the past, the learned representations are not very informative. On the other hand, methods learning more informative representations have recently been shown to improve over prior work [Sun et al., 2022, Sehwag et al., 2021].

Another line of work focused on estimating the training density with the help of generative models, detecting anomalies as samples from low probability regions [An and Cho, 2015, Schlegl et al., 2019, Nachman and Shih, 2020, Mirzaei et al., 2022]. However, these methods tend to generalize better to unseen distributions than to the observed training distribution [Nalisnick et al., 2018].

In addition to the traditional setting, where we assume training data without any labels, some works have slightly weakened this restriction and assumed access to a limited number of labeled samples. This setting is called anomaly detection with Outlier Exposure (OE) [Hendrycks et al., 2019a], and it has been shown that even just a few labeled samples can greatly boost performance over an unlabeled dataset [Ruff et al., 2020, Qiu et al., 2022, Liznerski et al., 2022]. Using large, pretrained models as feature extractors is a special case of OE, as additional data is not explicitly accessible. Some approaches have been introduced that use representations from pretrained models directly in zero-shot fashion[Bergman et al., 2020, Liznerski et al., 2022], while others demonstrate the benefit of fine-tuning [Cohen and Avidan, 2022, Reiss and Hoshen, 2023]. OE has been very successful in the past, often outperforming traditional AD settings across many benchmarks, though at the cost of either requiring labeled samples or vast amounts of data for pretraining, which are both often not available or hard to obtain in more specialized domains.

Another setting that has recently gained popularity is out-of-distribution (OOD) detection. In OOD detection, we have additional information about our dataset in the form of labels. Anomaly detection is thus a special case of OOD detection with only a single label. While the problem is similar, most approaches that tackle OOD detection make specific use of a classifier trained on the dataset labels [Hendrycks and Gimpel, 2017, Lee et al., 2018, Wang et al., 2022], which AD.

In comparison, our method operates in the traditional anomaly detection setting and can be applied to datasets without any knowledge about anomalies. Further, while we do assume access to a dataset containing only normal samples, our method does not rely on any additional labels, as they are potentially difficult and expensive to obtain, particularly in more specialized settings.

# B Background

In this section, we provide some terminology for contrastive learning and background about the anomaly detection setting.

## B.1 Contrastive Learning

Recently, contrastive learning has emerged as a popular approach for representation learning [van den Oord et al., 2019, Chen et al., 2020]. By design, contrastive learning has the capability to learn representations that are agnostic to certain invariances [von Kügelgen et al., 2021, Daunhawer et al., 2023], which makes contrastive learning a particularly interesting choice to learn informative representations of normal samples [Tack et al., 2020, Wang et al., 2023], as it allows us to incorporate prior knowledge about our data into the representing learning process in the form of data augmentations. More specifically, invariances are learned by forming positive and negative pairs over the training dataset by applying data augmentations that should retain the relevant content of a sample.

The goal of contrastive learning is to learn an encoding function $g_\theta(\boldsymbol{x})$, where representations of positive pairs of samples are close and negative pairs are far from each other. For a given pair of samples $\boldsymbol{x}, \boldsymbol{x}' \in X$, we can define the instance discrimination loss as [Sohn, 2016, Wu et al., 2018,

van den Oord et al., 2019]

$$\ell(\boldsymbol{x}, \boldsymbol{x}', X) = -\log \frac{\exp\left(\text{sim}(\boldsymbol{x}, \boldsymbol{x}')/\tau\right)}{\sum\limits_{\boldsymbol{x}'' \in X:\, \boldsymbol{x}'' \neq \boldsymbol{x}} \exp\left(\text{sim}(\boldsymbol{x}, \boldsymbol{x}'')/\tau\right)} \,.$$

Here, the function $\text{sim}(\boldsymbol{x}, \boldsymbol{x}')$ corresponds to a function that measures the similarity between $\boldsymbol{x}$ and $\boldsymbol{x}'$. For the rest of our work, we assume $\text{sim}(\boldsymbol{x}, \boldsymbol{x}')$ to be the cosine similarity between the two input vectors, as this is one of the most popular choices in the contrastive learning literature.

One of the most prominent contrastive methods is SimCLR [Chen et al., 2020], which creates positive pairs through sample augmentations. There exists a supervised extension called SupCon [Khosla et al., 2020], which incorporates class labels into the SimCLR loss. For a given set of augmentations $T$, a dataset $X = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$, and an augmented dataset $\tilde{X}$ where $|\tilde{X}| = 2N$ and $(\tilde{\boldsymbol{x}}_{2i}, y_i), (\tilde{\boldsymbol{x}}_{2i+1}, y_i) \in \tilde{X}$ denote two transformations of the same sample using random augmentations from $T$, SimCLR and SupCon introduce the following loss functions:

$$\mathcal{L}_{\text{SimCLR}}(\tilde{X}) = \frac{1}{2N} \sum_{i=1}^{N} \left(\ell(f_\Theta(\tilde{\boldsymbol{x}}_{2i}), f_\Theta(\tilde{\boldsymbol{x}}_{2i+1}), f_\Theta(\tilde{X})) + \ell(f_\Theta(\tilde{\boldsymbol{x}}_{2i+1}), f_\Theta(\tilde{\boldsymbol{x}}_{2i}), f_\Theta(\tilde{X}))\right) ,$$

$$\mathcal{L}_{\text{SupCon}}(\tilde{X}) = \sum_{(\tilde{\boldsymbol{x}}_i, y_i) \in \tilde{X}} \frac{1}{N(y_i) - 1} \sum_{\substack{(\tilde{\boldsymbol{x}}_j, y_j) \in \tilde{X}: \\ \tilde{\boldsymbol{x}}_j \neq \tilde{\boldsymbol{x}}_i \wedge y_i = y_j}} \ell(f_\Theta(\tilde{\boldsymbol{x}}_i), f_\Theta(\tilde{\boldsymbol{x}}_j), f_\Theta(\tilde{X})) \,.$$

Here, $\Theta = \{\theta, \theta'\}$ and $N(y) = \left|\{(\tilde{\boldsymbol{x}}_i, y_i) \mid (\tilde{\boldsymbol{x}}_i, y_i) \in \tilde{X} \wedge y_i = y\}\right|$ denotes the number of samples in $\tilde{X}$ with label $y$. We further denote $f_\Theta(\tilde{X}) = \{f_\Theta(\tilde{\boldsymbol{x}}) \mid (\tilde{\boldsymbol{x}}, y) \in \tilde{X}\}$ and $f_\Theta(\boldsymbol{x}) = h_{\theta'}(g_\theta(\boldsymbol{x}))$, where $\boldsymbol{z} = g_\theta(\boldsymbol{x})$ is a feature extractor and $h_{\theta'}(\boldsymbol{z})$ is a projection head that is typically only used during training [Chen et al., 2020].

### B.2 Anomaly Detection

In the anomaly detection setting, we are given an unlabeled dataset $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} = X \subset \mathcal{X}$, while assuming that most samples are normal, i.e., the dataset is practically free of outliers [Ruff et al., 2021]. The goal is to learn a model from the given dataset that discriminates between normal and anomalous data at test time.

In this work, we assume the challenging case where our dataset is completely free of anomalies. Hence, we aim to discriminate between the normal class and a completely unobserved set of anomalies at test time. This setting is sometimes called one-class classification or novelty detection.

To achieve this goal, one straightforward approach is to approximate the distribution $p_{\mathcal{X}}(\boldsymbol{x})$ directly using generative models [An and Cho, 2015, Schlegl et al., 2019]. Because we assume normal data to lie in high-density regions of $p_{\mathcal{X}}$, we can discriminate between normal and anomalous samples by applying a threshold function $p_{\mathcal{X}}(\boldsymbol{x}) \leq \tau$, where $\tau \in \mathbb{R}$ is an often task-specific threshold [Bishop, 1994]. As density-based approaches are often difficult to apply to high-dimensional data directly [Nalisnick et al., 2018], we follow a slightly different line of work.

In this paper, we focus on learning a function $g_\theta : \mathcal{X} \to \mathcal{Z}$ that provides us with representations that capture the normal attributes of samples in the dataset [Sehwag et al., 2021, Tack et al., 2020, Wang et al., 2023], by mapping normal samples close to each other in representation space. On the other hand, anomalies that lack the learned normal structure should be mapped to a different part of the representation space.

Given $g_\theta(\boldsymbol{x})$, a popular approach to detect anomalies is by defining a scoring function $\mathcal{S} : \mathcal{Z} \to \mathbb{R}$ [Breunig et al., 2000, Schölkopf et al., 2001, Tax and Duin, 2004, Liu et al., 2008]. The score function maps a representation onto a metric that estimates the anomalousness of a sample. To identify anomalies at test time, we can use $\mathcal{S}$ similarly to the density $p_{\mathcal{X}}$, i.e., we consider a new sample $\boldsymbol{x}$ to be normal if $\mathcal{S}(g_\theta(\boldsymbol{x})) \leq \tau$, whereas $\mathcal{S}(g_\theta(\boldsymbol{x})) > \tau$ means $\boldsymbol{x}$ is an anomaly.

## C Compute & Code

We run all our experiments on single GPUs on a compute cluster using a combination of RTX2080Ti, RTX3090, and RTX4090 GPUs. Each experiment can be run with 4 CPU workers and 16 GB of

Table 2: Approximate compute hours for the experiments for each dataset and method. SimCLR and SSD use the same representations, so we can evaluate both methods in one go and list their compute hours together.

| Method \ Dataset | CIFAR10 | CIFAR100 | ImageNet30 | Dogs vs. Cats | Muffin vs. Chihuahua | Pneumonia | Melanoma |
|---|---|---|---|---|---|---|---|
| SimCLR/SSD | 35 | 120 | 315 | 60 | 21 | 12 | 15 |
| CSI | - | - | - | 81 | 27 | 24 | 19 |
| UniCon-HA | - | - | - | 240 | 108 | 36 | 54 |
| CON$_2$ | 465 | 135 | 360 | 78 | 40 | 58.5 | 63 |

memory. We provide an overview of the compute for our experiments in Table 2. Our experiments are written using PyTorch [Ansel et al., 2024] with Lightning [Falcon and The PyTorch Lightning team, 2019].

In the following, we list for each of our methods and baselines how we arrive at results and which code we use.

**CON$_2$**: We implement CON$_2$ using PyTorch [Ansel et al., 2024] together with Lightning [Falcon and The PyTorch Lightning team, 2019]. To evaluate our method, we use various open-source Python libraries such as NumPy [Harris et al., 2020], scikit-learn [Pedregosa et al., 2011], Pandas [McKinney, 2010, team, 2020], or SciPy [Virtanen et al., 2020]. Implementation of the CON$_2$ objective is partially based on code provided by Khosla et al. [2020] (https://github.com/HobbitLong/SupContrast).

**SimCLR**: For this baseline, we implement SimCLR [Chen et al., 2020] and compute anomaly scores in a similar fashion as [Sun et al., 2022]. For this baseline, we rely on similar packages as CON$_2$.

**SSD**: We take results for SSD on CIFAR10 from Sehwag et al. [2021]. For the other experiments, we implement the baseline following the paper. Our implementation follows a similar structure as SimCLR.

**CSI**: We CSI results on CIFAR10, CIFAR100, and ImageNet30 from Tack et al. [2020]. For all other experiments, we download the code from https://github.com/alinlab/CSI and run it with new dataloaders.

**UniCon-HA**: Similar to CSI, we take results on CIFAR10, CIFAR100, and ImageNet30 from Wang et al. [2023]. For all other experiments, the authors shared their code with us, such that we could run the experiments for the other datasets by using the original code with new dataloaders.

# D Datasets

In the following, we provide details about preprocessing, sources, and licenses of the datasets we use in our experiments.

### Pneumonia

Our Pneumonia dataset was originally published by Kermany et al. [2018] and consists of $5'863$ lung xrays, which are labeled with *Pneumonia* and *Normal* labels. We first resize images to $256$ and apply center-cropping to feed $224 \times 224$ images to our model. We ran all our experiments on the Pneumonia dataset with a batch size of $128$. The dataset can be downloaded from https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia and is published under *CC BY 4.0* license.

### Melanoma

We use the Melanoma dataset of Javid [2022], which consists of $10'600$ images of Melanoma labeled with being *benign* or *malignant*. We resize each image to size $128 \times 128$ before passing them to the model with batch size $128$. The dataset is publicly available at https://www.kaggle.com/datasets/hasnainjaved/melanoma-skin-cancer-dataset-of-10000-images and is published under the *CC0: Public Domain* license.

### CIFAR10/CIFAR100

CIFAR10 and CIFAR100 are natural image datasets with $32 \times 32$ samples. Both datasets consist of a total of $60'000$ samples, with a total of $10$ and $100$ samples for CIFAR10 and CIFAR100, respectively. As CIFAR100 comes with only $600$ samples per class, the dataset authors additionally define a set of $20$ superclasses, aggregating $5$ labels each. In our one-class classification experiments on CIFAR100 we use the superclasses to ensure a manageable number of runs and a sufficient amount of training data. We ran all our experiments on CIFAR10 and CIFAR100 with a batch size of $512$. Both datasets were published by Krizhevsky et al. [2009] and can be downloaded from https://www.cs.toronto.edu/~kriz/cifar.html. To the best of our knowledge, these datasets come without a license.

### Imagenet30

The ImageNet30 dataset is a subset of the original ImageNet dataset [Russakovsky et al., 2015]. It was created by Hendrycks et al. [2019b] for the purpose of one-class classification. The dataset consists of $42'000$ natural images where each is labeled with one of $30$ classes. We preprocess the dataset by resizing the shorter edge to $256$ pixels, from which we randomly crop a $224 \times 224$ image patch every time we load an image for training. We ran all our experiments on ImageNet with a batch size of $128$. The dataset can be downloaded from https://github.com/hendrycks/ss-ood, which comes with the MIT License. Further, while we could not find a license for ImageNet, terms of use are provided on https://image-net.org/.

### Dogs vs. Cats

The Dogs vs. Cats was originally introduced in a Kaggle challenge by Microsoft Research [Cukierski, 2013] and consists of $25'000$ images of cats and dogs. We preprocess the dataset by resizing the shorter edge to $128$ pixels and then perform center cropping, feeding the resulting $128 \times 128$ image to our model. We ran all our experiments on Dogs vs. Cats with a batch size of $256$. The dataset can be downloaded from https://www.kaggle.com/competitions/dogs-vs-cats/data. To the best of our knowledge, there is no official license for the dataset, but the Kaggle page points to the Kaggle Competition rules https://www.kaggle.com/competitions/dogs-vs-cats/rules in the license section.

### Chihuahua vs. Muffin

The Chihuahua vs. Muffin dataset consists of $6'000$ images scraped from Google Images. We preprocess the dataset similar to ImageNet30, resizing the shorter edge of the images to $128$ pixels while feeding random $128 \times 128$ sized image crops to the model during training. We ran all our experiments on Chihuahua vs. Muffin with a batch size of $256$. The dataset was published by Cortinhas [2023] and can be downloaded from https://www.kaggle.com/datasets/samuelcortinhas/muffin-vs-chihuahua-image-classification/data. According to the datasets Kaggle page, the dataset is licensed under *CC0: Public Domain*.

In addition to the preprocessing mentioned above, we normalize each image with a mean and standard deviation of $0.5$ after applying the augmentations of $\text{CON}_2$.

## E Experimental Details

We evaluate our method in the so-called one-class classification setting [Ruff et al., 2021]. More specifically, during training we assume to have access to only the normal (healthy) class. At test time, the goal is to detect whether a new sample stems from the normal class seen during training or whether it seems anomalous, i.e., deviates from the training distribution.

Typically, there is a high-class imbalance between normal and anomalous samples in the one-class classification setting. Further, setting an appropriate threshold for the anomaly score is often task-dependent. Therefore, a popular approach to evaluating the performance of anomaly detection methods is to use the area under the receiver operator characteristic curve (AUROC) [Ruff et al., 2021]. This metric is threshold agnostic and robust to class imbalance.

We compare our work to a number of contrastive anomaly detection baselines, such as SSD [Sehwag et al., 2021], CSI [Tack et al., 2020], and UniCon-HA [Wang et al., 2023]. We further compare against a baseline that learns SimCLR embeddings and detects samples in nearest neighbor fashion similar to KNN+ [Sun et al., 2022], which was originally developed for out-of-distribution detection. To ensure comparability, we run all experiments with the same ResNet18 architecture [He et al., 2016].

Similar to our method, all baselines make use of test-time augmentations. By default, both CSI and UniCon-HA use $40$ test time augmentations, which we adopt for all baselines. In our experiments, we set the augmentation class $\mathcal{T}$ to the set of augmentations introduced by Chen et al. [2020]. For the context augmentation, we experiment with vertical flips (Flip), inverting the pixels of an image (Invert), i.e., $t_{\text{Invert}}(\boldsymbol{x}_{ij}) = 1 - \boldsymbol{x}_{ij}$, and histogram equalization (Equalize), see Figure 2 for an illustration.

We choose hyperparameters for $\text{CON}_2$ based on their performance on the CIFAR10 dataset and keep them constant across all experiments. We linearly anneal the hyperparameter $\alpha$ in $\mathcal{L}_{\text{CON}_2}$ from $0$ to $1$ over the course of training to encourage the model to first learn the context-specific cluster structure while gradually aligning representations over the course of training. We optimize our loss using the AdamW optimizer [Loshchilov and Hutter, 2019] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay $\lambda = 0.001$, and using a learning rate of $10^{-3}$ with a cosine annealing [Loshchilov and Hutter, 2017] schedule. We run all experiments for $2048$ epochs.

For all our experiments, we report mean and standard deviation over three seeds per class of the dataset. Note that the average results of a dataset are aggregated over different one-class classification settings, one per class of the dataset.

# F Ablations

In this section, we provide some additional experiments, illustrating the structure of the learned representations (Appendix F.1), additional experiments on natural images (Appendix F.2), and experiments going beyond only two context clusters (Appendix F.3).

## F.1 Context Clusters

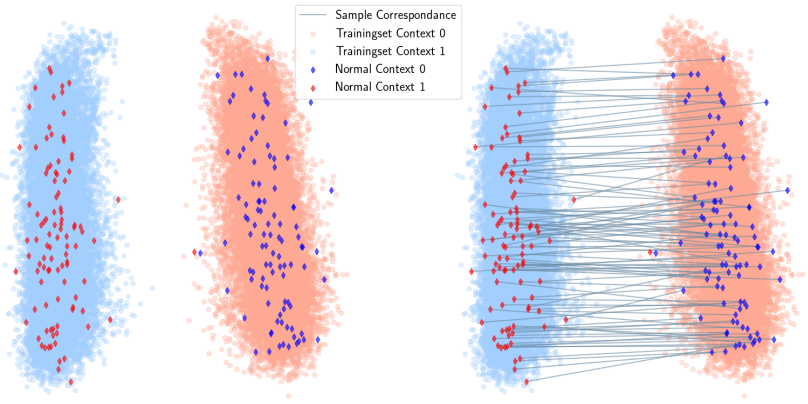
We illustrate the structure of representations learned by $\text{CON}_2$ in Figure 3, demonstrating how our intuition from Section 2 translates to what our model learns. More specifically, Figure 3 presents the PCA embeddings of train, normal test, and anomalous test samples when training $\text{CON}_2$ on the *car* class of CIFAR10.

We see that the normal samples cluster nicely according to their context for both train and unobserved normal test data. Further, we also observe that normal samples align well across contexts, as their relative positions within their respective cluster appear consistent from the parallel lines that mark correspondence. Conversely, anomalous data often fails to adhere to the context clustering structure or align well across contexts.

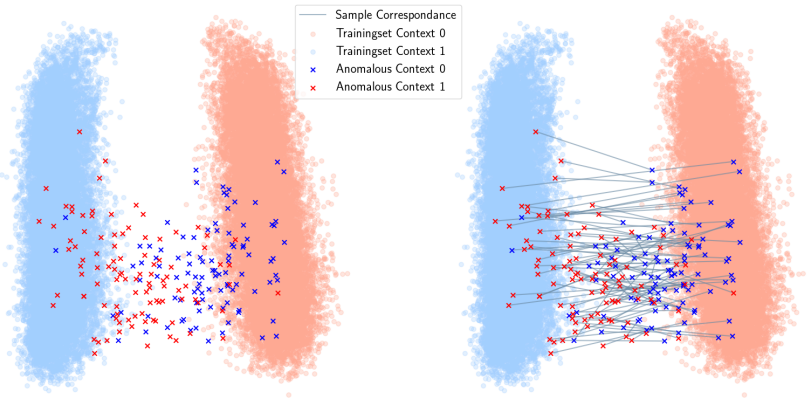
## F.2 Natural Image Benchmarks

In addition to the experiments on the medical datasets in Section 3, we also train our method in the common one-class classification setting [Ruff et al., 2021] on different natural imaging datasets on one-class CIFAR10/CIFAR100 [Krizhevsky et al., 2009], ImageNet30 [Russakovsky et al., 2015, Hendrycks et al., 2019b], Dogs vs. Cats [Cukierski, 2013], and Muffin vs. Chihuahua [Cortinhas, 2023], and compare to the baselines described in Appendix E. We present the results of this comparison in Table 3.

First, we note an interesting discrepancy between the Invert and Flip context augmentations and Equalize. On average, Equalize seems to perform quite a bit worse than the other two context augmentations. We suspect that this comes from the fact that Equalize does not always properly fulfill the distinctiveness assumption of context augmentations, as equalized samples are visually quite similar to the original sample for natural images (see Figure 2). Therefore, equalized samples could easily appear as part of the training set, which would violate distinctiveness. In contrast, Flip and Invert satisfy distinctiveness and alignment on these datasets and consequently perform relatively




(a) Alignment of normal test samples.




(b) Alignment of anomalous test samples.

Figure 3: Two dimensional PCA embedding of the train, normal test (a) and anomalous test samples (b). Lines connecting representations mark embeddings corresponding to the same sample in different contexts. Parallel lines indicate that sample representations are positioned approximately at the same location across context clusters, i.e., are aligned across contexts.

Table 3: One class classification results for CIFAR100, ImageNet30, Dogs vs. Cats, and Muffin vs. Chihuahua. Results with a * are taken from the original paper. For each dataset, we train models over three different seeds per dataset class. We report mean and standard deviation over all one-class settings of each dataset. We bold the **best** and underline the second best results.

| Method | Score | CIFAR10 | CIFAR100 | ImageNet30 | Dogs vs. Cats | Muffin vs. Chihuahua |
|---|---|---|---|---|---|---|
| SimCLR | $\mathcal{S}_{\text{NND}}$ | 89.2±6.7 | 81.5±8.6 | 74.7±12.2 | 84.7±2.2 | 78.6±11.4 |
| SSD | $\mathcal{S}_{\text{Mahalanobis}}$ | 97.4 ±8.1 | 79.1±9.5 | 76.8±13.0 | 84.5±0.6 | 75.0±14.0 |
| CSI | $\mathcal{S}_{\text{CSI}}$ | 94.3* | 89.6* | 91.6* | **90.3**±0.4 | **95.1**±2.4 |
| UniCon-HA | $\mathcal{S}_{\text{UniCon}}$ | **95.4*** | **92.4*** | **93.2*** | 67.9±6.2 | 91.9±1.3 |
| CON$_2$ (Equalize) | $\mathcal{S}_{\text{LH}}$ | 91.0±5.4 | 86.1±5.5 | 85.2±12.6 | 77.0±1.1 | 83.0±12.2 |
| | $\mathcal{S}_{\text{NND}}$ | 91.5±5.0 | 87.5±4.4 | 86.0±12.0 | 81.2±1.9 | 87.5±8.0 |
| CON$_2$ (Invert) | $\mathcal{S}_{\text{LH}}$ | 93.0±4.8 | 89.5±5.4 | 90.9± 8.8 | 87.8±1.0 | 91.4±4.2 |
| | $\mathcal{S}_{\text{NND}}$ | 93.9±4.2 | 90.6±4.9 | 91.2± 8.4 | 88.7±1.5 | 93.8±3.0 |
| CON$_2$ (Flip) | $\mathcal{S}_{\text{LH}}$ | 94.0±4.1 | 89.1±4.6 | 88.9±11.9 | 90.0±1.1 | 92.6±2.9 |
| | $\mathcal{S}_{\text{NND}}$ | 94.6±3.7 | 89.7±4.2 | 89.8±11.1 | **90.3**±1.7 | 94.0±1.7 |

well across all datasets. Our method also compares well against established baselines on natural images, consistently displaying the best or second-best results among all baselines.

From the relatively low standard deviations, we can further see that we are consistently achieving high AUROCs across all one-class settings within each dataset. Apart from results with the Equalize

context augmentation, the highest variability across one-class settings appears in Imagenet30 with the Flip context augmentation. A closer look at individual performances in Table 6 reveals that this is mainly due to two one-class settings for which our method seems to produce slightly worse results. More specifically, the normal classes "nail" and "pillow" perform very poorly with average AUROCs of 51.8 and 67.3, respectively. We suspect the poor performance is due to using the Flip context augmentation, which violates the distinctiveness assumption for nails and pillows, as these objects could be recorded from any arbitrary angle. However, apart from these outliers, we perform very well on ImageNet30, with a median AUROC of 93.4. For the Flip Context Augmentation, we provide a detailed overview of all the individual one-class classification results across all datasets in Tables 4 to 8.

Table 4: AUROCs of the experiments on one-class CIFAR10. We compare CON$_2$ with different context augmentations to pretext and contrastive AD methods. Both the Invert and Flip context augmentations fulfill our assumptions from Section 2.1, whereas samples in the Equalize context are sometimes similar to the sample in the original context, violating distinctiveness and thus resulting in slightly lower performance. For methods with a *, we adopt results from the original paper.

| Method | Score $\mathcal{S}$ | Plane | Car | Bird | Cat | Deer | Dog | Frog | Horse | Ship | Truck | Mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| SimCLR | $\mathcal{S}_{\text{NND}}$ | 78.6$_{\pm 0.6}$ | 98.9$_{\pm 0.1}$ | 87.1$_{\pm 0.6}$ | 84.9$_{\pm 0.3}$ | 81.0$_{\pm 1.4}$ | 92.3$_{\pm 0.3}$ | 94.8$_{\pm 0.5}$ | 94.7$_{\pm 0.1}$ | 84.7$_{\pm 1.7}$ | 95.3$_{\pm 0.6}$ | 89.2$_{\pm 6.7}$ |
| SSD* | $\mathcal{S}_{\text{Mahalanobis}}$ | 82.7 | 98.5 | 84.2 | 84.5 | 84.8 | 90.9 | 91.7 | 95.2 | 92.9 | 94.4 | 90.0 |
| CSI* | $\mathcal{S}_{\text{CSI}}$ | 89.9$_{\pm 0.1}$ | 99.1$_{\pm 0.0}$ | 93.1$_{\pm 0.2}$ | 86.4$_{\pm 0.2}$ | 93.9$_{\pm 0.1}$ | 93.2$_{\pm 0.2}$ | 95.1$_{\pm 0.1}$ | 98.7$_{\pm 0.0}$ | 97.9$_{\pm 0.0}$ | 95.5$_{\pm 0.1}$ | 94.3 |
| UniCon-HA* | $\mathcal{S}_{\text{UniCon}}$ | **91.7**$_{\pm 0.1}$ | 99.2 $_{\pm 0.0}$ | **93.9**$_{\pm 0.1}$ | **89.5**$_{\pm 0.2}$ | **95.1**$_{\pm 0.1}$ | **94.1** $_{\pm 0.2}$ | 96.6$_{\pm 0.1}$ | **98.9**$_{\pm 0.0}$ | **98.1**$_{\pm 0.0}$ | 96.6$_{\pm 0.1}$ | **95.4** |
| CON$_2$ (Equalize) | | 89.3$_{\pm 1.0}$ | 98.4$_{\pm 0.2}$ | 85.6$_{\pm 0.1}$ | 77.4$_{\pm 2.1}$ | 90.2$_{\pm 0.3}$ | 87.9$_{\pm 1.3}$ | 95.9$_{\pm 0.2}$ | 94.8$_{\pm 0.2}$ | 92.1$_{\pm 0.8}$ | 93.9$_{\pm 0.7}$ | 91.0$_{\pm 5.4}$ |
| CON$_2$ (Invert) | $\mathcal{S}_{\text{LH}}$ | 88.5$_{\pm 0.3}$ | 99.0$_{\pm 0.1}$ | 87.0$_{\pm 0.4}$ | 84.9$_{\pm 0.3}$ | 90.0$_{\pm 0.7}$ | 93.4$_{\pm 0.5}$ | **96.7**$_{\pm 0.2}$ | 97.3$_{\pm 0.0}$ | 95.8$_{\pm 0.1}$ | 97.0$_{\pm 0.0}$ | 93.0$_{\pm 4.8}$ |
| CON$_2$ (Flip) | | 88.9$_{\pm 1.2}$ | 99.2$_{\pm 0.0}$ | 89.8$_{\pm 0.2}$ | 87.0$_{\pm 0.4}$ | 92.8$_{\pm 0.9}$ | 93.9$_{\pm 0.1}$ | 96.3$_{\pm 0.3}$ | 98.4$_{\pm 0.1}$ | 97.0$_{\pm 0.1}$ | 96.9$_{\pm 0.2}$ | 94.0$_{\pm 4.1}$ |
| CON$_2$ (Equalize) | | 91.2$_{\pm 1.0}$ | 98.4$_{\pm 0.1}$ | 88.2$_{\pm 0.3}$ | 78.5$_{\pm 2.1}$ | 90.5$_{\pm 0.2}$ | 87.0$_{\pm 2.1}$ | 95.3$_{\pm 0.3}$ | 94.8$_{\pm 0.6}$ | 93.0$_{\pm 0.6}$ | 93.4$_{\pm 0.7}$ | 91.5$_{\pm 5.0}$ |
| CON$_2$ (Invert) | $\mathcal{S}_{\text{NND}}$ | 90.3$_{\pm 0.3}$ | **99.3**$_{\pm 0.0}$ | 89.3$_{\pm 0.2}$ | 87.0$_{\pm 0.1}$ | 90.2$_{\pm 1.1}$ | 94.0$_{\pm 0.4}$ | **96.7**$_{\pm 0.2}$ | 97.8$_{\pm 0.0}$ | 96.6$_{\pm 0.1}$ | **97.2**$_{\pm 0.1}$ | 93.9$_{\pm 4.2}$ |
| CON$_2$ (Flip) | | 90.1$_{\pm 0.8}$ | **99.3**$_{\pm 0.0}$ | 91.0$_{\pm 0.2}$ | 88.7$_{\pm 0.4}$ | 92.8$_{\pm 0.9}$ | **94.1**$_{\pm 0.2}$ | 96.4$_{\pm 0.2}$ | 98.5$_{\pm 0.1}$ | 97.5$_{\pm 0.1}$ | **97.2**$_{\pm 0.1}$ | 94.6$_{\pm 3.7}$ |

### F.3 Multiple Context Augmentations

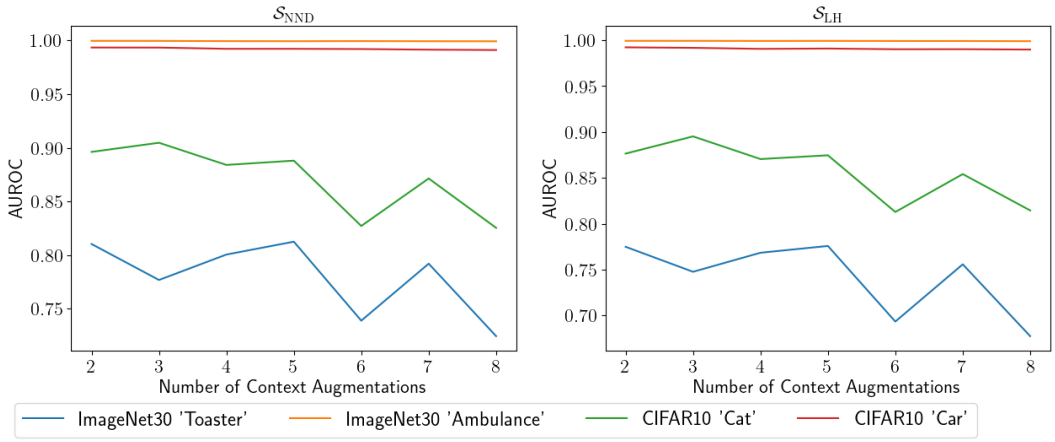



Figure 4: Ablation illustrating the effect of adding more context augmentations. While the performance of well-performing normal classes, such as ImageNet30 *Ambulance* or CIFAR10 *Car* stays consistent when adding more augmentations, we see a decrease for normal classes such as ImageNet30 *Toaster* or CIFAR10 *Cat* that already perform poor to begin with.

Our formulation in Section 2.1 can easily be extended beyond only one additional context by slightly adjusting $\mathcal{L}_{\text{Context}}$. However, in addition to a loss in efficiency due to requiring more memory, we did not find additional context augmentations to provide a performance benefit, as can be seen in Figure 4. There, we ran an ablation with different numbers of context augmentations on different classes of CIFAR10 and ImageNet30. In particular, we trained the adapted CON$_2$ loss for 2, 3, 4, 5, 6, 7, and 8 context augmentations, which we derived by combining *Flip*, *Invert*, and *Equalize* from our previous experiments. Adding more augmentations does not seem to harm cases where we experience good performance in the first place, however, we observe a diminishing performance for slightly more challenging classes.

Table 5: AUROCS for each superclass of CIFAR100 for both of our scores when applying the Flip context augmentation. For each setting, we evaluated our method across three seeds.

| One Class Index | CON$_2$ (Flip) | |
|---|---|---|
| | $\mathcal{S}_{\text{NND}}$ | $\mathcal{S}_{\text{LH}}$ |
| 0 | 85.1$_{\pm 0.4}$ | 83.3$_{\pm 0.1}$ |
| 1 | 85.8$_{\pm 0.8}$ | 85.5$_{\pm 1.3}$ |
| 2 | 93.3$_{\pm 1.2}$ | 93.7$_{\pm 0.9}$ |
| 3 | 90.1$_{\pm 0.4}$ | 91.1$_{\pm 0.2}$ |
| 4 | 94.8$_{\pm 0.4}$ | 94.2$_{\pm 0.5}$ |
| 5 | 84.7$_{\pm 0.3}$ | 82.5$_{\pm 0.2}$ |
| 6 | 92.1$_{\pm 0.5}$ | 91.7$_{\pm 0.6}$ |
| 7 | 84.3$_{\pm 0.7}$ | 84.4$_{\pm 0.6}$ |
| 8 | 90.3$_{\pm 0.6}$ | 89.2$_{\pm 0.5}$ |
| 9 | 95.5$_{\pm 0.2}$ | 94.6$_{\pm 0.3}$ |
| 10 | 87.9$_{\pm 1.0}$ | 85.9$_{\pm 1.0}$ |
| 11 | 91.4$_{\pm 0.3}$ | 91.0$_{\pm 0.4}$ |
| 12 | 91.1$_{\pm 0.3}$ | 90.5$_{\pm 0.4}$ |
| 13 | 83.2$_{\pm 0.5}$ | 80.8$_{\pm 0.6}$ |
| 14 | 96.7$_{\pm 0.0}$ | 96.4$_{\pm 0.2}$ |
| 15 | 80.6$_{\pm 0.7}$ | 79.4$_{\pm 0.6}$ |
| 16 | 86.4$_{\pm 0.7}$ | 85.7$_{\pm 0.6}$ |
| 17 | 97.9$_{\pm 0.1}$ | 97.4$_{\pm 0.2}$ |
| 18 | 96.1$_{\pm 0.3}$ | 95.8$_{\pm 0.2}$ |
| 19 | 94.4$_{\pm 0.3}$ | 93.5$_{\pm 0.3}$ |
| Mean | 89.7$_{\pm 4.2}$ | 89.1$_{\pm 4.6}$ |

Table 6: AUROCS for each class of ImageNet30 for both of our scores when applying the Flip context augmentation. For each setting, we evaluated our method across three seeds.

| One Class Index | CON$_2$ (Flip) | |
|---|---|---|
| | $\mathcal{S}_{\text{NND}}$ | $\mathcal{S}_{\text{LH}}$ |
| 0 | 95.1$_{\pm 0.5}$ | 94.1$_{\pm 0.4}$ |
| 1 | 99.2$_{\pm 0.3}$ | 99.0$_{\pm 0.1}$ |
| 2 | 99.8$_{\pm 0.0}$ | 99.8$_{\pm 0.0}$ |
| 3 | 88.0$_{\pm 0.2}$ | 90.9$_{\pm 0.2}$ |
| 4 | 95.3$_{\pm 0.2}$ | 95.9$_{\pm 0.3}$ |
| 5 | 98.0$_{\pm 0.4}$ | 97.2$_{\pm 0.5}$ |
| 6 | 95.5$_{\pm 0.3}$ | 96.0$_{\pm 0.2}$ |
| 7 | 64.1$_{\pm 3.5}$ | 68.0$_{\pm 2.4}$ |
| 8 | 94.1$_{\pm 0.3}$ | 94.5$_{\pm 0.3}$ |
| 9 | 84.3$_{\pm 0.5}$ | 82.4$_{\pm 0.8}$ |
| 10 | 97.9$_{\pm 0.1}$ | 98.1$_{\pm 0.2}$ |
| 11 | 87.0$_{\pm 0.7}$ | 85.5$_{\pm 0.5}$ |
| 12 | 97.5$_{\pm 0.1}$ | 95.4$_{\pm 0.1}$ |
| 13 | 92.0$_{\pm 0.9}$ | 92.9$_{\pm 0.2}$ |
| 14 | 87.6$_{\pm 0.3}$ | 87.6$_{\pm 1.0}$ |
| 15 | 90.6$_{\pm 0.7}$ | 91.0$_{\pm 0.2}$ |
| 16 | 99.0$_{\pm 0.2}$ | 99.0$_{\pm 0.1}$ |
| 17 | 51.5$_{\pm 1.7}$ | 48.6$_{\pm 1.0}$ |
| 18 | 90.6$_{\pm 0.9}$ | 90.9$_{\pm 1.0}$ |
| 19 | 65.5$_{\pm 2.2}$ | 64.4$_{\pm 1.4}$ |
| 20 | 90.7$_{\pm 0.6}$ | 90.1$_{\pm 0.2}$ |
| 21 | 94.2$_{\pm 1.3}$ | 94.6$_{\pm 0.8}$ |
| 22 | 95.2$_{\pm 0.1}$ | 97.4$_{\pm 0.2}$ |
| 23 | 95.9$_{\pm 0.2}$ | 96.2$_{\pm 0.3}$ |
| 24 | 85.4$_{\pm 1.1}$ | 82.3$_{\pm 1.1}$ |
| 25 | 81.3$_{\pm 5.0}$ | 84.7$_{\pm 3.6}$ |
| 26 | 88.9$_{\pm 0.3}$ | 90.3$_{\pm 0.6}$ |
| 27 | 96.8$_{\pm 0.3}$ | 96.9$_{\pm 0.4}$ |
| 28 | 74.1$_{\pm 1.5}$ | 71.1$_{\pm 0.7}$ |
| 29 | 90.7$_{\pm 0.3}$ | 89.2$_{\pm 0.6}$ |
| Mean | 88.9$_{\pm 11.4}$ | 88.8$_{\pm 11.7}$ |

Table 7: AUROCS for the two classes "Dog" and "Cat" for both of our scores when applying the Flip context augmentation. For each setting, we evaluated our method across three seeds.

| One Class Index | CON$_2$ (Flip) | |
|---|---|---|
| | $\mathcal{S}_{\text{NND}}$ | $\mathcal{S}_{\text{LH}}$ |
| 0 | 91.7$_{\pm 0.2}$ | 91.0$_{\pm 0.2}$ |
| 1 | 88.8$_{\pm 0.8}$ | 89.1$_{\pm 0.4}$ |
| Mean | 90.3$_{\pm 1.7}$ | 90.0$_{\pm 1.1}$ |

Table 8: AUROCS for the two classes "Muffin" and "Chihuahua" for both of our scores when applying the Flip context augmentation. For each setting, we evaluated our method across three seeds.

| One Class Index | CON$_2$ (Flip) | |
|---|---|---|
| | $\mathcal{S}_{\text{NND}}$ | $\mathcal{S}_{\text{LH}}$ |
| 0 | 95.7$_{\pm 0.2}$ | 95.3$_{\pm 0.1}$ |
| 1 | 91.9$_{\pm 0.0}$ | 89.9$_{\pm 0.2}$ |
| Mean | 93.8$_{\pm 2.1}$ | 92.6$_{\pm 2.9}$ |