Towards reliable outlier detection using invertible one-class networks

Robert Schmier[1,2] **Ullrich Koethe**[1] **Christoph-Nikolas Straehle**[2]

[1]Computer Vision and Learning Lab, Heidelberg University, Germany
[2]Bosch Center for Artificial Intelligence, Renningen, Germany

Abstract

This work presents a novel approach to the one-class classification problem by leveraging invertible neural networks (INNs). Our method, "Invertible One-Class Classification" (IOCN), maps the data distribution to a *compact* latent distribution, specifically a uniform distribution on a hypercube. In contrast to the usual latent Gaussian, the uniform distribution defines a clear boundary between inliers and outliers and thus facilitates outlier detection by simply measuring the signed distance to the boundary. To train our mapping, we propose a novel objective function and prove that its optimum is the transport from the data distribution to the uniform distribution in the latent hypercube. Interestingly, this objective is simpler than the traditional maximum likelihood training because it does not require the flow's Jacobian determinant. Experiments demonstrate we outperform standard normalizing flows in outlier detection performance and match the state of the art.

1 INTRODUCTION

Detecting outliers is a central problem in machine learning. As discriminative model may rely on shortcuts [Geirhos et al., 2020], generative models seem to be a better fit for anomaly detection: To learn the generative process, the model needs to learn all semantic information of inliers from the training data. Nalisnick et al. [2019] show that naively using the density learned by the generative model, e.g. a normalizing flow, does not perform well as anomaly score, and subsequent works tries to fix this behavior [Ren et al., 2019, Serrà et al., 2020], often relying on additional data to improve outlier detection [Schirrmeister et al., 2020, Hendrycks et al., 2019, Schmier et al., 2023]. While for images it is straightforward to either generate additional contrastive data trough augmentations as in Ren et al. [2019] or to find hierarchies of datasets as in Schirrmeister et al. [2020], finding such augmentations or hierarchies is challenging for other data domains. Our method does not rely on additional contrastive data, making it applicable across various settings.

We follow the argumentation of Le Lan and Dinh [2021], that the underlying problem is more fundamental: The density relies on the presentation in the data space, and - under a reparametrization of the data space - the density of a data point (and therefore its outlier score) can change significantly.

Therefore, we propose a new approach for outlier detection: By learning the transformation of the data distribution onto a compact latent distribution, we define the outlier score as the signed distance to the boundary of the compact density in the latent space. Reparametrization of the data space may change the magnitude of the outlier score of a point, but will not change its sign, i.e., if the data point is mapped inside the support of the compact latent density (and therefore is considered an inlier) or outside of the support (and therefore is considered an outlier). Our method follows a line of research where the enclosing hypervolume of the inlier data is learned [Schölkopf et al., 1999, Ruff et al., 2018, Goyal et al., 2020]. We use invertible neural networks [Ardizzone et al., 2019] to learn the mapping into the latent space. INNs have gained prominence due to their ability to bijectively map data between input and latent space without destroying information.

The key innovation of our approach lies in our new training algorithm, allowing us to map the training data to a compact latent distribution. This is in strong contrast to maximum likelihood training with normalizing flows [Papamakarios et al., 2021], whose latent distribution must have support over the whole latent space. The basic idea of our training objective is to pull transformed training samples towards the latent distribution, while adding an outward pointing gradient on samples from the latent distribution. We show

Accepted for the 7th Workshop on Tractable Probabilistic Modeling at UAI (TPM 2024).

that our invertible network converges to the desired mapping from the data to the compact latent distribution. By using a compact latent distribution, we create a hypervolume in the data space that encapsulates the training data points.

We use the learned model for anomaly detection: At inference time, we measure the signed distance of transformed data points to the boundary of the compact latent distribution in the latent space. This distance-based measure provides a robust and interpretable indicator of outlier presence in the data.

Our contributions are the following:

1. Proposing a new training paradigm for distribution learning with invertible neural networks
2. Training an invertible neural network with a compact latent base distribution
3. Mathematical derivation of the training objective

2 RELATED WORK

2.1 ANOMALY DETECTION METHODS

Anomaly detection constitutes a foundational challenge in data science, with a multitude of classical methods. For our experimental comparisons, we leverage the PyOD library [Zhao et al., 2019] encompassing these classical methods: KNN is a proximity-based technique, using the distance to the kth neighbor as the outlier score [Angiulli and Pizzuti, 2002]. SOD also relies on neighbours of the test sample, but uses them to construct lower dimensional subspaces to detect outliers [Kriegel et al., 2009] Isolation Forest (IFOREST) tries to isolate the test sample from the training data by randomly selecting features and splits. It uses the path length until the test sample is isolated as measure for its outlierness [Liu et al., 2008]. COPOD is a parameter-free and computationally efficient method, it constructs an empirical copula to the outlierness of the test sample [Li et al., 2020]. We refer to the PyOD library for the implementations and further details. Our approach draws inspiration from the One-Class SVM [Schölkopf et al., 1999], which extends the support vector algorithm to handle unlabelled data by seeking the maximum margin hyperplane encapsulating all inliers in a kernel space. Building upon this, Ruff et al. [2018] refined the concept by employing deep neural networks for embedding learning, imposing strict constraints on network flexibility to prevent trivial solutions. Goyal et al. [2020] add the assumption that points from the inlier class lie on a locally linear low dimensional submanifold to further robustify Deep one-class classification. Moreover, methods like Fu et al. [2024] utilize learned embeddings, measuring the distance to training instances in the embedding space for outlier assessment.

Our idea is closely related to Xiao et al. [2023], who try to learn a mapping from the data distribution to a lower dimensional target distribution by simultaneously minimizing the distance between the projected target distribution to the target distribution and the reconstructing error in the data space. In contrast, our use of invertible neural networks ensures zero reconstruction error and is also well defined for outlier data, while the behavior of an auto-encoder is undefined for data points away from the inlier training data.

There are several notable approaches for anomaly detection, which we compare our results against in the experiments: RCA by Liu et al. [2021] enhances anomaly detection robustness using a framework that integrates robust statistics and machine learning, outperforming existing methods in noisy conditions. ICL [Shenkar and Wolf, 2022] employs invariant contrastive learning to improve anomaly distinguishability by learning invariant representations of normal data. GOAD by Bergman and Hoshen [2020] and SLAD Xu et al. [2023b] leverage classification-based and self-supervised learning approaches, respectively, to achieve state-of-the-art results in anomaly detection by training models on synthetic outliers and self-supervised tasks.

Our focus lies on outlier detection, where models exclusively access normal samples during training, aiming to discern inlier instances from other data at test time on a per-sample basis. This paradigm has been investigated across various methodologies, including k-nearest neighbors-based methods [Nizan and Tal, 2023, Papernot and McDaniel, 2018] and density-based techniques [Schirrmeister et al., 2020, Schmier et al., 2023]. Following Zong et al. [2018] we assume that our model has only access to normal ("clean") data and does not have access to anomalous data or additional training data with different semantics. Another line of research uses unlabeled normal and anomalous ("dirty") data at train time, e.g. the recent survey of Jiang et al. [2023]. One-Class classification methods often rely on an estimate of the outliers in the "dirty" training set Jiang et al. [2023], Ruff et al. [2018], Goyal et al. [2020]. Many approaches use additional labeled data to train their anomaly detection model, e.g., Schirrmeister et al. [2020], Schmier et al. [2023], Hendrycks et al. [2019].

2.2 INVERTIBLE NEURAL NETWORKS

Invertible neural networks represent a specialized class tailored to model diffeomorphisms, transformations that are smooth and possess a smooth inverse. Predominantly utilized within the realm of normalizing flows [Papamakarios et al., 2021], these networks aim to transform data distributions onto simpler base distributions, often achieved by maximizing the likelihood of the training data under the transformed distribution. We utilize in this work the coupling block architecture, where for each block the dimensions are split into a passive and active part. The active part is transformed as an invertible function of the passive part,

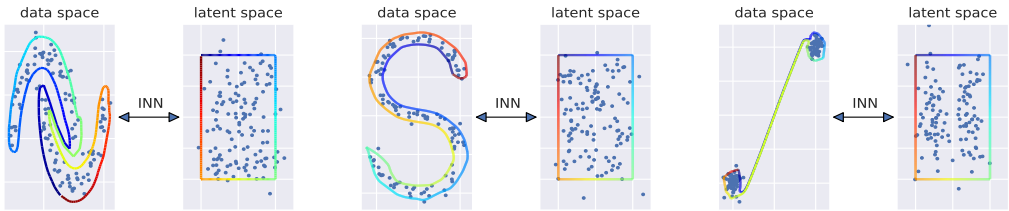


Figure 1: Our method on three 2D datasets. The first two are taken from the scikit-learn library [Pedregosa et al., 2011]. We map the data distribution onto the unit cube in the latent space. Using a compact latent distribution gives a natural outlier decision boundary by mapping the boundary of the latent space cube back into the data space. Since we use an invertible neural net and a uniform latent distribution, the inlier region is simply connected, we see this as a small bridge between the two modes in the moon example (left) and the two gaussians (right).

and the passive part remains unchanged. This was first introduced by Dinh et al. [2015] with addition as invertible transformation, and extended by Dinh et al. [2017] to affine transformations. Coupling blocks are analytically invertible, forward and backward pass are equally efficient to compute. Other methods are only numerically invertible, e.g. autoregressive flows, where the dimensions are ordered and every dimension is transformed as a function of the previous dimensions [Kingma et al., 2017, Papamakarios et al., 2018] or IResNet [Behrmann et al., 2019], which introduces constraints on the Lipschitz constant to render standard ResNet architectures invertible. Invertible neural networks have also been explored in anomaly detection contexts. Grcić et al. [2023] employ an invertible neural network as a normalizing flow to generate negative examples during training, albeit relying on maximum likelihood for training,

3 METHOD

Our method "Invertible one-class network" aims to learn a bijective mapping that transports the inlier data onto a latent hypercube and outliers outside of the hypercube. To achieve this, we utilize a two step strategy that involves pulling inlier data towards the origin and simultaneously adding a gradient on the latent space distribution. First, we sample a minibatch of N training points $\mathbf{x}_i$ and N samples from the latent distribution $\mathbf{z}_i$. We use the invertible neural network (INN) to map the latent distribution in the data space $\tilde{\mathbf{x}}_i = \text{INN}^{-1}(\mathbf{z}_i)$ without computing the gradients. Then we perform a gradient step of the contrastive loss between the two types of points:

$$\mathcal{L} = \sum_{i=0}^{N} \left(f(\text{INN}(\mathbf{x}_i)) - f(\text{INN}(\tilde{\mathbf{x}}_i)) \right). \tag{1}$$

The function f in the loss computation needs to be a convex function, we use the max function over the feature dimensions of a sample. We show pseudocode for our training algorithm in the appendix in section A.4. The loss term for the generated samples tries to push these points out of the inlier region, unless this force is counteracted by an opposing force from a "twin" point in the batch's training set portion. We show in the next section, that this has the net effect that the inlier region converges towards the data points as desired.

3.1 MATHEMATICAL DERIVATION

Without detaching the cube pre-images, the forward and backward pass on the latent samples would cancel each other out, and the network would learn a trivial solution by mapping all inlier data into a small neighborhood of the origin. With the additional gradient, we show that the mapping from data distribution onto the hypercube is a minimum of the optimization problem:

Theorem 3.1. *For a continuous data distribution $p_d(x)$, a continuous latent distribution $p_c(z)$, a pull function $f(z)$ with $f'(z)p_c'(z) < 0\ \forall z$ and a sufficiently powerful invertible network INN, the invertible network has the mapping from the data distribution to the latent distribution as a local minimum for the loss function in equation (1).*

Proof sketch. We introduce a function $g(z, \epsilon, \xi) = \text{INN}_\epsilon(\text{INN}_\xi^{-1}(\mathbf{z})))$, where ϵ and ξ are the deviations from the desired mapping from the data to the latent distribution and reformulate the loss functions in terms of g. We approximate g for small deviations from the optimal solution, calculate the gradient of the loss with respect to the deviations from the desired solution and show that $\nabla_\epsilon \mathcal{L} \approx -\alpha\epsilon$ for $\alpha > 0$. Therefore, gradient descent on the deviation ϵ leads back to the optimal solution. We show the complete derivation in appendix A.3. □

3.2 OUTLIER MEASURE

To quantify outliers, we use the signed distance from data points to the boundaries of the hypercube in the latent space.

Table 1: AUROC scores on CIFAR-10 classes for the One-Vs-Rest setting. PCA, KNN, KDE, COPOD and IForest are taken from the PyOD library [Zhao et al., 2019]. DeepSVDD, RCA, ICL, GOAD and SLAD are taken from the deepOD library [Xu et al., 2023a]. FLOW is a normalizing flow with the same architecture as our "Invertible One-Class Network" (IOCN).

method	plane	car	bird	cat	deer	dog	frog	horse	ship	truck	mean
COPOD [Li et al., 2020]	85.4	97.6	77.5	83.3	90.3	84.6	92.9	91.3	97.2	96.3	89.6
IForest [Liu et al., 2008]	92.8	97.7	85.3	85.5	91.0	88.6	93.4	93.4	97.3	96.5	92.1
KDE [Latecki et al., 2007]	94.7	**99.1**	90.9	90.7	93.6	92.2	96.6	95.4	98.8	98.3	95.0
PCA [Shyu et al., 2003]	94.5	99.0	91.1	90.7	93.7	93.1	97.0	95.6	98.8	98.4	95.2
KNN [Angiulli and Pizzuti, 2002]	95.9	98.7	92.7	90.9	93.9	95.7	98.2	96.2	98.8	97.7	95.9
DeepSVDD [Ruff et al., 2018]	91.2	95.1	84.7	83.9	84.4	87.3	94.2	91.3	94.5	94.5	90.1
GOAD [Bergman and Hoshen, 2020]	94.6	**99.1**	90.7	90.2	93.7	91.6	96.4	95.4	98.7	98.2	94.9
ICL [Shenkar and Wolf, 2022]	96.1	98.8	90.1	89.8	94.0	95.5	98.2	96.3	98.4	98.4	95.6
RCA [Liu et al., 2021]	95.9	99.0	93.1	92.5	93.7	96.0	98.3	96.1	98.8	98.4	96.2
SLAD [Xu et al., 2023b]	**96.8**	99.0	93.9	92.4	94.3	**96.3**	98.5	**96.5**	**99.0**	**98.5**	**96.5**
FLOW [Dinh et al., 2017]	95.5	97.6	93.3	89.5	93.4	95.5	97.8	94.1	98.1	97.1	95.2
IOCN (ours)	96.7	98.3	**94.6**	**92.9**	**94.5**	**96.3**	**98.7**	96.3	98.4	97.9	**96.5**

This distance-based measure distinguishes inliers, which reside within the hypercube, from outliers, which are mapped to the outside of its boundaries.

The outlier measure $s(\mathbf{x})$ for a data point $\mathbf{x}$ is computed as:

$$s_{out}(\mathbf{x}) = \sum_{k=1}^{D} \max\left(|\text{INN}(\mathbf{x})_k| - 1, 0\right), \tag{2}$$

$$s_{in}(\mathbf{x}) = \min(\max_k(|\text{INN}(\mathbf{x})_k|) - 1, 0) \tag{3}$$

$$s(\mathbf{x}) = s_{out}(\mathbf{x}) + s_{in}(\mathbf{x}) \tag{4}$$

where $\text{INN}(x)_k$ represents the k-th dimensions of the latent representation of $\mathbf{x}$.

4 EXPERIMENTS

We conduct experiments across different scenarios. We first investigate two-dimensional datasets in section 4.1. In section 4.2, we apply out method on features extracted from the CIFAR10 dataset [Krizhevsky and Hinton, 2009].

4.1 ILLUSTRATIVE 2D EXPERIMENTS

In Figure 1 we plot several data distributions and the outlier decision boundary, i.e. the surface of the uniform latent distribution mapped back to the data space using the trained INN. Our model effectively maps the inlier data to a cube in the latent space. A drawback of our method is the topological constraint that the inlier distribution must be simply connected, however we observe that the connecting bridges in the latent space have negligible volume in the data space, even though this effect is stronger the lower the dimension D of the data. For $D >= 3$ we expect these bridges to have negligible volume.

4.2 CIFAR EXPERIMENTS

We follow the experimental setup of Schmier et al. [2023] for our CIFAR10 [Krizhevsky and Hinton, 2009] experiments. We do not train directly on images, but use 128-dimensional feature vectors extracted by MoCo [He et al., 2020] as input. We report implementation details and all used methods in the appendix in section A.5. We compare solely to unsupervised methods which do not require negative or contrastive data for training. Training is done on a clean dataset of the inlier class using features extracted from the 5000 training images of the regular CIFAR10 split. We report the area under the receiver operating characteristic curve (AUROC) for all methods when using all other CIFAR10 classes as outliers at test time in table 1. Our method outperforms almost all used anomaly detection methods. While our method matched the performance of SLAD [Xu et al., 2023b], it stands out by being simpler and easier to implement. To get a better insight in our method, we report the confusion matrix for all classes in the appendix in section A.6 and show image examples in Figure 3.

5 CONCLUSION & FURTHER WORK

We introduce a novel training paradigm for one-class classification utilizing invertible neural networks. Our approach shows that it is possible to learn the desired mapping from the data distribution to a latent distribution without relying on the Jacobian determinant. Unlike standard normalizing flows, our method allows for the use of a compact latent distribution, which provides an interpretable outlier measure. We outperform standard normalizing flows in outlier detection and match the state of the art performance of SLAD [Xu et al., 2023b]. Since we do not require the Jacobian determinant future research will investigate alternative invertible architectures compared to split-coupling flows. To alleviate the simple connectedness constraint, other compact latent distributions will be explored.

References

Fabrizio Angiulli and Clara Pizzuti. Fast outlier detection in high dimensional spaces. In *European conference on principles of data mining and knowledge discovery*, pages 15–27. Springer, 2002.

Lynton Ardizzone, Till Bungert, Felix Draxler, Ullrich Köthe, Jakob Kruse, Robert Schmier, and Peter Sorrenson. Framework for Easily Invertible Architectures (FrEIA). Technical report, Computer Vision and Learning Lab, University of Heidelberg, 2018-2023. URL `https://github.com/VLL-HD/FrEIA`.

Lynton Ardizzone, Jakob Kruse, Sebastian Wirkert, Daniel Rahner, Eric W. Pellegrini, Ralf S. Klessen, Lena Maier-Hein, Carsten Rother, and Ullrich Köthe. Analyzing inverse problems with invertible neural networks, 2019.

Jens Behrmann, Will Grathwohl, Ricky T. Q. Chen, David Duvenaud, and Jörn-Henrik Jacobsen. Invertible residual networks, 2019.

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

Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation, 2015.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. In *International Conference on Learning Representations*, 2017.

Dazhi Fu, Zhao Zhang, and Jicong Fan. Dense projection for anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 8398–8408, 2024.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, November 2020. ISSN 2522-5839. doi: 10.1038/s42256-020-00257-z. URL `http://dx.doi.org/10.1038/s42256-020-00257-z`.

Sachin Goyal, Aditi Raghunathan, Moksh Jain, Harsha Vardhan Simhadri, and Prateek Jain. Drocc: Deep robust one-class classification, 2020.

Matej Grcić, Petra Bevandić, Zoran Kalafatić, and Siniša Šegvić. Dense out-of-distribution detection by robust learning on synthetic negative data, 2023.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross 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.

Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations*, 2019.

Minqi Jiang, Chaochuan Hou, Ao Zheng, Songqiao Han, Hailiang Huang, Qingsong Wen, Xiyang Hu, and Yue Zhao. Adgym: Design choices for deep anomaly detection. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 70179–70207. Curran Associates, Inc., 2023.

Diederik P. Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improving variational inference with inverse autoregressive flow, 2017.

Hans-Peter Kriegel, Peer Kröger, Erich Schubert, and Arthur Zimek. Outlier detection in axis-parallel subspaces of high dimensional data. In *Advances in Knowledge Discovery and Data Mining: 13th Pacific-Asia Conference, PAKDD 2009 Bangkok, Thailand, April 27-30, 2009 Proceedings 13*, pages 831–838. Springer, 2009.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario, 2009.

Longin Jan Latecki, Aleksandar Lazarevic, and Dragoljub Pokrajac. Outlier detection with kernel density functions. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 61–75. Springer, 2007.

Charline Le Lan and Laurent Dinh. Perfect density models cannot guarantee anomaly detection. *Entropy*, 23(12): 1690, December 2021. ISSN 1099-4300. doi: 10.3390/e23121690. URL `http://dx.doi.org/10.3390/e23121690`.

Zheng Li, Yue Zhao, Nicola Botta, Cezar Ionescu, and Xiyang Hu. Copod: copula-based outlier detection. In *2020 IEEE international conference on data mining (ICDM)*, pages 1118–1123. IEEE, 2020.

Boyang Liu, Ding Wang, Kaixiang Lin, Pang-Ning Tan, and Jiayu Zhou. Rca: A deep collaborative autoencoder approach for anomaly detection. In *IJCAI: proceedings of the conference*, volume 2021, page 1505. NIH Public Access, 2021.

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422. IEEE, 2008.

Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do deep generative models know what they don't know?, 2019.

Ori Nizan and Ayellet Tal. k-nnn: Nearest neighbors of neighbors for anomaly detection, 2023.

George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation, 2018.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *The Journal of Machine Learning Research*, 22(1):2617–2680, 2021.

Nicolas Papernot and Patrick McDaniel. Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning. *arXiv preprint arXiv:1803.04765*, 2018.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Jie Ren, Peter J Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark Depristo, Joshua Dillon, and Balaji Lakshminarayanan. Likelihood ratios for out-of-distribution detection. *Advances in neural information processing systems*, 32, 2019.

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

Robin Schirrmeister, Yuxuan Zhou, Tonio Ball, and Dan Zhang. Understanding anomaly detection with deep invertible networks through hierarchies of distributions and features. *Advances in Neural Information Processing Systems*, 33:21038–21049, 2020.

Robert Schmier, Ullrich Köthe, and Christoph-Nikolas Straehle. Positive difference distribution for image outlier detection using normalizing flows and contrastive data, 2023.

Bernhard Schölkopf, Robert C Williamson, Alex Smola, John Shawe-Taylor, and John Platt. Support vector method for novelty detection. In S. Solla, T. Leen, and K. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999. URL https://proceedings.neurips.cc/paper_files/paper/1999/file/8725fb777f25776ffa9076e44fcfd776-Paper.pdf.

Joan Serrà, David Álvarez, Vicenç Gómez, Olga Slizovskaia, José F Núñez, and Jordi Luque. Input complexity and out-of-distribution detection with likelihood-based generative models. In *International Conference on Learning Representations*, 2020.

Tom Shenkar and Lior Wolf. Anomaly detection for tabular data with internal contrastive learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=_hszZbt46bT.

Mei-Ling Shyu, Shu-Ching Chen, Kanoksri Sarinnapakorn, and LiWu Chang. A novel anomaly detection scheme based on principal component classifier. In *Proceedings of the IEEE foundations and new directions of data mining workshop*, pages 172–179. IEEE Press, 2003.

Feng Xiao, Ruoyu Sun, and Jicong Fan. Restricted generative projection for one-class classification and anomaly detection. *arXiv preprint arXiv:2307.04097*, 2023.

Hongzuo Xu, Guansong Pang, Yijie Wang, and Yongjun Wang. Deep isolation forest for anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–14, 2023a. doi: 10.1109/TKDE.2023.3270293.

Hongzuo Xu, Yijie Wang, Juhui Wei, Songlei Jian, Yizhou Li, and Ning Liu. Fascinating supervisory signals and where to find them: Deep anomaly detection with scale learning. In *International Conference on Machine Learning*, pages 38655–38673. PMLR, 2023b.

Yue Zhao, Zain Nasrullah, and Zheng Li. Pyod: A python toolbox for scalable outlier detection. *Journal of Machine Learning Research*, 20:1–7, 2019.

Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=BJJLHbb0-.

A APPENDIX

A.1 EXPLANATORY 1D EXAMPLE

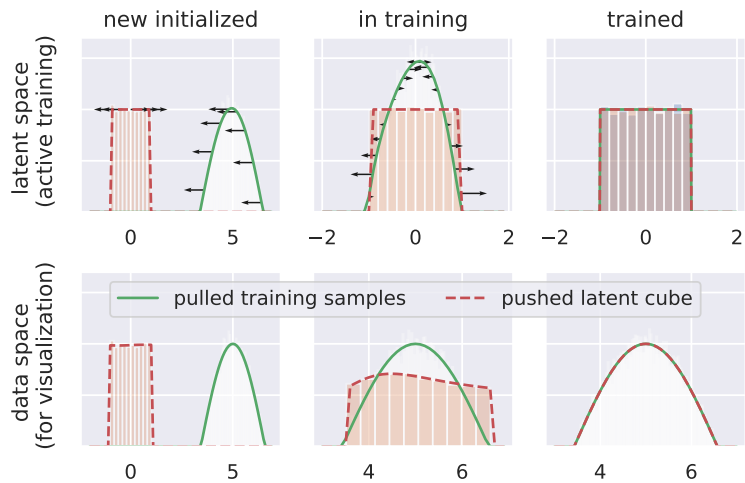


Figure 2: Explanatory 1D example of our method. We show the data and latent space before, while and after training. The black arrows in the latent space show the direction and strength of the loss gradients for the respective x-values. For the untrained model, we see the "pull" on the transformed data distribution and the outwards pointing gradients on the latent distribution. While training, the invertible network experiences gradients where the transformed data and the latent distribution do not coincide. For the perfectly trained model, the gradients vanish.

A.2 MATHEMATICAL DERIVATION FOR L2 LOSS

To explain our mathematical derivation in section A.3, we show the same approach for the L2 classification loss:

Theorem A.1. *For given training data $\{x, y\}_i$ and a prediction function $f_\theta(x)$ from the x-space to the y-space, the prediction function has a local minimum if $f_\theta(x_i) = y_i \; \forall i$*

Proof.

$$\mathcal{L} = \sum_i (f_\theta(\mathbf{x}_i) - y_i)^2 \tag{5}$$

$$\Rightarrow \nabla_\theta \mathcal{L} = \sum_i 2(f_\theta(\mathbf{x}_i, \theta) - y_i) \nabla_\theta f_\theta(\mathbf{x}_i, \theta) \tag{6}$$

$$\Rightarrow \nabla_\theta \mathcal{L}(\tilde{\theta}) = 0 \Leftarrow f_{\tilde{\theta}}(\mathbf{x}_i) = y_i \tag{7}$$

$$\text{linear pertubation in } \epsilon:$$

$$f_\theta(x_i) = f_{\tilde{\theta}}(x_i) + h(x_i)\epsilon = y_i + h(x_i)\epsilon \tag{8}$$

$$\Rightarrow \nabla_\epsilon \mathcal{L} = \sum_i 2h(x_i)\epsilon \cdot h(x_i) \tag{9}$$

$$\Rightarrow \epsilon_{\text{new}} = \epsilon - \alpha \nabla_\epsilon \mathcal{L} = \epsilon(1 - \alpha \sum_i 2h(x_i)^2) \tag{10}$$

$$\Rightarrow |\epsilon_{\text{new}}| < |\epsilon| \text{ for small } \alpha \tag{11}$$

As the new pertubation ϵ_{new} is smaller then the previous pertubation ϵ, the solution $f(x_i) = y_i$ is a local minimum of the optimization problem. □

A.3 MATHEMATICAL DERIVATION

Theorem A.2. *For a continuous data distribution $p_d(x)$, a continuous latent distribution $p_c(z)$, a pull function $f(z)$ with $f'(z)p'_c(z) < 0\ \forall z$ and a sufficiently flexible invertible network, the invertible network INN has the mapping from the data distribution to the latent distribution as a local minimum for the loss function in equation (1).*

Proof. To show that the mapping from the learned network has the mapping from the latent distribution to the data distribution as an minimum, we show that for a small pertubation the gradient of the loss in respect to the pertubation is in the direction of the pertubation, and therefore gradient descent reduces the pertubation. To illustrate the approach, we show the same derivation for the L2 loss in section A.2.

$$\mathcal{L} = \int \mathrm{p}_{\text{data}}(\mathbf{x}) f(\text{INN}(\mathbf{x}))\, d\mathbf{x} - \int \mathrm{p}_{\text{cube}}(\mathbf{z}) f(\text{INN}(\text{sg}(\text{INN}^{-1}(\mathbf{x})))) \, d\mathbf{z} \tag{12}$$

$$= \int \mathrm{p}_{\text{cube}}(\mathbf{z}) f(\text{INN}(\text{INN}_*^{-1}(\mathbf{z})))\, d\mathbf{z} - \int \mathrm{p}_{\text{cube}}(\mathbf{z}) f(\text{INN}(\text{sg}(\text{INN}^{-1}(\mathbf{x})))) \, d\mathbf{z} \tag{13}$$

where INN_*^{-1} is the mapping from the cube to the data distribution

$$= \int \mathrm{p}_{\text{cube}}(\mathbf{z}) \left(f(g(z, \epsilon, 0)) - f(g(z, \epsilon, \xi)) \right) \, d\mathbf{z} \tag{14}$$

with $g(z, \epsilon, \xi) = \text{INN}_\epsilon(\text{INN}_\xi^{-1}(\mathbf{z}))$and $\text{INN}_0 = \text{INN}_*$

where ϵ and ξ are the linear perturbations.

As $g(z, \epsilon, \epsilon) = z$ for the same perturbations of both network directions,

the expansion of g in ϵ and ξ simplifies:

$$g(z, \epsilon, \xi) = z + h_1(z)(\epsilon - \xi) + h_2(z)\epsilon(\epsilon - \xi) + h_3(z)\xi(\epsilon - \xi) + O(\epsilon^3) \tag{15}$$

Using the approximation for g, we approximate $f(g)$):

$$\Rightarrow f(g(z, \epsilon, \xi)) = f(z) + f'(z)h_1(z)(\epsilon - \xi) + f'(z)h_2(z)\epsilon(\epsilon - \xi) + f'(z)h_3(z)\xi(\epsilon - \xi) + f''(z)h_1(z)^2(\epsilon - \xi)^2 + O(\epsilon^3, \xi^3) \tag{16}$$

We plug the approximation of $f(g)$ into eq. 14 to obtain:

$$\Rightarrow \mathcal{L} = \int \mathrm{p}_{\text{cube}}(\mathbf{z}) \left(f'(z)h_1(z)\xi + f'(z)h_2(z)\epsilon\xi) - f'(z)h_3(z)\xi(\epsilon - \xi) + f''(z)h_1(z)^2(2\epsilon\xi - \xi^2) + O(\epsilon^3) \right) d\mathbf{z} \tag{17}$$

Taking the gradient with respect to ϵ gives:

$$\Rightarrow \nabla_\epsilon \mathcal{L} = \int \mathrm{p}_{\text{cube}}(\mathbf{z}) \left(f'(z)h_2(z)\xi - f'(z)h_3(z)\xi + f''(z)h_1(z)^2 2\xi + O(\epsilon^2) \right) \, d\mathbf{z} \tag{18}$$

$$= \int \mathrm{p}_{\text{cube}}(\mathbf{z}) (f'(z)(h_2(z) - h_3(z))\xi + f''(z)h_1(z)^2 2\xi + O(\epsilon^2))\, d\mathbf{z} \tag{19}$$

Using the derivation in section A.3.1:

$$= \int \mathrm{p}_{\text{cube}}(\mathbf{z}) (f'(z)h_1(z)h_1'(z)\xi + f''(z)h_1(z)^2 2\xi + O(\epsilon^2))\, d\mathbf{z} \tag{20}$$

$$= \int \mathrm{p}_{\text{cube}}(\mathbf{z}) (f'(z)\frac{1}{2}(h_1(z)^2)'\xi + f''(z)h_1(z)^2 2\xi + O(\epsilon^2))\, d\mathbf{z} \tag{21}$$

$$= \int \mathrm{p}_{\text{cube}}(\mathbf{z}) ((f'(z)h_1(z)^2)'\xi + f''(z)h_1(z)^2\xi + O(\epsilon^2))\, d\mathbf{z} \tag{22}$$

$$= -\,\xi \int \mathrm{p}'_{\text{cube}}(\mathbf{z}) f'(z)h_1(z)^2 d\mathbf{z} + \xi \int \mathrm{p}_{\text{cube}}(\mathbf{z}) f''(z)h_1(z)^2\, d\mathbf{z} + O(\epsilon^2)) \tag{23}$$

When evaluating, we set $\epsilon = \xi$, this corresponds to the stopgrad in the original loss formulation, as we only take the derivative with respect to ϵ, but when evaluating the forward and backward pass are identical. Therefore, if $\mathrm{p}'_{\text{cube}}(\mathbf{z}) f'(z) < 0\ \forall z \in \sup(p_c)$, both terms are non-negative and gradient descent leads to the optimal solution INN_*. This condition is

easily met by standard loss functions and latent distributions, e.g., uniform cube or normal distribution as latent distribution and squared or maximal absolute distance to the origin as loss function.

□

A.3.1 Additional derivation

We leverage additional knowledge of the introduced function g to establish a relationship between various terms in its Taylor approximation. Specifically, we use the fact that g is a concatenation of two inverse transformations. By interchanging the parameters of these transformations and concatenating two g functions, we obtain the identity. Approximating this concatenated version in terms of the Taylor expansion of the single g function, we derive the relation $h_1' h_1 = h_2 - h_3$ for the Taylor expansion of g.

With $g(z, \epsilon, \xi) = \text{INN}_\epsilon(\text{INN}_\xi^{-1}(\mathbf{z})))$, we can apply g twice with interchanged parameters ϵ and ξ:

$g(g(z, \xi, \epsilon), \epsilon, \xi) = \text{INN}_\epsilon(\text{INN}_\xi^{-1}(\text{INN}_\xi(\text{INN}_\epsilon^{-1}(\mathbf{z})))) = \text{INN}_\epsilon(\text{INN}_\epsilon^{-1}(\mathbf{z})) = \mathbf{z}$,

and therefore with $g(z, \epsilon, \xi) = z + h_1(z)(\epsilon - \xi) + h_2(z)\epsilon(\epsilon - \xi) + h_3(z)\xi(\epsilon - \xi)$:

$\Rightarrow g(g(z, \xi, \epsilon), \epsilon, \xi) = g(z, \xi, \epsilon) + h_1(g(z, \xi, \epsilon))(\epsilon - \xi) + h_2(g(z, \xi, \epsilon))\epsilon(\epsilon - \xi) + h_3(g(z, \xi, \epsilon))\xi(\epsilon - \xi)$

$= z + h_1(z)(\xi - \epsilon) + h_1'(z)h_1(z)(\xi - \epsilon)(\epsilon - \xi) + h_2(z)\xi(\xi - \epsilon) + h_3(z)\epsilon(\xi - \epsilon) + h_1(z)(\epsilon - \xi) + h_2(z)\epsilon(\epsilon - \xi) + h_3(z)\xi(\epsilon - \xi)$

Comparison of coefficient leads to $h_1' h_1 = h_2 - h_3$

A.4 PSEUDOCODE

Algorithm 1 Anomaly Detection using Invertible One-Class Networks

Input: Training dataset $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$
Invertible neural net INN, pull function $f(\mathbf{z})$, e.g., $f(z) = \max_i |z_i|$
Initialize INN parameters

function TRAININN($\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$)
  for epoch $= 1$ to T **do**
    for $i = 1$ to N **do**
      $\mathbf{z} \leftarrow \text{INN}(\mathbf{x}_i)$ ▷ Forward pass through INN
      $\mathcal{L}_{\text{pull}} \leftarrow f(\mathbf{z})$ ▷ Pull latent points towards origin
      Sample latent point $\tilde{\mathbf{z}} \sim \mathcal{U}[-1, 1]^D$
      $\tilde{\mathbf{x}} \leftarrow \text{INN}^{-1}(\tilde{\mathbf{z}})$ ▷ Backward pass from uniform latent point
      $\tilde{\mathbf{x}} \leftarrow \text{stopgrad}(\tilde{\mathbf{x}})$ ▷ Do not use the inverse pass for gradient computation
      $\mathbf{z}_{\text{recon}} \leftarrow \text{INN}(\tilde{\mathbf{x}})$ ▷ Forward pass for reconstructed data
      $\mathcal{L}_{\text{push}} \leftarrow -f(\mathbf{z}_{\text{recon}})$ ▷ Push latent points away from origin
      Update INN parameters by minimizing $\mathcal{L}_{\text{pull}} + \mathcal{L}_{\text{push}}$
    end for
  end for
end function

function ANOMALYDETECTION($\mathbf{x}$)
  $\mathbf{z} \leftarrow \text{INN}(\mathbf{x})$
  $\text{score}_{out} \leftarrow \|\mathbf{z} - \text{clip}(\mathbf{z}, \text{-1}, 1)\|_1$ ▷ Calculate outer distance
  $\text{score}_{in} \leftarrow \min(\max_k(|(\mathbf{z})_k|) - 1, 0)$ ▷ Calculate inner distance
  AnomalyScore $\leftarrow \text{score}_{in} + \text{score}_{in}$
  return AnomalyScore
end function

A.5 IMPLEMENTATION DETAILS AND BASELINE METHODS FOR CIFAR10 EXPERIMENTS

For our CIFAR10 experiments in section 4.2, the MoCo network is pretrained on imagenet and the output of the last layer is used for the training of the anomaly detection methods. We use the same preprocessing as Schmier et al. [2023], namely normalizing the feature vectors on the hypersphere and adding a small amount of noise to obtain a valid density.

We employ the FrEIA library [Ardizzone et al., 2018-2023] for the invertible neural network and use 12 sequential allInOne coupling blocks. We use subnetworks with a single hidden layer with a width of 128 and ReLU activation function. We train a normalizing flow with the same architecture for comparison, and use the negative log likelihood as outlier score. Using a distance based outlier measure for the normalizing flow did not yield an effective outlier score.

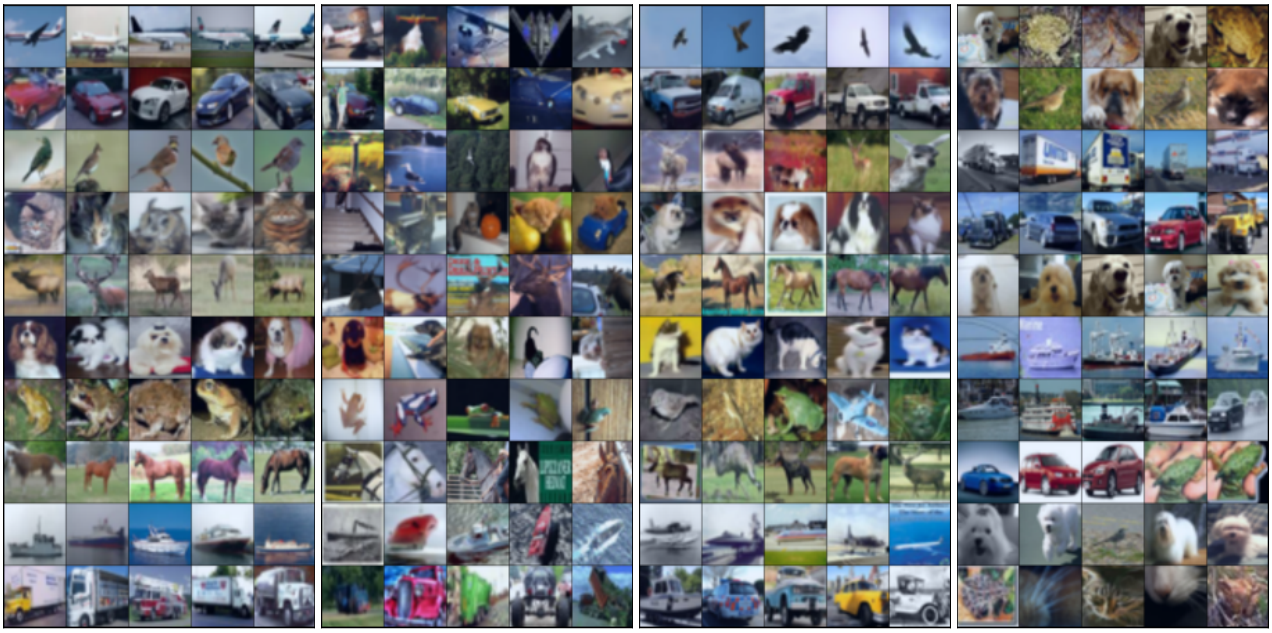
We compare our method only to other unsupervised anomaly detection methods, which do not require additional data and can be trained on a clean dataset of inliers. We used classical anomaly detection methods from the pyod library [Zhao et al., 2019]. We report the following unsupervised methods: k Nearest Neighbors (KNN) [Angiulli and Pizzuti, 2002], "Outlier Detection with Kernel Density Functions" (KDE) [Latecki et al., 2007], "A novel anomaly detection scheme based on principal component classifier" (PCA)[Shyu et al., 2003], "copula-based outlier detection (COPOD)" [Li et al., 2020] and "Isolation Forest (IForest)" [Liu et al., 2008]. We additionally compare with recent deep anomaly detection methods, namely DeepSVDD [Ruff et al., 2018], RCA [Liu et al., 2021], ICL [Shenkar and Wolf, 2022], GOAD [Bergman and Hoshen, 2020] and SLAD [Xu et al., 2023b]. For the deep anomaly detection methods, the implementation of the deepOD library [Xu et al., 2023a] is used. All methods are run with the default hyperparameters.

A.6 CIFAR10-CONFUSION MATRIX

Table 2: Confusion matrix for our "Invertible One-Class Network" on CIFAR10. The model is trained on features extracted by MoCo [He et al., 2020]. The row indicates the training distribution, and the column the test distribution. We see that the performance varies drastically for the class pairs, and semantically similar classes are difficult to separate, e.g. cars and trucks.

	plane	**car**	**bird**	**cat**	**deer**	**dog**	**frog**	**horse**	**ship**	**truck**
plane		98.6	97.0	99.7	99.0	100.0	99.1	99.4	85.8	96.6
car	99.4		100.0	99.9	99.9	100.0	99.9	99.9	99.4	84.9
bird	94.8	100.0		92.2	81.4	95.9	91.3	92.0	99.7	100.0
cat	98.6	100.0	94.2		90.7	66.4	92.1	93.3	99.8	99.9
deer	99.2	100.0	92.7	95.0		96.2	95.6	68.0	99.7	99.9
dog	99.9	100.0	98.3	84.0	95.8		99.2	90.1	100.0	100.0
frog	99.4	100.0	97.1	96.8	96.8	99.5		99.6	99.9	100.0
horse	99.3	100.0	97.2	96.3	81.8	93.3	99.2		99.6	99.9
ship	94.2	98.0	99.8	99.8	99.8	99.9	99.7	99.9		97.5
truck	98.5	87.8	100.0	100.0	100.0	100.0	100.0	100.0	98.6	

B IMAGES



(a) In-distribution with lowest anomaly score (TP)

(b) In-distribution with highest anomaly score (FN)

(c) Outlier with lowest anomaly score (FP)

(d) Outlier with highest anomaly score (TN)

Figure 3: Illustrative images of our CIFAR10 experiments. Each row represents one of the ten CIFAR10 classes used as inliers, showcasing the in-distribution and out-of-distribution images with the highest and lowest anomaly scores. We argue that the model's correct predictions (true positives and true negatives) and the model's errors (false negatives and false positives) are understandable from a human perspective. The FP appear close to the inlier distribution for the human eye, while the FN are non-typical examples of the inlier class.