Anomaly Detection using Contrastive Normalizing Flows

Anonymous authors
Paper under double-blind review

Abstract

Detecting test data deviating from training data is a central problem for safe and robust machine learning. Likelihoods learned by a generative model, e.g., a normalizing flow via standard log-likelihood training, perform poorly as an anomaly score. We propose to use an unlabelled auxiliary dataset and a probabilistic outlier score for anomaly detection. We use a self-supervised feature extractor trained on the auxiliary dataset and train a normalizing flow on the extracted features by maximizing the likelihood on in-distribution data and minimizing the likelihood on the auxiliary dataset. We show that this is equivalent to learning the normalized positive difference between the in-distribution and the auxiliary feature density. We conduct experiments on benchmark datasets and show a robust improvement compared to likelihood, likelihood ratio methods and state-of-the-art anomaly detection methods.

1 Introduction

The performance of neural nets is governed by the availability of vast amount of data, and - with enough training data - neural nets can achieve superhuman performance in various tasks such as classification. If an image at test time is not similar to the training images, but stems from another distribution, classical neural networks may fail to classify the image (Heaven et al., 2019; Boult et al., 2019). The detection of such anomalies, also called out-of-distribution detection or outlier detection, is a central problem in modern machine learning and is crucial for safe and trustable neural networks: At test time, we want to know if a given input stems from the same distribution as during training time, in order to know if we can trust our trained net on that input.

This may not significantly influence the prediction performance at test time, either because the events are that rare, or because the test data is similar to the training data, but introduces a major security risk: If the camera of a self-driving car has a malfunction or something is occluding the view, the car should be able to detect the rareness of the situation and should not use the input of the camera. The use of anomaly detection to improve performance for rare events or unseen data is manifold: One can use anomaly detection while training to purify the data set (Zhao et al., 2019b), at train time to give rare training data points a stronger training signal (Steininger et al., 2021), or at test time to find and react to anomalies (Wang et al., 2021). In contrast to discriminative networks, where the texture (Geirhos et al., 2020) or the background (Beery et al., 2018) can be enough to get a sufficient performance, for anomaly detection the network must distinguish between the in-distribution data and all other -unknown- data. The network therefore must "understand" what the in-distribution data characterizes and following Richard Feynman's famous saying "What I cannot create, I do not understand", we believe that generative models are therefore the most promising approach to anomaly detection. We turn to the fast growing field of normalizing flows (Kobyzev et al., 2021), which allow exact density estimation, fast sampling, and suffer (almost) no mode collapse. While an important line of research deals with detecting small anomalies within an image, so-called defects, mostly in an industrial setting (e.g., Roth et al. (2021a)), we focus in this work on semantic anomaly detection: The goal is to find anomalies on image level. Inspired by the work of Ren et al. (2019) and Schirrmeister et al. (2020), who used the ratio of likelihoods for anomaly detection, we modify the training objective for normalizing flows to learn the positive difference of two distributions: An inlier distribution p and an auxiliary distribution q. We prove that this is achieved by maximizing the log-likelihood on data which stems from p while minimizing the

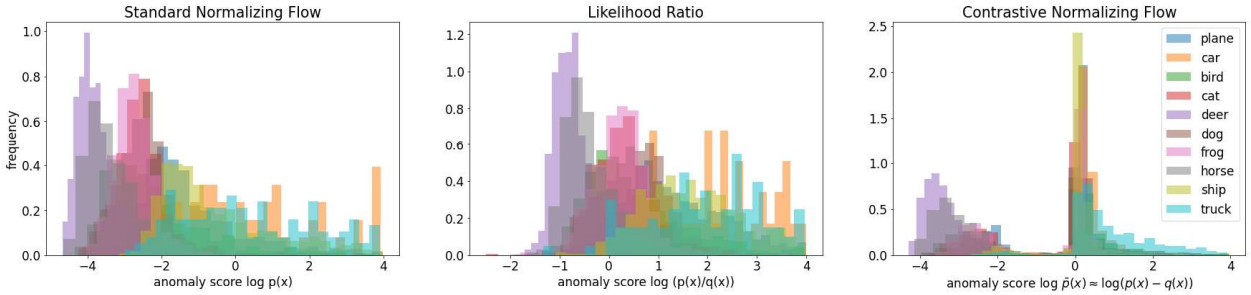

Figure 1: Histogram of the anomaly scores of all CIFAR-10 classes under models trained with the inlier class deer. The semantically similar classes deer and horse are difficult to separate for all methods, our method "contrastive normalizing flow" achieves good separation for the other classes.

log-likelihood on data stemming from q. We use IMAGENET (Deng et al., 2009) as auxiliary distribution, a popular dataset of natural images. We do not use label information, but see the auxiliary dataset as a collection of unlabeled images. We motivate the use of such an auxiliary dataset twofold: Collecting unlabeled images is easy and cheap, but labeling them is expensive and introduces a potential bias in the training. We find that using the difference between the two distributions results in an improved anomaly score. To concentrate on semantic features and to use the auxiliary distribution to full extent, we do not train directly on images but employ a feature extractor trained self-supervised on IMAGENET. This fits the ongoing development of employing pretrained models to use prior knowledge (Bergman et al., 2020).

Our method "anomaly detection using contrastive normalizing flows" combines a pre-trained feature extractor, a generative model with exact density estimation (normalizing flow) and a new training objective for normalizing flows. Our contributions are the following:

- Developing a likelihood-based anomaly score with state-of-the-art performance on benchmark datasets
- Presenting a new objective to train normalizing flows
- Proving the equivalence of our new objective to a negative log-likelihood training of an intractable difference distribution better suited for anomaly detection

2 Related work

Anomaly detection Anomaly detection, also known as out-of-distribution or novelty detection, is the task of detecting unknown images at test time. See Salehi et al. (2021) for a extensive discussion of the field. In this work, we will focus on semantic anomaly detection, by either using one class of a given dataset as inliers and all other classes as anomalies (one-vs-rest setting) or by using a complete dataset as inliers and comparing against other datasets (dataset-vs-dataset setting). Most related work can be categorized as either reconstruction based or using a representation ansatz: In a reconstruction approach, the model tries to reconstruct a given image, and the score relies on the difference between the reconstruction and the original image. This idea goes back to Japkowicz et al. (1995) and has been applied to various domains, e.g. time series (Zhang et al., 2020), medical diagnosis (Lu and Xu, 2018) and flight data (Memarzadeh et al., 2020). Recent models used for image data are reconstruction with a memory module in the latent space (Gong et al., 2019) or combinations of VAE and GAN (Perera et al., 2019). An interesting new idea is the combination of VAE and energy-based models by Yoon et al. (2021), where the reconstruction loss is interpreted as the energy of the model. For the representation approach the model tries to learn a feature space representation and introduces a measure for the outlierness in this representation: Ruff et al. (2018) map all data inside a hypersphere and define the score as the distance to the center. Another approach uses transformations to either define negatives for contrastive learning (Tack et al., 2020) or to directly train a

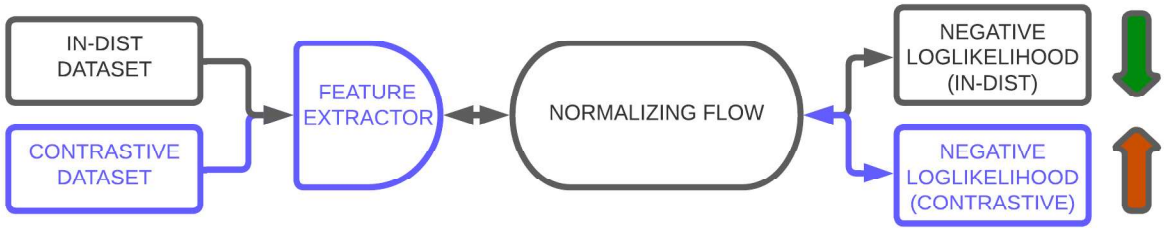

Figure 2: Schematic overview of our method. Standard normalizing flow in black.

classifier (Bergman and Hoshen, 2020). Zong et al. (2018) fit a gaussian mixture model to the latent space of an autoencoder.

Normalizing flows While most generative models are either able to easily generate new samples like GANs or VAEs or to give an exact (possible unnormalized) density estimation (like energy-based models), Normalizing flows, first introduced by Rezende and Mohamed (2015) are able to perform both tasks.Normalizing flows rely on the change of variable formula to compute the likelihood of the the data and therefore need a tractable Jacobian: Most normalizing flows achieve this by either using mathematical "tricks" (e.g., Rezende and Mohamed (2015)), as autoregressive models which restrict the Jacobian to a triangular shape (e.g., Kingma et al. (2016)) or by special architectures which allow invertibility (e.g., RealNVP by Dinh et al. (2017)). RealNVP makes use of coupling blocks to apply an affine transformation on a subset of the features conditioned on the subsets complement.

Density of normalizing flows for anomaly detection The use of generative models seems like a perfect fit for anomaly detection. Classifiers tend to focus on features to distinguish the given classes, whereas generative models need to include all relevant information to be able to generate new samples. Methods with exact density estimation (e.g., normalizing flows, energy-based methods) directly offer a good anomaly score via the density. Unfortunately, normalizing flows work poorly in the case of anomaly detection when employed directly on images: Various works showed that the likelihood is dominated by low-level statistics (Nalisnick et al., 2019b) and pixel-correlations (Kirichenko et al., 2020) and fail to detect anomalies (see also Zhang et al. (2021)). There have been multiple attempts to improve the anomaly score directly by introducing a complexity measure (Serrà et al., 2020), using typicality (Nalisnick et al., 2019a), using hierarchies (Schirrmeister et al., 2020) or employing ensembles (Choi et al., 2019).

Likelihood ratio for anomaly detection Another line of research is using the ratio of two likelihood models as anomaly score: Ren et al. (2019) use an augmented version of the in-distribution as auxiliary distribution and computed the likelihood ratio with Pixel-CNN, an autoregressive generative model. Schirrmeister et al. (2020) train separate normalizing flows on the in-distribution data and Tiny Images as auxiliary dataset. They require the in-distribution dataset to be included in the auxiliary dataset to meet their hierarchical principle.

Feature extractor With the broad availability of trained deep models and their good generalization capacity, the use of such models as feature extractor has gained popularity: Most implementations use the output of intermediate layers of a model trained as a discriminator on an auxiliary dataset (e.g. IMAGENET by Deng et al. (2009)) as features: Prominent examples are ResNet (He et al., 2016), used by e.g., Cohen and Hoshen (2021) and Roth et al. (2021b), or ViT (Dosovitskiy et al., 2021), used by Cohen and Avidan (2021) for anomaly detection. We argue that the use of label information introduces a bias, especially when working on similar datasets to IMAGENET as CIFAR-10 (Krizhevsky et al.) or CIFAR-100) and propose to use a feature extractor trained in a self-supervised fashion via a contrastive learning objective (Chen et al.,

2020). Their and the improved method by He et al. (2020) showed remarkable results on down-stream tasks without relying on label information.

Combining feature extractors and normalizing flows An existing line of work uses the representations given by pretrained feature extractors to detect outliers: Cohen and Hoshen (2021) use the intermediate layer of a pretrained ResNet as the feature representation and use the sum of distances to the k-nearest Neighbours as their anomaly score. Yu et al. (2021) and Rudolph et al. (2021) train a unconditioned normalizing flow on a feature space of a pretrained feature extractor trained on IMAGENET, while Gudovskiy et al. (2021) use conditional normalizing flows (Ardizzone et al., 2019). All of these methods focus mainly on defect detection and localization, while our paper works on the task of semantic anomaly or out-of distribution detection. In contrast to MOCO (He et al., 2020), which is used in this work for the feature extraction, all their feature extractors are trained in a supervised fashion.

3 Background

3.1 Normalizing flows

Normalizing flows are a class of generative models, which allow exact density estimation and fast sampling. A comprehensive guide to normalizing flows can be found in Papamakarios et al. (2021). A normalizing flow maps the given data via an invertible transformation T^{-1} to a normal distribution (of the same dimensionality) by minimizing the empirical KL-Divergence between the transformed data distribution in the latent space and a multivariate normal distribution. This is equivalent to minimizing the negative log-likelihood of the training data under the model. This likelihood can be calculated by the change of variable formula. Therefore, the Jacobian of the transformation needs to be tractable:

$$L(\theta) = -\sum_{\boldsymbol{x}\in X} \log p_\theta(\boldsymbol{x}) = \sum_{\boldsymbol{x}\in X} \frac{||T^{-1}(\boldsymbol{x})||_2^2}{2} - \log|\mathrm{Jac}_T(\boldsymbol{x})|, \tag{1}$$

where X is the training data, θ are the parameters of the invertible transformation T and Jac_T denotes the determinant of the Jacobian. To apply the change of variable formula in equation 1 the transformation needs to be invertible and needs a tractable determinant of the Jacobian. We achieve this by using the realNVP architecture established by Dinh et al. (2017).

3.2 Feature extractor

To achieve good anomaly detection performance, the feature extractor should be trained on diverse images and without class information. We use a pretrained MOCO (He et al., 2020) feature extractor, which is trained self-supervised on IMAGENET. Self-supervised contrastive methods learn a representation by maximizing the similarity of different views of the same image in the feature space. For a training step, every training image is augmented twice, by augmentations consisting of color distortions, horizontal flipping, and random cropping. These augmented images are feed into an encoder network and for a positive pair $\boldsymbol{z}_0, \boldsymbol{z}_1$ and negative representations $\boldsymbol{z}_2, ..., \boldsymbol{z}_N$ (other augmented images) the following loss is optimized for all augmented images:

$$l_0 = -\log \frac{\exp(sim(\boldsymbol{z}_0, \boldsymbol{z}_1))/\tau}{\sum_{k=1}^{N} \exp(sim(\boldsymbol{z}_0, \boldsymbol{z}_i)/\tau}. \tag{2}$$

τ is a temperature scalar and $sim(\boldsymbol{x}, \boldsymbol{y})$ denotes the cosine similarity (cosine of the angle) between $\boldsymbol{x}$ and $\boldsymbol{y}$. The contrastive loss is trained without any label information.

4 Method: Contrastive normalizing flow

We introduce the contrastive normalizing flow as a novel way to train normalizing flows: We adapt the training of normalizing flows by maximizing the log-likelihood on in-distribution data $\boldsymbol{x} \sim p$ and minimizing

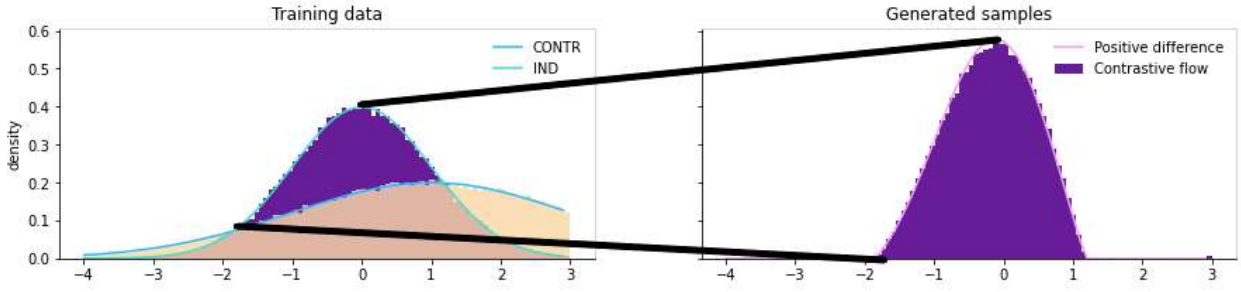

Figure 3: Example of an 1D contrastive normalizing flow: We train on samples $x \sim p = N(0, 1)$ as in-distribution and $y \sim q = N(1, 2)$ as contrastive dataset. The training distributions p and q are shown on the left. The learned distribution of our contrastive normalizing flow method and the analytic normalised positive difference between the ground truth densities are shown on the right. The black lines connect equivalent points in both diagrams.

the log-likelihood on a broader auxiliary dataset $\boldsymbol{y} \sim q$. The intuition behind this approach is simple: We want the model to learn high likelihoods on in-distribution data and low likelihoods everywhere else. We train our model with the objective in equation 3, for illustrative purposes we rearrange the objective into equation 6, which gives a better intuition what this objective achieves :

$$L = \min_\theta \mathbb{E}_{\boldsymbol{x}\sim p}[-\log p_\theta(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{y}\sim q}[-\log p_\theta(\boldsymbol{y})] \tag{3}$$

$$= \min_\theta - \int d\boldsymbol{x} \log p_\theta(\boldsymbol{x})[p(\boldsymbol{x}) - q(\boldsymbol{x})] \tag{4}$$

$$= \min_\theta - \int_{\text{supp}(p>q)} d\boldsymbol{x} \log p_\theta(\boldsymbol{x})[p(\boldsymbol{x}) - q(\boldsymbol{x})] + \min_\theta \int_{\text{supp}(q>p)} d\boldsymbol{x} \log p_\theta(\boldsymbol{x})[q(\boldsymbol{x}) - p(\boldsymbol{x})] \tag{5}$$

$$= \frac{1}{C} \min_\theta \mathbb{E}_{\boldsymbol{x}\sim C(p-q)1_{\{p>q\}}}[-\log p_\theta(\boldsymbol{x})] + \min_\theta \int_{\text{supp}(q>p)} d\boldsymbol{x} \log p_\theta(\boldsymbol{x})[q(\boldsymbol{x}) - p(\boldsymbol{x})], \tag{6}$$

where for the optimal $\hat{\theta}$ the second term in equation 6 gives $p_{\hat{\theta}}(\boldsymbol{x}) \to 0 \ \forall \boldsymbol{x} \in \text{supp}(q > p)$. The first term is the negative log-likelihood objective for $\bar{p} = C(p - q)1_{\{p>q\}}$, with normalizing constant C. This term results in $p_{\hat{\theta}}(\boldsymbol{x}) = C(p - q) \ \forall \boldsymbol{x} \in \text{supp}(p > q)$, as we show below. For the first term in equation 6 holds:

$$L_1 = \frac{1}{C} \min_\theta \mathbb{E}_{\boldsymbol{x}\sim\bar{p}}[-\log p_\theta(\boldsymbol{x})] \tag{7}$$

$$= \frac{1}{C} \min_\theta \mathbb{E}_{\boldsymbol{x}\sim\bar{p}}[\log \frac{\bar{p}(\boldsymbol{x})}{p_\theta(\boldsymbol{x})} - \log \bar{p}(\boldsymbol{x})] \tag{8}$$

$$= \frac{1}{C} \min_\theta KL(\bar{p}, p_\theta) + H(\bar{p}), \tag{9}$$

where $KL(\bar{p}, p_\theta)$ is the KL-divergence between $\bar{p}$ and p_θ with the global minimum for $p_{\hat{\theta}} = \bar{p}$ and the second term is the entropy H of $\bar{p}$ independent of θ. By training our model with equation 3, we learn the normalized positive difference between in-distribution and auxiliary distribution. We give an 1D toy example of a contrastive normalizing flow in figure 3. The pseudocode for our training procedure is given in section 4.1. We argue that the difference distribution is especially suited for anomaly detection: When using a broader distribution as auxiliary contrastive distribution we can assume that $p(vx) > q(\boldsymbol{x})$ on in-distribution data. In regions where the contrastive distribution $q(\boldsymbol{x})$ has a high density the learned positive difference will be small or zero. This behaviour is similar to a likelihood ratio method. An advantage of our method compared to the likelihood ratio is that our learned density p_θ is well defined in areas where $p(\boldsymbol{x})$ and q are

very small, i.e., where the in-distribution data and the contrastive distribution data are sparse. The learned density p_θ is normalized and therefore integrates to 1. Thus, in the areas where both distributions p and q have no support it is zero or almost zero. When comparing $p_\theta(\boldsymbol{x})$ at test time to a threshold to see whether it is an inlier (bigger than threshold) or outlier (smaller than threshold) data points outside of the support of p and q will be reliably detected as outliers. In contrast, the density ratio $p(\boldsymbol{x})/q(\boldsymbol{x})$ is sometimes ill-defined, and due to the division of two small numbers can either be very large or very small. This might lead to problems, because the data which one might want to detect as outlier can be very far from the contrastive distribution which was used during training – a outlier can be anything, and here the likelihood ratio fails.

When separating the probability density into a semantic part and a low-level pixel correlation part, these common low-level correlations cancel out since they are the same for in-distribution and out-of-distribution (Ren et al., 2019). We argue that this is even more beneficial for our difference distribution: We separate the feature dimensions of our data in a high level semantic representation s and a low-level feature representation f, where the low level pixel feature conditional distribution $p(f|s)$ is the same for all natural images. This strong assumption of Ren et al. (2019) is not necessary for our theory to hold but is purely there to help explain the method and give the reader some intuition. We can now rewrite the difference between the in-distribution p and the auxiliary distribution q as

$$p(\boldsymbol{x}) - q(\boldsymbol{x}) = p(\boldsymbol{f}|\boldsymbol{s})p(\boldsymbol{s}) - q(\boldsymbol{f}|\boldsymbol{s})q(\boldsymbol{s}) = p(\boldsymbol{f}|\boldsymbol{s})(p(\boldsymbol{s}) - q(\boldsymbol{s})).$$

This results in a low score when either the low-level features have a low density given the semantic content of the image or the semantic likelihood of the image is higher for the broader auxiliary distribution than for the inlier distribution. We find both cases important classes of anomalies. The anomaly score of our model is the output of the model, which is equivalent to the negative log-likelihood of the positive difference density described in equation 6. By setting a threshold δ, all inputs with a anomaly score $> \delta$ are classified as anomalies. For evaluating, we use area under the receiver operating characteristic (AUROC).

4.1 Training Pseudocode

To clarify how the model is trained, we report the training process in pseudocode in algorithm 1.

Algorithm 1 Training process

With MOCO feature extractor, normalizing flow T_θ and $\text{nll}_\theta(\boldsymbol{x}) = \frac{1}{2}||T_\theta(\boldsymbol{x})||_2^2 - \log|Jac_T(\boldsymbol{x})|$
for all epochs **do**
  for all batches $\boldsymbol{x} \sim$ inlier data and $\boldsymbol{y} \sim$ contrastive data **do**
    $\hat{\boldsymbol{x}}_i = \text{MOCO}(\boldsymbol{x}_i), \hat{\boldsymbol{y}}_i = \text{MOCO}(\boldsymbol{y}_i)$
    $\text{lossPos}_i = \text{nll}_\theta(\hat{\boldsymbol{x}}_i)$
    $\text{lossNeg}_j = \text{nll}_\theta(\hat{\boldsymbol{y}}_j)$
    $\text{lossNeg}_i = \text{clamp}(\text{lossNeg}_i, \text{None}, \text{tsh})$
    $\text{loss} = \frac{1}{N}\sum_i^N \text{lossPos}_i - \frac{1}{M}\sum_j^M \text{lossNeg}_j$
    GradientStep(θ, loss)
  end for
end for

5 Experiments

To extract useful features, we use a MOCO encoder pretrained on IMAGENET (He et al., 2020). As features we use the output of the network, this is in contrast to the original MOCO implementation, where classification is done via an MLP head on the penultimate layer. For our case, working with the last layer results in a better performance for anomaly detection. Because the contrastive MOCO objective is invariant under changes of the norm of the features, we normalize all features to a hypersphere and add small noise afterwards to obtain a valid density. For the normalizing flow implementation, we use the "Framework for Easily Invertible Architectures" (Ardizzone et al., 2018-2022). Unless otherwise specified, we use eight of their "AllinOne"-blocks for our architecture. We list all hyperparamters for training in section A.1 in the

Table 1: AUROC scores on CIFAR-10 classes for the One-Vs-Rest setting. PCA, KDE and KNN are taken from the PyOD library (Zhao et al., 2019a). GOAD from Bergman and Hoshen (2020), CSI from Tack et al. (2020), and Rot+T(rans) from Hendrycks et al. (2019b). MSE, MSE-ratio, and Flow are three ablation methods of our method contrastive normalizing flow (CF). Flow-ratio is the 'Hierarchies of Distributions'-method by Schirrmeister et al. (2020), but applied on the MOCO feature space. OE denotes the outlier exposure method by Hendrycks et al. (2019a). Best AUROC scores per class are bold. † denotes methods using an auxiliary dataset.

method	plane	car	bird	cat	deer	dog	frog	horse	ship	truck	mean
KDE	94.4	99.1	90.4	90.4	93.3	92.1	96.2	94.7	98.5	98.2	94.7
PCA	94.2	98.9	90.6	90.4	93.2	93.0	96.6	95.0	98.6	98.3	94.9
KNN	96.0	98.7	92.3	90.4	93.6	95.7	98.0	95.8	98.5	97.3	95.6
GOAD	75.5	94.1	81.8	72.0	83.7	84.4	82.9	93.9	92.9	89.5	85.1
Rot+T	77.5	96.9	87.3	80.9	92.7	90.2	90.9	96.5	95.2	93.3	90.1
CSI	89.9	99.1	93.1	86.4	**93.9**	93.2	95.1	**98.7**	97.9	95.5	94.3
MSE	94.6	99.1	90.4	90.5	93.7	91.4	96.3	95.2	**98.7**	98.2	94.8
MSE-ratio †	92.8	98.5	89.8	89.7	91.8	92.5	95.4	94.3	98.3	97.6	94.1
Flow	96.1	97.5	92.6	89.8	93.3	95.7	98	94.7	97.8	96.6	95.2
Flow-ratio †	95.9	97.7	93.5	90.0	93.2	95.9	**98.2**	95.2	97.5	96.6	95.4
OE†	96.5	**99.2**	92.9	92.6	93.8	93.8	97.6	96.6	98.4	98.6	96.0
CF (ours)†	**96.9**	99.0	**94.6**	**92.8**	93.5	**96.1**	**98.2**	96.3	98.6	**98.5**	**96.5**

Table 2: Confusion matrix on CIFAR-10 for the contrastive normalizing flow. Every row shows results for a model trained on one Cifar-10 class as inlier distribution and evaluated against all other CIFAR-10 classes. Hard cases (AUROC<90) are printed bold.

	plane	car	bird	cat	deer	dog	frog	horse	ship	truck	mean
plane		98.3	96.5	98.7	98.2	99.4	98.1	98.2	**88.3**	97.2	96.9
car	99.6		99.9	99.8	99.9	99.9	99.8	99.8	99.2	93.0	99.0
bird	94.9	99.9		94.6	**83.5**	97.3	**88.8**	93.3	99.3	99.8	94.6
cat	97.9	99.7	93.9		**88.7**	**72.3**	90.9	92.0	99.1	99.7	92.8
deer	97.7	99.4	92.3	93.8		96.1	93.2	**70.5**	98.8	99.7	93.5
dog	99.5	99.6	98.3	**82.9**	95.8		98.6	91.2	99.4	99.8	96.1
frog	98.8	99.7	96.8	95.9	95.9	98.8		99.4	99.1	99.9	98.2
horse	98.6	99.8	97.1	96.4	**82.3**	94.4	99.0		99.4	99.8	96.3
ship	93.8	98.3	99.6	99.4	99.5	99.6	99.5	99.5		98.4	98.6
truck	98.4	91.2	99.8	99.6	99.7	99.8	99.7	99.5	98.7		98.5

Table 3: AUROC scores in the dataset-vs-dataset setting. We train on CIFAR-10 as IN-DIST and IMAGENET as auxiliary contrastive distribution. Additional results are reported in table 10.

method	CIFAR-10 vs CIFAR-100	CIFAR-10 vs SVHN	CIFAR-10 vs celebA	mean
MSE	70.0	20.0	92.2	60.7
MSE-Ratio†	74.8	42.0	91.5	69.4
Flow	82.9	65.4	**100**	82.8
Flow-ratio	**84.7**	68.3	**100**	84.3
OE†	83.5	65.0	**100**	82.8
CF (ours)	84.4	**90.3**	99.8	**91.5**

appendix. For readability reasons we decided to include the standard deviation over multiple runs not in the main paper, but in the appendix in A.4.

Contrastive Loss Clamping For our loss, we apply a small deviation from the theory: The second term in equation 6 results in the theoretical optimum $p_\theta(\boldsymbol{x}) = 0 \; \forall \boldsymbol{x} \in \text{supp}(q > p)$, but the loss diverges because $\lim_{x \to 0} \log x$ is unbounded. To handle this mismatch, we clamp $\log p(\boldsymbol{x})$ at a threshold ϵ as a lower bound on the likelihood. Therefore the objective leads to $\log p(\boldsymbol{x}) < \epsilon \; \forall \boldsymbol{x} \in \text{supp}(q > p)$, which we find to be sufficient for successful anomaly detection.

MOCO Finetuning For datasets with image sizes significantly smaller than the images used in the MOCO implementation with 224 by 224 pixels, the features are dominated by upsampling artefacts and are highly correlated between different images. To reduce this problem without changing the setup, we finetuned a MOCO trained on 244*244 images on smaller images by shrinking IMAGENET images to 32 by 32 pixels and then using upsamling to 224 by 224 pixels again. We discuss this in appendix A.3.

5.1 Benchmark methods

We used the PyDO library (Zhao et al., 2019a) to compare our results to standard anomaly detection techniques. We present results for PCA, KDE, and KNN conducted on the MOCO feature space, and show additional results for other methods and training directly on the image data (without feature extractor) in the appendix (see A.2). As a simple baseline, we use the mean squared error (MSE) to the mean of the feature space representations of the training set as an outlier score. This is equivalent to fitting a normal distribution to the feature space around the mean of the training data. This simple baseline already gives good results with the finetuned MOCO feature extractor. We extend the MSE to also employ the auxiliary distribution by taking the difference of the MSE to the mean of the inlier set and the contrastive set as anomaly score. This corresponds to the likelihood ratio of two normal distributions. We call this method MSE-ratio. As a third method (Flow), we train an unconditioned flow with the same architecture as our model on the MOCO feature representation and use the negative log-likelihood under the learned distribution as an outlier measure. We show that this results on datasets similar to the auxiliary distribution in a meaningful anomaly measure because of the semantic content of the MOCO features. As fourth baseline method (Flow-ratio), we train an additional flow (with the same architecture as our model) on the auxiliary dataset and use the likelihood ratio of the learned inlier distribution and the learned auxiliary distribution using the hierarchies of distribution method of Schirrmeister et al. (2020). In contrast to their work, the normalizing flows are trained on the MOCO feature space and not on the images directly. We also employ the outlier exposure method of Hendrycks et al. (2019a) on our architecture by training a standard normalizing flow on the inlier data and finetune afterwards with their likelihood margin loss using the auxiliary dataset. At last, we compare our method to "CSI: Novelty Detection via Contrastive Learning on Distributionally Shifted Instances" (Tack et al., 2020). This method is the unsupervised state of the art method for one-class anomaly detection on CIFAR-10. They train their model via contrastive learning, with transformed (shifted) instances of an image itself as additional negatives.

Table 4: AUROC scores on CIFAR-100 superclasses for the One-Vs-Rest setting. Our method contrastive normalizing flow (CF) hast the best anomaly detection performance over all classes. Best AUROC scores per class are bold. † denotes methods using an auxiliary dataset.

method	0	1	2	3	4	5	6	7	8	9	
KDE	90.7	91.0	96.4	95.2	95.7	90.9	96.4	92.7	92.6	96.5	
PCA	91.2	91.0	96.6	95.9	95.6	90.9	96.1	93.2	92.9	96.7	
KNN	91.6	92.7	96.3	**97.1**	**97.2**	95.4	97.2	94.5	95.6	96.5	
CSI	86.3	84.8	88.9	85.7	93.7	81.9	91.8	83.9	91.6	95.0	
MSE	90.0	90.5	96.4	94.5	94.9	89.3	96.0	92.0	91.7	96.0	
MSE-ratio †	90.5	89.5	95.8	95.3	94.7	90.6	95.9	92.6	93.4	96.5	
Flow	91.7	92.7	91.7	92.7	91.7	92.7	91.7	92.7	91.7	92.7	
Flow-ratio †	93.1	92.9	96.0	96.5	97.0	94.5	**97.2**	94.5	95.1	95.9	
OE †	93.3	94.2	94.8	96.8	97.1	94.4	95.5	**94.9**	95.3	**97.2**	
CF (ours) †	**96.8**	**95.4**	**97.0**	95.0	94.4	**96.7**	92.4	93.8	**95.8**	94.3	

method	10	11	12	13	14	15	16	17	18	19	mean
KDE	96.1	92.0	88.7	88.3	96.7	88.8	85.9	97.8	96.4	91.5	93.0
PCA	96.3	92.0	89.5	88.0	96.7	88.6	86.9	97.8	96.0	91.3	93.2
KNN	**97.4**	**94.6**	92.3	90.6	**98.5**	88.8	90.7	97.8	**97.3**	94.0	94.8
CSI	94.0	90.1	90.3	81.5	94.4	85.6	83.0	97.5	95.9	95.1	89.6
MSE	95.8	90.7	88.0	87.0	95.5	87.0	85.5	97.6	95.2	89.4	92.2
MSE-ratio	95.3	92.7	88.8	87.9	96.7	88.7	85.7	96.1	95.9	90.2	92.3
Flow	96.3	93.2	90.1	91.9	97.9	88.8	90.3	97.4	96.7	94.7	93.0
Flow-ratio	96.1	93.6	89.7	91.7	92.6	89.1	91.0	98.0	96.8	95.0	94.6
OE †	**97.4**	94.2	92.0	**92.0**	98.4	91.5	91.3	**98.4**	97.0	95.2	95.0
CF (ours)	91.0	91.4	**98.1**	91.3	96.4	**96.4**	**97.1**	97.2	95.4	**96.7**	**95.1**

5.2 One-vs-rest - results

We run experiments on CIFAR-10, CIFAR-100 (Krizhevsky et al.) and celebA (Liu et al., 2015). We work in the one-vs-rest setting, where one class is used as inlier data, and all the other classes of the same dataset are used as anomalies at test time. For CIFAR-100, we show results for superclasses. For celebA, we divide the dataset into two classes by the given gender attribute and use one of the classes as inlier and the other one as anomalies.

CIFAR-10 and CIFAR-100 superclasses To evaluate how our method performs when the auxiliary contrastive distribution and inlier distribution have significant overlap we evaluated on the CIFAR-10 and CIFAR-100 datasets as inlier distribution with the IMAGENET dataset as contrastive distribution. Note that all CIFAR-10 and -100 classes are a subset of the IMAGENET dataset. The discussion in Section 4 holds also in practice: The broader contrastive IMAGENET distribution with overlap of the CIFAR inlier distributions does not negatively affect the performance, in contrary our method achieves state of the art results on almost all classes when evaluated under a One-vs-Rest setting. We show qualitative results in figure 4: The OOD images with the lowest anomaly score are also for the human eye close to the IN-DIST, while the IN-DIST images with the highest anomaly score are sensible outliers. We show quantitative results in table 1: One can see that our method beats the ablation methods reliably, while the flow-ratio ablation method performs similarly to the simple flow approach. We think this is a result of the training on the intermediate MOCO feature space. To further investigate our method, we show the confusion matrix for all classes in table 2: By looking at the failure cases, we can verify that the model focuses on semantic features: the two worst pairs are deer-horse and cat-dog, which are semantic similar classes. Truck and car have nearly perfect scores; only truck vs car fails (AUROC of about 90). This is also shown in figure 1, where we show the scores of all CIFAR-10 images for a model trained on the deer class as IN-DIST. We conduct the same experiments on the twenty CIFAR-100 superclasses and verify the CIFAR-10 results: Our method outperforms the benchmarks and the ablations methods and increases the anomaly detection performance. The results for all superclasses and methods can be found in table 4.

Table 5: AUROC scores of models trained on the celebA dataset split into the provided gender attribute and evaluated against the other class using contrastive normalizing, CSI, and our ablation methods. LOF, ABOD, MDC and KNN are taken from the PyOD library (Zhao et al., 2019a). Please note that the model only sees images of the in-distribution class and the unlabeled auxiliary IMAGENET dataset at training time.

IN-DIST	LOF	ABOD	MCD	KNN	CSI	Flow	Ratio	OE	**CF (our)**
female	52.5	68.7	69.7	74.1	68.3	75.8	76.7	80.3	**83.6**
male	54.2	76.5	76.6	82.3	79.1	86.7	86.2	**88.2**	87.1
average	53.4	72.6	73.2	78.2	73.7	81.3	81.5	84.3	**85.3**

CelebA To evaluate the interesting setting where the contrastive distribution and the test-time outlier distribution have no significant overlap, we performed an evaluation on the CelebA dataset. We treated each class (we use the gender attribute given in the dataset) once as inlier distribution and the other class as test-time outlier distribution. Even though our contrastive distribution for training the contrastive normalizing flow (and the flow-ratio method) was the IMAGENET dataset, which does not contain close-up pictures of human faces, our method beats the baselines. However, the lacking overlap of the contrastive distribution with the test-time outlier distribution explains why the contrastive approaches such as the flow-ratio method and our contrastive flow do not outperform the other baselines by a clear margin and the overall performance leaves room for improvement. We report the results in table 5.

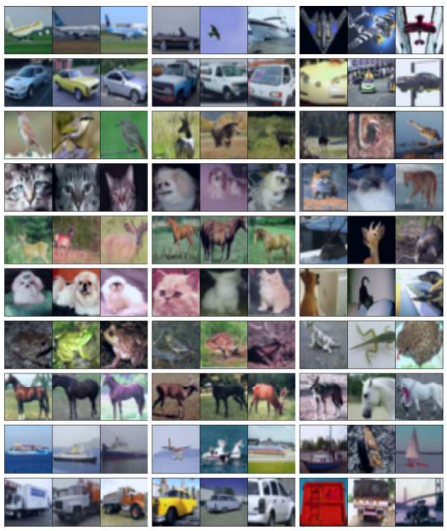

Figure 4: Visual examples from the IN-DIST and OOD test sets. Every row is another CIFAR10 class as IN-DIST. From left to right: the three images with the lowest anomaly score of the IN-DIST test set, the three images with the lowest anomaly score of the OOD test set and the three images of the IN-DIST test set with the highest anomaly score.

5.3 Dataset-vs-dataset

For the dataset-vs-dataset setting, the complete unlabeled dataset is used as the inlier distribution at training time. We train a contrastive normalizing flow on CIFAR-10 and use CIFAR-100, celebA and SVHN as anomaly distributions at test time. We report the AUROC scores in table 3. This setting is well studied, e.g., by Nalisnick et al. (2019b).

6 Conclusion

We propose a novel method to train normalizing flows, which we call contrastive normalizing flow, and show its application to anomaly detection. The method relies on the use of an auxiliary contrastive dataset for training: The training objective is the maximization of the log-likelihood of in-distribution data while minimizing the log-likelihood on the auxiliary dataset. We show that using our training objective, the contrastive normalizing flow learns the normalized positive difference of the in-distribution and the auxiliary distribution. Improvements compared to a standard normalizing flow and a likelihood ratio score for anomaly detection were observed on various tasks. To focus on semantic anomalies rather than on low-level features, we employ a pretrained feature extractor on which the contrastive normalizing flow operates. We show that even under contrastive distributions overlapping with the in-distribution training data and using contrastive distributions far from the tested anomaly distribution, we improve over the standard normalizing flow, likelihood ratio methods and state-of-the-art anomaly detection methods. We believe that the contrastive normalizing flow can be used for various applications outside of anomaly detection and the scope of this work: It gives the possibility to sample from the - usually intractable - positive difference of distributions.

References

Lynton Ardizzone, Till Bungert, Felix Draxler, Ullrich Köthe, Jakob Kruse, Robert Schmier, and Peter Sorrenson. Framework for Easily Invertible Architectures (FrEIA), 2018-2022. URL `https://github.com/VLL-HD/FrEIA`.

Lynton Ardizzone, Carsten Lüth, Jakob Kruse, Carsten Rother, and Ullrich Köthe. Guided image generation with conditional invertible neural networks. *arXiv preprint*, 2019.

Sara Beery, Grant van Horn, and Pietro Perona. Recognition in terra incognita. *arXiv preprint*, 2018.

Liron Bergman and Yedid Hoshen. Classification-based anomaly detection for general data. *ICLR*, 2020.

Liron Bergman, Niv Cohen, and Yedid Hoshen. Deep nearest neighbor anomaly detection, 2020.

Terrance E Boult, Steve Cruz, Akshay Raj Dhamija, Manuel Gunther, James Henrydoss, and Walter J Scheirer. Learning and the unknown: Surveying steps toward open world recognition. *AAAI*, 2019.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *ICML*, 2020.

Hyunsun Choi, Eric Jang, and Alexander A. Alemi. Waic, but why? generative ensembles for robust anomaly detection. *arXiv preprint*, 2019.

Matan Jacob Cohen and Shai Avidan. Transformaly – two (feature spaces) are better than one. *arXiv preprint*, 2021.

Niv Cohen and Yedid Hoshen. Sub-image anomaly detection with deep pyramid correspondences. *arXiv preprint*, 2021.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *CVPR*, 2009.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. *ICLR*, 2017.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

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*, 2020.

Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. *IEEE/CVF*, 2019.

Denis Gudovskiy, Shun Ishizaka, and Kazuki Kozuka. Cflow-ad: Real-time unsupervised anomaly detection with localization via conditional normalizing flows. *arXiv preprint*, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CVPR*, 2016.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. *CVPR*, 2020.

Douglas Heaven et al. Why deep-learning aais are so easy to fool. *Nature*, 2019.

Dan Hendrycks, Mantas Mazeika, and Thomas G. Dietterich. Deep anomaly detection with outlier exposure. *ICLR*, 2019a.

Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. *NeurIPS*, 2019b.

Nathalie Japkowicz, Catherine Myers, and Mark A. Gluck. A novelty detection approach to classification. *IJCAI*, 1995.

Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. *Advances in neural information processing systems*, pages 4743–4751, 2016.

Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Why normalizing flows fail to detect out-of-distribution data. *NeurIPS*, 2020.

Ivan Kobyzev, Simon J.D. Prince, and Marcus A. Brubaker. Normalizing flows: An introduction and review of current methods. *PAMI*, 2021.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). URL `http://www.cs.toronto.edu/~kriz/cifar.html`.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. *ICCV*, 2015.

Yuchen Lu and Peng Xu. Anomaly detection for skin disease images using variational autoencoder. *arXiv preprint*, 2018.

Milad Memarzadeh, Bryan Matthews, and Ilya Avrekh. Unsupervised anomaly detection in flight data using convolutional variational auto-encoder. *Aerospace*, 2020.

Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, and Balaji Lakshminarayanan. Detecting out-of-distribution inputs to deep generative models using typicality. *arXiv preprint*, 2019a.

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

George Papamakarios, Eric T. Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *J. Mach. Learn. Res.*, 2021.

Pramuditha Perera, Ramesh Nallapati, and Bing Xiang. OCGAN: one-class novelty detection using gans with constrained latent representations. *CVPR*, 2019.

Jie Ren, Peter J. Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark A. DePristo, Joshua V. Dillon, and Balaji Lakshminarayanan. Likelihood ratios for out-of-distribution detection. *NeurIPS*, 2019.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. *ICML*, 2015.

Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. *arXiv preprint*, 2021a.

Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. *arXiv preprint*, 2021b.

Marco Rudolph, Bastian Wandt, and Bodo Rosenhahn. Same same but differnet: Semi-supervised defect detection with normalizing flows. *Winter Conference on Applications of Computer Vision*, 2021.

Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. *ICML*, 2018.

Mohammadreza Salehi, Hossein Mirzaei, Dan Hendrycks, Yixuan Li, Mohammad Hossein Rohban, and Mohammad Sabokrou. A unified survey on anomaly, novelty, open-set, and out-of-distribution detection: Solutions and future challenges. *arXiv preprint*, 2021.

Robin Schirrmeister, Yuxuan Zhou, Tonio Ball, and Dan Zhang. Understanding anomaly detection with deep invertible networks through hierarchies of distributions and features. *NeurIPS 2020*, 2020.

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. *ICLR*, 2020.

Michael Steininger, Konstantin Kobs, Padraig Davidson, Anna Krause, and Andreas Hotho. Density-based weighting for imbalanced regression. *Mach. Learn.*, 2021.

Jihoon Tack, Sangwoo Mo, Jongheon Jeong, and Jinwoo Shin. CSI: novelty detection via contrastive learning on distributionally shifted instances. *NeurIPS*, 2020.

Hang Wang, David J. Miller, and George Kesidis. Anomaly detection of test-time evasion attacks using class-conditional generative adversarial networks. *arXiv preprint*, 2021.

Sangwoong Yoon, Yung-Kyun Noh, and Frank Chongwoo Park. Autoencoding under normalization constraints. *ICML*, 2021.

Jiawei Yu, Ye Zheng, Xiang Wang, Wei Li, Yushuang Wu, Rui Zhao, and Liwei Wu. Fastflow: Unsupervised anomaly detection and localization via 2d normalizing flows. *arXiv preprint*, 2021.

Chunkai Zhang, Shaocong Li, Hongye Zhang, and Yingyang Chen. Velc: A new variational autoencoder based model for time series anomaly detection. *arXiv preprint*, 2020.

Lily H. Zhang, Mark Goldstein, and Rajesh Ranganath. Understanding failures in out-of-distribution detection with deep generative models. *ICML*, 2021.

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

Zilong Zhao, Robert Birke, Rui Han, Bogdan Robu, Sara Bouchenak, Sonia Ben Mokhtar, and Lydia Y. Chen. Rad: On-line anomaly detection for highly unreliable data. *arXiv preprint*, 2019b.

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. *ICLR*, 2018.