FlowCon: Out-of-Distribution Detection using Flow-Based Contrastive Learning

Saandeep Aathreya and Shaun Canavan

University of South Florida
{saandeepaath,scanavan}@usf.edu

Abstract. Identifying Out-of-distribution (OOD) data is becoming increasingly critical as the real-world applications of deep learning methods expand. Post-hoc methods modify softmax scores fine-tuned on outlier data or leverage intermediate feature layers to identify distinctive patterns between In-Distribution (ID) and OOD samples. Other methods focus on employing diverse OOD samples to learn discrepancies between ID and OOD. These techniques, however, are typically dependent on the quality of the outlier samples assumed. Density-based methods explicitly model class-conditioned distributions but this requires long training time or retraining the classifier. To tackle these issues, we introduce *FlowCon*, a new density-based OOD detection technique. Our main innovation lies in efficiently combining the properties of normalizing flow with supervised contrastive learning, ensuring robust representation learning with tractable density estimation. Empirical evaluation shows the enhanced performance of our method across common vision datasets such as CIFAR-10 and CIFAR-100 pretrained on ResNet18 and WideResNet classifiers. We also perform quantitative analysis using likelihood plots and qualitative visualization using UMAP embeddings and demonstrate the robustness of the proposed method under various OOD contexts. Code will be open-sourced post decision.

Keywords: OOD detection · flow-based models · contrastive learning

1 Introduction

Visual recognition systems are trained under the closed-world assumption that the input distribution at test time remains consistent with the training distribution. This is seldom the case and the model is expected to identify and reject unknown data instances [2, 13, 15]. In practical scenarios, the test samples may experience gradual distributional shifts and as a result, the model can make arbitrarily incorrect predictions. These shifts can be categorized into *semantic* and *covariate*. Semantic shift (far-OOD) is defined by inclusion of new categories of objects during test time, thereby changing label space. Note that changes in label space naturally impacts the input space as well. On the other hand, covariate shift (near-OOD) is defined by change in the input space only, where the label space remains the same during test time. Collectively, the two present a significant challenge for real world deployment of well-trained systems. This is especially critical in applications such as medical diagnosis [36, 42] and autonomous driving [9].

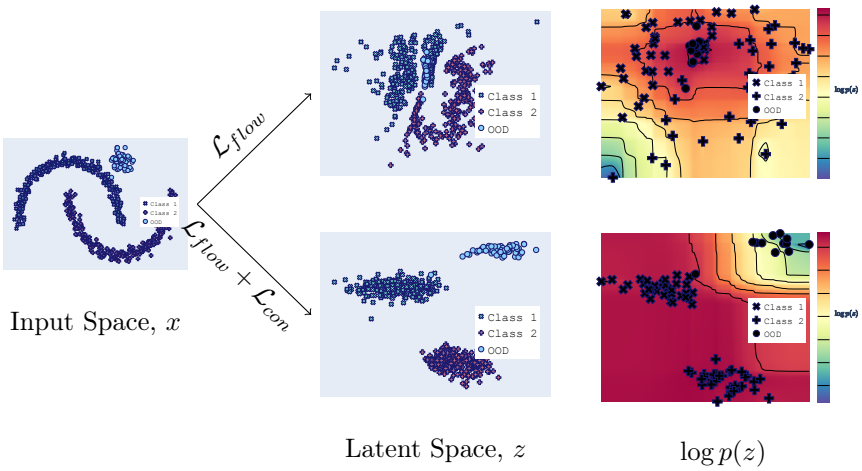


Fig. 1: *Intuition* behind *FlowCon* on toy moons dataset with OOD samples. Normalizing flows trained without contrastive loss ($\mathcal{L}_{flow}$ only) does not account for the class-specific information in the dataset and transforms the data into a unimodal Gaussian distribution as latent space, z. Flow model trained with $\mathcal{L}_{flow} + \mathcal{L}_{con}$ is able to learn class-specific multimodal Gaussian distributions. Consequently, when plotted against $\log p(z)$ using heatmaps, the unimodal Gaussian cluster does not account for OOD samples and assigns high likelihood irrespective of samples it was trained on. Conversely, optimizing $\mathcal{L}_{flow} + \mathcal{L}_{con}$ pushes ID data into the high density region and OOD samples into the low density region.

Existing methods for OOD detection primarily focus on semantic shift detection. Often fine-tuning the softmax scores of the pretrained classifier through temperature scaling [24], energy scoring [26], or thresholding [39]. While these methods are simple but powerful, they have been shown to be less effective under near-OOD context [44,46]. Other methods leverage large OOD datasets in their training paradigm to make the models sensitive to the OOD test set. Nevertheless, it is unrealistic to make assumption on the vast data space of OOD, which might ultimately introduce bias in the model [38].

Density-based methods define the score function using the likelihood values, which explicitly model the ID data and identify the low density test data as OOD [24,49,51]. Although usually reliable, these methods require preserving the class information by training one model per class [51], or retraining the entire classifier under hybrid settings [49]. Zisselman *et al.* [51] proposed deep residual flows to train one model per class for each layer. This results in significant training requirements given the model and ID dataset being used. For example, deep residual flows trained on CIFAR-100 and pretrained with ResNet-18, will result in 400 flow models (100 $classes \times 4$ $layers$). This is potentially infeasible as the complexity of model and dataset grows. Meanwhile, Zhang *et al.* [49] introduced joint training of the flow model and classifier to represent a multi-modal distribution which can be leveraged for OOD detection. This requires retraining the original classifier models which is not suitable for real-world deployment.

In this work, we closely follow OOD detection using density estimation. We build upon the principles of generative models, specifically normalizing flows [33], to develop a contrastive learning-based approach to tackle the above mentioned constraints. More precisely, in addition to maximizing the log-likelihood of the ID data, we introduce a new loss function which contrastively learns the class specific *distributions* of the ID data. Unlike the conventional contrastive losses [5,18] where the similarity function is typically the cosine similarity of two feature vectors, we leverage *Bhattacharyya coefficient* [3], which is designed to measure the similarity between two distributions. The new score emphasizes the network to understand and differentiate between distributions in a contrastive manner. Collectively, the two losses ensure that the network is encouraged to learn semantically meaningful representations enriched with tractable densities. For readability, we name our approach as *FlowCon*. Fig. 1 demonstrates the idea of *FlowCon* on a toy dataset. Maximizing the likelihood ($\mathcal{L}_{flow}$) of the dataset without considering the class information results in a latent space with a single Gaussian cluster. Inclusion of the class-preserving contrastive loss ($\mathcal{L}_{flow} + \mathcal{L}_{con}$) pulls the latent Gaussian distribution belonging to same class together, while pushing other distributions away. The figure also shows $\log p(z)$ values using heatmaps to show *FlowCon*'s discriminative properties along with ID/OOD separability.

To ensure that the original classifier is not modified, we train the flow model and apply the two loss functions ($\mathcal{L}_{flow} + \mathcal{L}_{con}$) on the penultimate layer of the pretrained classifier. As shown by Kirichenko *et al.* [22], training a flow model on deep features focuses on the semantics of the data rather than learning pixel to pixel correlations. To assess the effectiveness of *FlowCon*, we perform quantitative evaluations on benchmark datasets CIFAR-10 and CIFAR-100. We also investigate OOD contexts including far-OOD, near and far-OOD, and near-OOD. The proposed method is competitive or outperforms state-of-the-art OOD detection methods across multiple metrics. To summarize, the contribution of our work is three-fold:

1. A new density-based OOD detection technique called *FlowCon* is proposed. We introduce a new loss function $\mathcal{L}_{con}$ which contrastively learns class separability in the probability distribution space. This learning occurs without any external OOD dataset and it operates on fixed classifiers.
2. The proposed method is evaluated on various metrics - FPR95, AUROC, AUPR-Success, and AUPR-Error and compared against state of the art. We observe that *FlowCon* is competitive or outperforms most methods under different OOD conditions. Additionally, FlowCon is stable even for a large number of classes and shows improvement for high-dimensional features.
3. Histogram plots are detailed along with unified manifold approximations (UMAP) embeddings [28] of the trained FlowCon model to respectively showcase it's OOD detection and class-preserving capabilities. We also show FlowCon's discriminative capabilities.

2 Related Work

Post-hoc methods have the benefit of being straightforward to use. They avoid retraining the original classifier or additional training on top of the classifier.

Hendrycks *et al*. [14] proposed early work on OOD detection by considering the classifier predicted softmax probabilities as the OOD scores. The authors emperically show that the softmax scores of OOD data sufficiently differ from the ID data and therefore forms a baseline for all the subsequent methods. Liang *et al*. [25] applied temperature scaling to the softmax probabilities to improve the ID/OOD separability. Additionally, they applied inverse FGSM [11] on the test data that further improved the separability. Liu *et al*. [26] employed a parameter-free softmax caliberation instead of temperature scaling. The authors replace the softmax score with an energy score whose computation forms a theoretical perspective of likelihoods [30]. Sun *et al*. [39] designed a truncation technique called ReAct on the penultimate activation layer using a threshold. This truncation threshold was chosen to be the 90th percentile of the ID activations. The authors additionally showcase the compatibility of ReAct with previous techniques such as ODIN [25], MSP [14] and Energy [26] that further improved the scores. Lee at al. [24] use Mahalanobis distance to separate ID/OOD samples. The authors compute class-wise empirical mean and covariance for all the average network activations. This is performed over all the training sets which are then modelled as class conditioned Gaussian distributions. During test time, the score is the maximum weighted Mahalanobis distance between test sample and each distribution. In our experiments, we compare FlowCon with these state-of-the-art techniques and show competitive results across multiple metrics.

Outlier-based methods introduce additional training phases extending the pretrained classifier [27, 29, 38, 46]. Hendrycks *et al*. [16] proposed exposing the network to outlier samples thereby enhancing its ability to identify and flag test samples that it has not encountered before. Moreover, the authors leverage OOD data to learn heuristics of the ID data without explicitly modelling them. Hornauer *et al*. [17] propose a heatmap-based approach by attaching a decoder network to a trained classifier layer. Similar to the work from Hendrycks et al., they use outlier OOD samples to define boundaries between ID/OOD samples. A zero-response heatmap output is recognized as ID and a high-response output is categorized as OOD. In our experimental design, we compare FlowCon with the heatmap-based results and follow a similar experimental setup. It is important to note, however, that we did not include them in experiments that require reproducing the results as the OOD dataset they used [41] has been withdrawn[1].

Density-based methods model the ID data without usage of outlier exposure [1,7,34,37,52]. Zhang *et al*. [49] propose joint training of classifier and flow models to ensure stronger discriminative and OOD detection capabilities. The authors present a strong motivation to model likelihood of ID data using normalizing flows and present better results on hybrid training as opposed to using fixed classifiers. However, hybrid approaches require retraining the entire classifier which is not suitable for real-world applications. Zisselman *et al*. [51] address normality assumptions with Mahalanobis distance [24] by training class-wise residual flows [4] for each layer of the model. This ensured that the latent features post residual training are a true Gaussian distribution. On the other hand, we train *FlowCon* only on the penultimate layer of the fixed classifier wherein a single model learns class-wise distribution in a supervised manner. We improve

[1] https://groups.csail.mit.edu/vision/TinyImages/

upon the ResFlow [51] model by resolving the training pipeline to consist of only a single model. We compare *FlowCon* with ResFlows and evaluate extensively on OOD detection performance and histogram interpretability (Section 5.4).

3 Background

In this section, we briefly introduce the formulation of normalizing flows based on coupling layers and supervised contrastive learning.

3.1 Normalizing Flows

Normalizing flows [40] are a class of deep generative models [21] that learn to transform a complex distribution, $p_X(x)$ to a base distribution, $p_Z(z)$ using a sequence of invertible transformations, $z = f(x)$. These transformations are typically in the form of neural networks parametrized by their weights. Using the change of variables formula, the log-likelihood for a datapoint x is maximized by,

$$\mathcal{L}_{flow} = -\log p_X(x) = -\left[\log p_Z(f(x)) + \log \left| \det \frac{\partial f(x)}{\partial x} \right|\right]. \tag{1}$$

The base distribution $p_Z(z)$ is commonly chosen to be standard Gaussian. To satisfy the properties required for Equation 1, f has additional constraints in model architecture. More specifically, f should be bijective and the Jacobian determinant of f should be easy to compute. Addtionally, due to bijective properties of flow models, $x \in \mathbb{R}^d$, and $z \in \mathbb{R}^d$ have same dimensions with $z \sim \mathcal{N}(0, I)$. We refer the readers to the works of Papamakarios et. al. [33] and Dinh et. al. [8] for a comprehensive introduction.

3.2 Supervised Contrastive Learning (SCL)

SCL [18] is a family of representation learning frameworks that aim at learning the most informative deep embeddings of images. Given a set of I data instances $\{x_i, y_i\}_{i=1,\ldots,I}$ in a multi-viewed batch, SCL takes the following form

$$\mathcal{L}_{supcon} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{S(z_i, z_p)/\tau}{\sum_{a \in A(i)} S(z_i, z_a)/\tau)}. \tag{2}$$

Here, $S(z_i, z_j) = \exp(z_i \cdot z_p)$ is the similarity function, $z_i = f(x_i)$ is the latent embedding of the anchor image x_i, P is the set of all positives where $y_i = y_p$, except z_i, and $A(i) \equiv I \backslash i$ is the set of all positives and negatives, except z_i. SCL has shown strong results in creating successful semantic representations of input datasets [43]. Moreover, Winkens *et al.* [44] demonstrated that the contrastive approach to classification further improves OOD detection capabilities of the classifier. In general, Equation 2 aims to minimize distances between data pairs of similar classes while maximizing the distance between dissimilar classes using the dot product between the feature vectors. For a more detailed introduction, we refer the reader to works from Khosla et. al. [18] and Frosst et. al. [10].

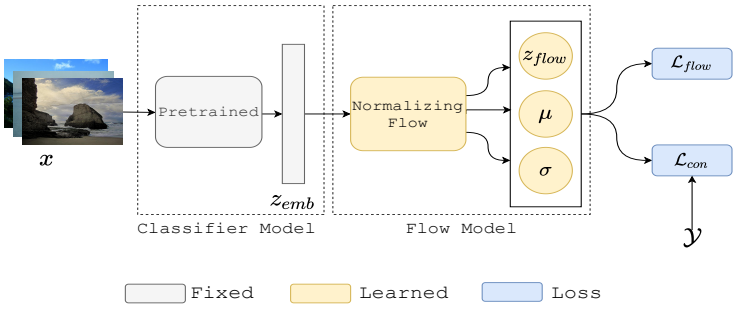


Fig. 2: Training pipeline of *FlowCon*. Given an input image, x, the pretrained classifier first extracts deep features, z_{emb}. The flow model then operates on z_{emb} to obtain the latent vector z_{flow}, and its corresponding distribution, $\mathcal{N}(\mu, \sigma)$. The loss $\mathcal{L}_{flow}$ maximizes the likelihood of z_{flow} on $\mathcal{N}(\mu, \sigma)$, and simultaneously, $\mathcal{L}_{con}$ ensures high inter-class separability and low intra-class separability among the distribution $\mathcal{N}(\mu, \sigma)$ in a contrastive fashion.

4 Flow-based Contrastive Learning

Given an input image $x \in \mathbb{R}^D$ in the batch, *FlowCon* initially extracts high dimensional deep features using a pretrained classifier network. This new embedding, $z_{emb} \in \mathbb{R}^d$ is now used as input to the normalizing flow model to obtain the latent embedding $z_{flow} \in \mathbb{R}^d, \mu \in \mathbb{R}^d$, and $\sigma \in \mathbb{R}^d$. Details regarding training and classification are given in Sections 4.1 and 4.2, respectively. See Fig. 2 for an overview of the proposed architecture.

4.1 Training

To strike a good balance between discriminative and semantic properties of latent embeddings z_{flow}, we propose to effectively combine Equations 1 and 2. Instead of a naive merging of equations, we use an efficient similarity measure, S_{flow}, that uses the likelihood information, $p_Z(z)$, obtained in Equation 1. This reduces the high-dimensional vector dot-product to a simple scalar product of likelihoods. The new similarity function S_{flow} for a given batch is then written as

$$S_{flow}(z_i, z_j, \mathcal{N}_i) = \exp\left((p_Z(z_i|\mathcal{N}_i) \cdot p_Z(z_j|\mathcal{N}_i))^{\tau_1} \right) \tag{3}$$

where,

$$p_Z(z_i|\mathcal{N}_i) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[\frac{-1}{2} \left(\frac{z_i - \mu_i)}{\sigma_i} \right)^2 \right], \tag{4}$$

and τ_1 is a hyperparameter. Here, $p_Z(z_i|\mathcal{N}_i)$ denotes the likelihood of latent embedding z_i belonging to distribution $\mathcal{N}_i$. Note that $p_Z(z_i|\mathcal{N}_i)$ is obtained from Equation 1 (as $\log p_Z(z)$). For ease of sampling in normalizing flows, $\mathcal{N}_i$ resolves

to unit hypersphere. However, since our aim is to learn class-specific distributions, we let the flow network learn the distributions (see Fig 2). [2]

Here, $p_Z(z_i|\mathcal{N}_i)\cdot p_Z(z_j|\mathcal{N}_i)$ should yield a high value when $y_i = y_j$. Conversely, the product should yield a lower value when $y_i \neq y_j$. This is analogous to traditional SCL, where, the higher the dot product between the latent vectors, the more similar the images are, and therefore, the closer they are in the feature space. Conversely, if the dot product is low, the images are dissimilar, and hence, farther apart in the feature space. Therefore, contrastive loss fits naturally in this context. Additionally, the term inside exp in Equation 3 is the generalized form of Bhattacharyya coefficient [3] when the hyperparameter $\tau_1 = 0.5$. Finally, combining Equations 2 and 3 together we get,

$$\mathcal{L}_{con} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{S_{flow}(z_i, z_p, \mathcal{N}_i)/\tau_2}{\sum_{a \in A(i)} S_{flow}(z_i, z_a, \mathcal{N}_i)/\tau_2}. \tag{5}$$

It is important to note that unlike the traditional contrastive learning methods, besides the latent vector z_i, the distribution $\mathcal{N}_i$ also serves as anchor.

Overall, optimizing $\mathcal{L}_{con}$ has an intuitive interpretation as it learns distributions $\mathcal{N}$ in a contrastive manner. On the other hand, $\mathcal{L}_{flow}$ learns embeddings z_{flow} that belongs to the distributions learned by optimizing $\mathcal{L}_{con}$. The loss function in Equation 1 remains the same and we optimize Equations 1 and 5 concurrently with a scaling constant λ as $\mathcal{L} = \mathcal{L}_{con} + \lambda\mathcal{L}_{flow}$.

4.2 OOD Detection with *FlowCon*

Ideally, at the end of training, we obtain a distribution, $\mathcal{N}$, for each data point in the training set $\mathcal{X} = \{x_1, x_2, ..., x_n\}$. For n data points, we will obtain $\mathcal{N}_\mathcal{X} = \{\mathcal{N}_1, \mathcal{N}_2, ..., \mathcal{N}_n\}$ distributions. To enable downstream tasks, such as OOD detection, we simplify the task of dealing with n distributions by reducing them to a smaller set of k distributions, where k is the number of classes. We perform this by taking the emperical mean of the distributions per class. Therefore, the parameters μ_c and σ_c of the distribution $\mathcal{N}_c$ for a class c is computed as

$$\mu_c = \frac{1}{|\mathcal{X}_c|} \sum_{i \in \mathcal{X}_c} \mu_i; \sigma_c = \frac{1}{|\mathcal{X}_c|} \sum_{i \in \mathcal{X}_c} \sigma_i, \tag{6}$$

where $\mathcal{X}_c \equiv \{i \in \mathcal{X} : y_i = c\}$ is the total instances in the training set with class label c. Repeating the process for each class, we get a total of k distributions for the training set $\mathcal{X}$, given by $\mathcal{N}_\mathcal{X} = \{\mathcal{N}_1, ..., \mathcal{N}_k\}$. To compute the score, S for OOD detection on a test sample z_{test}, we simply compute its likelihood on all distributions and take the maximum value $S(x_{test}) = \max_{i \in \{1,...,k\}} p_Z(z_{test}|\mathcal{N}_{y=i})$.

[2] The idea of learned μ and σ was first adopted in the GLOW implementation in https://github.com/openai/glow

5 Experiments

5.1 Setup

Dataset and Models. We use CIFAR-10 [23] and CIFAR-100 [23] datasets as in-distribution (D_{in}). For OOD datasets (D_{ood}), we rely on 6 external test sets: iSUN [45], LSUN-Crop [47], LSUN-Resize [47], SVHN [32], Textures [6], and Places365 [50]. For the pretrained classifier, we use ResNet18 [12] and WideResNet [48] with depth 40 and width 2 which have been trained on both CIFAR-10 and CIFAR-100. This allows us to evaluate FlowCon on various scales of penultimate feature dimensions (512 for ResNet18 and 128 for WideResNet).

Evaluation Metrics. To provide the best metrics independent of a particular OOD score threshold, we evaluate *FlowCon* using four standard metrics, AUROC, AUPR-Success (AUPR-S), AUPR-Error (AUPR-E), and FPR at 95% TPR (FPR-95). AUROC integrates under the reciever operating characteristics (ROC) curve measuring the model's performance across various threshold settings. AUPR-S is area under the precision-recall curve. It focuses on performance of the model in correctly classifying ID samples. Conversely, AUPR-E is the perfomance of the model in correctly identifying OOD samples. FPR-95 is a measure of how often the model incorrectly identifies an OOD sample as ID when it is correctly identifying 95% of ID samples. Similar to prior works [16,26], the entire test set of ID samples are considered and the number of OOD samples are randomly selected to be one-fifth of ID test set. The results are averaged for all OOD datasets.

Implementation Details We follow the experimental setup of Hornauer et al. [17], wherein we build *FlowCon* on top of the pretrained Resnet18 and WideResNet.[3] For the flow model, we adopt a standard RealNVP architecture with 8 coupling blocks and a single flow layer. [4] For ResNet18, we train *FlowCon* on the fixed 512 dimensional penultimate features and 128 dimensional features for WideResNet. The multitask loss function $\mathcal{L}_{total}$ is optimized using Adam optimizer [19] with a fixed learning rate of $1e-5$ and weight decay of $1e-5$. For all experiments, the flow model is trained for 700 epochs with a batch size of 64 with an image size of 32×32. Moreover, we emperically find λ value to be 0.07. We fix the $\mathcal{L}_{con}$ hyperparameters τ_1 and τ_2 at 1.5 and 0.1, respectively.

***FlowCon* vs. Competitive baselines** We compare *FlowCon* with methods that operate only on fixed classifiers. These include:

- *Post-hoc* methods that either calibrate or scale the softmax scores like MSP [14], ODIN [25], Mahalanobis [24], Energy [26], and ReAct [39]. Note that ODIN and Mahalanobis require finetuning the hyperparameters based on external OOD datasets. For all the post-hoc methods, we follow the hyperparameter selection which is consistent with Hornauer et al. [17].

[3] Classifier weights obtained from https://github.com/jhornauer/heatmap_ood
[4] https://github.com/PolinaKirichenko/flows_ood

- *Outlier-based.* Since our approach performs additional training, we also consider methods that train on outlier OOD datasets and a flow-based method. For outlier trained methods, we compare with the heatmap-based approach as proposed by Hornauer *et al.* [17] which showed state-of-the-art results.
- *Flows.* We consider *residual flows* [51] and train the classifier features for all layers in a class-wise manner. Similar to Mahalanobis, residual flows additionally trains a regressor fine-tuned on OOD test sets to predict the scores. In contrast, *FlowCon* operates on penultimate features without a post inference regressor. We show that *FlowCon* outperforms residual flows across the majority of experiments, especially under challenging OOD contexts.

5.2 Result on OOD Contexts

We study the OOD detection capabilities of *FlowCon* under three type of (D_{in}, D_{ood}) pairs. These pairs are cover a broad spectrum of OOD test instances encountered in the real-world. Similar to the works of Winkens et al. [44], each ID dataset will be paired against:

Far-OOD. We compare both CIFAR-10 and CIFAR-100 against the six external datasets (D_{ood}) listed in Section 5.1. These far-OOD experiment pairs are characterized by semantic shifts. The performance of *FlowCon* along with its benchmarks is listed in Table 1. We observe that *FlowCon* performs exceedingly well for CIFAR-10 and CIFAR-100 pretrained on ResNet18 model. This indicates that *FlowCon* is robust even for a higher number of classes. For WideResNet model trained on CIFAR-10, *FlowCon* reports the highest performance for AUROC and AUPR-E with 96.2 and 86.90, respectively. It achieves second-best performance for AUPR-S and FPR-95 reported as 98.84 and 19.10, respectively, after the Heatmap approach [17]. For WideResNet on CIFAR-100, *FlowCon* obtains the best performance on AUPR-S with a score of 96.60 while retaining competitive measures across all other metrics. For instance, ResFlow secures the best AUROC and AUPR-E, however, its AUPR-S remains low, indicating more misclassified ID samples in an effort to filter out OOD samples. Heatmap uses external OOD datasets during training and remains consistent in it's performance with the lowest FPR-95. We provide results on individual OOD datasets in the supplementary material.

Mixed near- & far-OOD. CIFAR-10 (D_{in}) is assessed against CIFAR-100 (D_{ood}), which is regarded as a mixed near- and far-OOD scenario due to the shared classes between the two datasets. This particular pairing involves both semantic and covariate shifts within the test data. Table 2 compares our approach with post-hoc methods and ResFlow. Note that Heatmap is evaluated only on far-OOD context in this work since the dataset primarily used by outlier training methods is 80 million TinyImages [41], which has been withdrawn from further usage[5]. *FlowCon* demonstrates the best performance for ResNet18 model across all metrics. For WideResNet, it acheives the best AUPR-S (96.90) and second best FPR-95 (56.90) scores. Energy-based thresholding [26] achieves the highest AUROC and AUPR-E. Once again, ResFlow reports the lowest FPR-95 which is obtained at the cost of poor AUPR-S measure.

[5] https://groups.csail.mit.edu/vision/TinyImages/

Table 1: *Far-OOD:* Comparison of OOD detection performance during only semantic shift. ∗ Uses OOD data to finetune the hyperparameters. † Uses OOD dataset for training. ‡ Explicitly uses flow models. The results are averaged over the number of OOD test sets (D_{ood}) mentioned in Section 5.

D_{in} (model)	Method	AUROC ↑	AUPR-S ↑	AUPR-E ↑	FPR-95 ↓
CIFAR-10 (ResNet)	MSP [14]	90.72	97.89	63.48	55.21
	ODIN* [25]	88.33	96.67	71.49	38.35
	Mahalanobis* [24]	92.33	98.29	71.30	39.52
	Energy [26]	91.72	97.90	72.12	37.97
	ReAct [39]	91.71	97.89	72.55	36.52
	ResFlow‡ [51]	95.6	99.35	82.82	**13.22**
	Heatmap† [17]	96.47	99.17	83.73	15.37
	FlowCon (Ours)	**97.19**	**99.43**	**85.65**	16.26
CIFAR-10 (WideResNet)	MSP [14]	91.48	98.18	63.47	56.77
	ODIN* [25]	95.01	98.68	84.39	21.09
	Mahalanobis* [24]	92.03	98.09	75.44	32.73
	Energy [26]	94.91	98.75	80.89	24.26
	ReAct [39]	51.92	85.46	17.53	97.12
	ResFlow‡ [51]	81.58	66.09	86.78	49.11
	Heatmap† [17]	96.36	**99.07**	86.73	**14.06**
	FlowCon (Ours)	**96.42**	98.84	**86.90**	19.10
CIFAR-100 (ResNet)	MSP [14]	79.29	95.04	40.34	76.58
	ODIN* [25]	83.28	95.96	48.74	67.96
	Mahalanobis* [24]	73.46	93.00	35.90	79.46
	Energy [26]	82.07	95.71	43.92	74.45
	ReAct [39]	84.22	96.27	49.08	67.78
	ResFlow‡ [51]	85.12	71.45	**67.89**	42.55
	Heatmap† [17]	86.74	96.49	58.78	52.73
	FlowCon (Ours)	**88.22**	**96.85**	**67.89**	**41.85**
CIFAR-100 (WideResNet)	MSP [14]	65.31	90.38	26.21	88.45
	ODIN* [25]	79.43	94.60	43.98	73.19
	Mahalanobis* [24]	73.99	92.58	43.80	68.45
	Energy [26]	77.11	93.95	39.07	78.03
	ReAct [39]	80.74	95.24	48.04	67.47
	ResFlow‡ [51]	**88.58**	62.36	**89.17**	65.77
	Heatmap† [17]	85.98	95.96	61.14	**49.86**
	FlowCon (Ours)	83.62	**96.60**	53.34	60.28

Near-OOD. When CIFAR-100 (D_{in}) is tested against CIFAR-10 (D_{ood}), it is treated as a near-OOD context because the test set experiences covariate shift without any semantic alterations. Some literature treat covariate shift in test set as ID data [46]. We believe that correctly classifying a test data under extreme covariate shift reflects a classifier's generalization ability, thus its identification is crucial. As can be seen in Table 3, *FlowCon* exhibits superior performance over other methods by attaining the highest score on ResNet18 model and reports the best outcome on FPR-95 (82.85) for WideResNet model. Overall, the scores for WideResNet model are shared by MSP, ODIN, Energy, ResFlow, and FlowCon, further highlighting the challenges with near-OOD scenarios. Interestingly, for WideResNet, we note that post-hoc methods in general display moderately better performance for near-OOD scenarios as opposed to far-OOD and mixed-OOD where training based methods (ResFlow/Heatmap) displayed

Table 2: *Near-far and near-OOD.* Comparison of OOD detection performance during both semantic and covariate shift. ∗ uUses OOD data to finetune the hyperparameters. † Uses OOD dataset for training. ‡ Explicitly uses flow models.

D_{in} (model)	D_{ood}	Method	AUROC ↑	AUPR-S ↑	AUPR-E ↑	FPR-95 ↓
CIFAR-10 (ResNet)	CIFAR-100	MSP [14]	86.45	96.49	53.15	65.95
		ODIN* [25]	64.79	89.86	24.32	90.85
		Mahalanobis* [24]	63.90	88.83	29.57	82.55
		Energy [26]	85.60	95.87	57.66	55.2
		ReAct [39]	85.36	95.76	57.51	54.85
		ResFlow‡ [51]	76.40	26.23	66.23	67.2
		FlowCon (Ours)	**93.97**	**98.74**	**73.84**	**35.95**
CIFAR-10 (WideResNet)	CIFAR-100	MSP [14]	86.47	96.87	52.43	67.65
		ODIN* [25]	71.89	91.48	33.74	80.6
		Mahalanobis* [24]	65.40	88.95	29.83	83.4
		Energy [26]	**87.50**	96.84	60.90	**52.85**
		ReAct [39]	63.7	90.36	22.31	92.25
		ResFlow‡ [51]	53.38	12.01	**90.59**	94.53
		FlowCon (Ours)	85.24	**96.90**	57.77	56.9

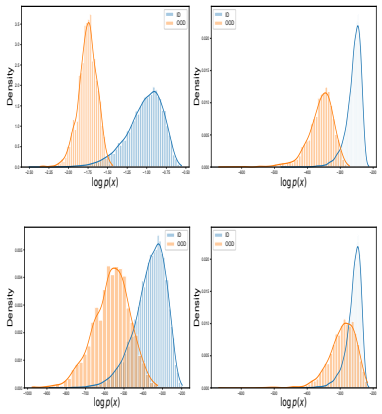
Table 3: *Near-OOD* Comparison of OOD detection performance during only covariate shift. ∗ Uses OOD data to finetune the hyperparameters. † Uses OOD dataset for training. ‡ Explicitly uses flow models.

D_{in} (model)	D_{ood}	Method	AUROC ↑	AUPR-S ↑	AUPR-E ↑	FPR-95 ↓
CIFAR-100 (ResNet)	CIFAR-10	MSP [14]	76.53	94.25	35.28	82.5
		ODIN* [25]	60.46	88.79	20.91	93.01
		Mahalanobis* [24]	42.54	81.21	13.53	98.6
		Energy [26]	77.06	94.26	36.00	81.15
		ReAct [39]	50.49	73.63	16.7	95.2
		ResFlow‡ [51]	58.29	46.34	47.48	79.0
		FlowCon (Ours)	**82.80**	**95.79**	**48.79**	**67.6**
CIFAR-100 (WideResNet)	CIFAR-10	MSP [14]	72.85	**93.46**	31.54	85.75
		ODIN* [25]	62.00	89.39	22.08	92.05
		Mahalanobis* [24]	42.97	81.13	13.7	98.35
		Energy [26]	**74.30**	**93.46**	32.94	83.6
		ReAct [39]	49.08	82.9	16.33	95.01
		ResFlow‡ [51]	59.22	18.08	**92.34**	90.81
		FlowCon (Ours)	67.03	90.16	27.86	**82.85**

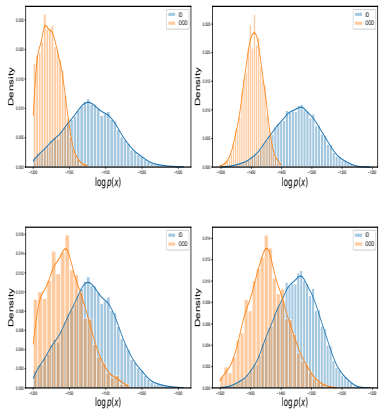
optimum performance. Moreover, the difference in performance between ResNet and WideResNet for *FlowCon* is, in part, due to the feature dimensions on which it operates on. Coupling-based flow architectures like RealNVP [8] and Glow [20] have shown promising results on higher dimensional data as opposed to low-dimensional features [35] (e.g., 128 for WideResNet).

5.3 Likelihood Plots

As we move across different OOD spectrums (Tables 1 - 3), we observe a gradual decrease in performance. Since *FlowCon* is modelled on probability densities, it allows us to effectively understand and visualize the impact of the OOD spectrum by plotting the histogram of log-likelihood. Figs. 3a and 3b plot likelihood values for CIFAR-10 and CIFAR-100 respectively. For each figure, the top row presents



(a) D_{in} = CIFAR-10. **Top** row represent Far-OOD where D_{ood} is external OOD dataset. **Bottom** row indicates Near-OOD where D_{ood} is CIFAR-100.



(b) D_{in} = CIFAR-100. **Top** row represent Far-OOD where D_{ood} is external OOD dataset. **Bottom** row indicates Near-OOD where D_{ood} is CIFAR-10.

Fig. 3: Log-likelihood plots of trained *FlowCon*

the far-OOD context. For Fig. 3a, the bottom row presents the mixed-OOD context where *FlowCon* trained on CIFAR-10 is evaluated on CIFAR-100 test data as OOD. Conversely, in Fig. 3b the bottom presents near-OOD context where *FlowCon* is trained on CIFAR-100 and evaluated on CIFAR-10 as OOD.

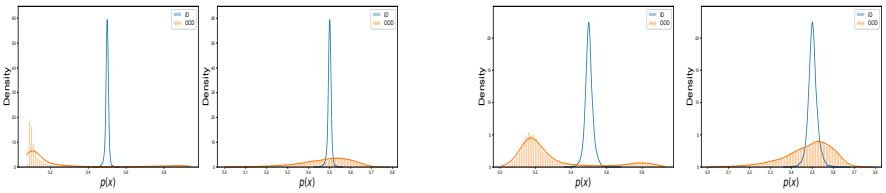
For both Figs. 3a and 3b, as we move from top to bottom row, it is apparent that the overlap between likelihood plots increases. This is in agreement with the performance reported in Section 5.2 as we observe a decline in metrics under more challenging OOD contexts. However, it is crucial to highlight that even under near-OOD conditions, the highest likelihood of falsely accepted OOD samples never exceeds that of the highest accepted ID sample. This aspect of *FlowCon* is pivotal to its robust performance since it addresses an important issue of flow models described by Kirichenko *et al.* [22] where normalizing flow models assign highest likelihood to OOD samples regardless of its training dataset.

5.4 Comparison with ResFlow

FlowCon addresses a critical constraint of ResFlow [51] models. Unlike ResFlows, our training framework is independent of the number of layers of the classifier or the number of classes in the ID dataset. Since ResFlow models all the of intermediate features of the classifier, we evaluate the regressor scores assigned on near-OOD context. Fig. 4 plots the histogram of the scores predicted by the ResFlow model on CIFAR-100 as ID and CIFAR-10 as OOD dataset. Due to a well calibrated regressor, the scores for the ID dataset have a low variance. However, unlike the likelihood plots for *FlowCon*, ResFlows assigns the highest likelihood for OOD samples and not the ID samples.

5.5 *FlowCon* as Classifier

Since we train *FlowCon* in a contrastive manner, we hypothesize that the emperical class-wise distributions $\mathcal{N}_{\mathcal{X}}$ computed in Section 4.2 captures the discrimi-



(a) Far-OOD **(b)** Near-OOD ResNet18 **(c)** Far-OOD **(d)** Near-OOD WideResNet

Fig. 4: Histogram plots on regressor scores of CIFAR-100 trained on ResFlow.

native information of the original classifier. To test this, we predict the class of a given test sample, z_{test} using Bayes' decision rule as

$$y_{test} = \arg \max_{i \in \{1,\dots,k\}} p_Z(z_{test}|\mathcal{N}_{y=i}) \tag{7}$$

Using this, we compute the image classification accuracy and compare it with the original pretrained classifier on CIFAR-10 and CIFAR-100 using both ResNet18 and WideResNet. Table 4 reports the accuracy scores. We observe that difference between *FlowCon* and the original classifier is negligible, therefore the assertion that *FlowCon* is a class-preserving approach remains valid. Furthermore, the in case of WideResNet, our approach marginally outperforms the original classifier. This implies that a single branch suffices in both OOD detection and ID classification.

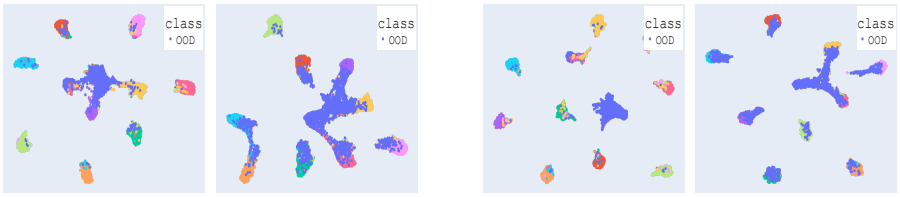
Table 4: Class-preserving property of *FlowCon*. The classification accuracy of our approach remains closely bounded to the original classifier.

D_{in}	Model	Method	Accuracy ↑
CIFAR-10	ResNet18	Orig	94.3
		FlowCon	94.2
CIFAR-10	WideResNet	Orig	93.3
		FlowCon	93.8
CIFAR-100	ResNet18	Orig	75.8
		FlowCon	74.9
CIFAR-100	WideResNet	Orig	70.9
		FlowCon	71.1

6 Discussion

Latent Space Visualization. Apart from visualizing the density plots, we plot the low-dimensional UMAP embeddings [28] of the learned features z_{flow}. We show the plots for CIFAR-10 trained on both ResNet18 and WideResNet in Fig. 5. The blue color represents the OOD data. For far-OOD, we use SVHN [32] and for near-OOD, we use CIFAR-100. The UMAP embeddings exhibit a well-clustered latent space that further supports the classification ability of *FlowCon*. Moreover, for near-OOD contexts, we can observe the ID class clusters overlap

with OOD data with similar semantics. This is analogous to overlapping of the likelihood plots as shown in Fig. 3 (bottom rows).



(a) Far-OOD **(b)** Near-OOD ResNet18 **(c)** Far-OOD **(d)** Near-OOD WideResNet

Fig. 5: UMAP embeddings of z_{flow} trained on CIFAR-10 using *FlowCon*

Impact of λ Nalisnick *et al.* [31] first explored the role of λ in the context of flow-based classification. The value $\lambda = 1/d$, where d is the dimension of z_{flow} was found to be most suitable for OOD detection. In case of *FlowCon*, we found 0.07 to be the most suitable value optimizing both $\mathcal{L}_{flow}$ and $\mathcal{L}_{con}$. We experiment with different λ values in the range of $[0.05, 1]$ for WideResNet trained on CIFAR-100 and report the results in Table 5. The table demonstrates that an increasing λ value reduces the overall performance of *FlowCon*.

Table 5: Effect of λ in optimizing $\mathcal{L}_{flow}$ and $\mathcal{L}_{con}$ under far-OOD context. Shaded region reports λ values used in experiments.

D_{in} (model)	λ	AUROC ↑	AUPR-S ↑	AUPR-E ↑	FPR-95 ↓
CIFAR-100 (WideResNet)	0.05	75.62	92.7	41.84	72.58
	0.07	83.62	96.60	53.34	60.28
	0.3	75.75	92.76	48.61	63.67
	0.5	78.60	93.96	49.07	65.92
	1.0	78.57	93.24	45.94	67.85

Limitations and Future Work. One of the constraints of normalizing flows [33] is that the dimensions of input (z_{emb}) and output (z_{flow}) should be the same. This potentially enforces the model to operate on low dimensional feature vectors depending on the classifier network being used, as observed in experiments pertaining to WideResNet. We emphasize that the reduced dimensionality constrains the learning of *FlowCon*, which we will address in future work.

7 Conclusion

A new approach to OOD detection called *FlowCon* was proposed, which does not use an external dataset as OOD or retrain the original classifier. The key intuition of our approach is that the class-informed density estimator can recognize OOD data simply by filtering out low density samples. The proposed approach operates on deep features, instead of the raw input space, and therefore can be extended to different domains. The best results were obtained on ResNet18 features on all OOD contexts and exhibited competitive performance on WideResNet features.

References

1. Abati, D., Porrello, A., Calderara, S., Cucchiara, R.: Latent space autoregression for novelty detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 481–490 (2019)
2. Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., Mané, D.: Concrete problems in ai safety. arXiv preprint arXiv:1606.06565 (2016)
3. Bhattacharyya, A.: On a measure of divergence between two multinomial populations. Sankhyā: the indian journal of statistics (1946)
4. Chen, R.T., Behrmann, J., Duvenaud, D.K., Jacobsen, J.H.: Residual flows for invertible generative modeling. Advances in Neural Information Processing Systems **32** (2019)
5. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PMLR (2020)
6. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., Vedaldi, A.: Describing textures in the wild. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3606–3613 (2014)
7. Deecke, L., Vandermeulen, R., Ruff, L., Mandt, S., Kloft, M.: Image anomaly detection with generative adversarial networks. In: Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2018, Dublin, Ireland, September 10–14, 2018, Proceedings, Part I 18. pp. 3–17. Springer (2019)
8. Dinh, L., Sohl-Dickstein, J., Bengio, S.: Density estimation using real nvp. arXiv preprint arXiv:1605.08803 (2016)
9. Filos, A., Tigkas, P., McAllister, R., Rhinehart, N., Levine, S., Gal, Y.: Can autonomous vehicles identify, recover from, and adapt to distribution shifts? In: International Conference on Machine Learning. pp. 3145–3153. PMLR (2020)
10. Frosst, N., Papernot, N., Hinton, G.: Analyzing and improving representations with the soft nearest neighbor loss. In: International conference on machine learning. pp. 2012–2020. PMLR (2019)
11. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
13. Hendrycks, D., Carlini, N., Schulman, J., Steinhardt, J.: Unsolved problems in ml safety. arXiv preprint arXiv:2109.13916 (2021)
14. Hendrycks, D., Gimpel, K.: A baseline for detecting misclassified and out-of-distribution examples in neural networks. arXiv preprint arXiv:1610.02136 (2016)
15. Hendrycks, D., Mazeika, M.: X-risk analysis for ai research. arXiv preprint arXiv:2206.05862 (2022)
16. Hendrycks, D., Mazeika, M., Dietterich, T.: Deep anomaly detection with outlier exposure. arXiv preprint arXiv:1812.04606 (2018)
17. Hornauer, J., Belagiannis, V.: Heatmap-based out-of-distribution detection. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2603–2612 (2023)
18. Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D.: Supervised contrastive learning. Advances in neural information processing systems **33**, 18661–18673 (2020)
19. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
20. Kingma, D.P., Dhariwal, P.: Glow: Generative flow with invertible 1x1 convolutions. Advances in neural information processing systems **31** (2018)

21. Kingma, D.P., Mohamed, S., Jimenez Rezende, D., Welling, M.: Semi-supervised learning with deep generative models. Advances in neural information processing systems **27** (2014)
22. Kirichenko, P., Izmailov, P., Wilson, A.G.: Why normalizing flows fail to detect out-of-distribution data. Advances in neural information processing systems **33**, 20578–20589 (2020)
23. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
24. Lee, K., Lee, K., Lee, H., Shin, J.: A simple unified framework for detecting out-of-distribution samples and adversarial attacks. Advances in neural information processing systems **31** (2018)
25. Liang, S., Li, Y., Srikant, R.: Enhancing the reliability of out-of-distribution image detection in neural networks. arXiv preprint arXiv:1706.02690 (2017)
26. Liu, W., Wang, X., Owens, J., Li, Y.: Energy-based out-of-distribution detection. Advances in neural information processing systems **33**, 21464–21475 (2020)
27. Lu, F., Zhu, K., Zhai, W., Zheng, K., Cao, Y.: Uncertainty-aware optimal transport for semantically coherent out-of-distribution detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3282–3291 (2023)
28. McInnes, L., Healy, J., Melville, J.: Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426 (2018)
29. Mohseni, S., Pitale, M., Yadawa, J., Wang, Z.: Self-supervised learning for generalizable out-of-distribution detection. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 5216–5223 (2020)
30. Morteza, P., Li, Y.: Provable guarantees for understanding out-of-distribution detection. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 7831–7840 (2022)
31. Nalisnick, E., Matsukawa, A., Teh, Y.W., Gorur, D., Lakshminarayanan, B.: Hybrid models with deep and invertible features. In: International Conference on Machine Learning. pp. 4723–4732. PMLR (2019)
32. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning (2011)
33. Papamakarios, G., Nalisnick, E., Rezende, D.J., Mohamed, S., Lakshminarayanan, B.: Normalizing flows for probabilistic modeling and inference. The Journal of Machine Learning Research **22**(1), 2617–2680 (2021)
34. Pidhorskyi, S., Almohsen, R., Doretto, G.: Generative probabilistic novelty detection with adversarial autoencoders. Advances in neural information processing systems **31** (2018)
35. Reyes-González, H., Torre, R.: Testing the boundaries: Normalizing flows for higher dimensional data sets. In: Journal of Physics: Conference Series. vol. 2438, p. 012155. IOP Publishing (2023)
36. Roy, A.G., Ren, J., Azizi, S., Loh, A., Natarajan, V., Mustafa, B., Pawlowski, N., Freyberg, J., Liu, Y., Beaver, Z., et al.: Does your dermatology classifier know what it doesn't know? detecting the long-tail of unseen conditions. Medical Image Analysis **75**, 102274 (2022)
37. Sabokrou, M., Khalooei, M., Fathy, M., Adeli, E.: Adversarially learned one-class classifier for novelty detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3379–3388 (2018)
38. Shafaei, A., Schmidt, M., Little, J.J.: A less biased evaluation of out-of-distribution sample detectors. arXiv preprint arXiv:1809.04729 (2018)
39. Sun, Y., Guo, C., Li, Y.: React: Out-of-distribution detection with rectified activations. Advances in Neural Information Processing Systems **34**, 144–157 (2021)

40. Tabak, E.G., Turner, C.V.: A family of nonparametric density estimation algorithms. Communications on Pure and Applied Mathematics **66**(2), 145–164 (2013)
41. Torralba, A., Fergus, R., Freeman, W.T.: 80 million tiny images: A large data set for nonparametric object and scene recognition. IEEE transactions on pattern analysis and machine intelligence **30**(11), 1958–1970 (2008)
42. Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., Summers, R.M.: Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2097–2106 (2017)
43. Wen, X., Zhao, B., Zheng, A., Zhang, X., Qi, X.: Self-supervised visual representation learning with semantic grouping. Advances in Neural Information Processing Systems **35**, 16423–16438 (2022)
44. Winkens, J., Bunel, R., Roy, A.G., Stanforth, R., Natarajan, V., Ledsam, J.R., MacWilliams, P., Kohli, P., Karthikesalingam, A., Kohl, S., et al.: Contrastive training for improved out-of-distribution detection. arXiv preprint arXiv:2007.05566 (2020)
45. Xu, P., Ehinger, K.A., Zhang, Y., Finkelstein, A., Kulkarni, S.R., Xiao, J.: Turkergaze: Crowdsourcing saliency with webcam based eye tracking. arXiv preprint arXiv:1504.06755 (2015)
46. Yang, J., Wang, H., Feng, L., Yan, X., Zheng, H., Zhang, W., Liu, Z.: Semantically coherent out-of-distribution detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8301–8309 (2021)
47. Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., Xiao, J.: Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv preprint arXiv:1506.03365 (2015)
48. Zagoruyko, S., Komodakis, N.: Wide residual networks. arXiv preprint arXiv:1605.07146 (2016)
49. Zhang, H., Li, A., Guo, J., Guo, Y.: Hybrid models for open set recognition. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16. pp. 102–117. Springer (2020)
50. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: A 10 million image database for scene recognition. IEEE transactions on pattern analysis and machine intelligence **40**(6), 1452–1464 (2017)
51. Zisselman, E., Tamar, A.: Deep residual flow for out of distribution detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13994–14003 (2020)
52. Zong, B., Song, Q., Min, M.R., Cheng, W., Lumezanu, C., Cho, D., Chen, H.: Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In: International conference on learning representations (2018)