CRAFT: Class Ranking Aware Fine-Tuning for Enhanced Out-of-Distribution Detection

Naveen Karunanayake[1] Suranga Seneviratne[1] Sanjay Chawla[2]
[1]The University of Sydney {naveen.karunanayake, suranga.seneviratne}@sydney.edu.au
[2]Qatar Computing Research Institute, HBKU schawla@hbku.edu.qa

Abstract

Out-of-distribution (OOD) detection remains a key challenge preventing the rollout of key AI technologies like autonomous vehicles into the mainstream as classifiers trained on in-distribution (ID) data are unable to gracefully handle OOD data. While OOD detection remains an active area of research, current post-hoc methods often suffer from limited separability between ID and OOD, and outlier exposure-based methods lack generalisation to unseen outlier types. We present CRAFT, a fine-tuning approach for arming pre-trained classifiers against OOD inputs without requiring access to outliers. The key insight that underpins our approach is that during pre-training, classifiers implicitly learn a ranking across the ID classes that is not respected by OOD data. Therefore, a form of fine-tuning without outliers of a pre-trained classifier can sharpen the rank order of the classes, making them sensitive to the presence of OOD data. Furthermore, the fine-tuned model does not impact the ability of the classifier to correctly classify ID inputs to their respective classes. Experiments on CIFAR-10, CIFAR-100, and ImageNet-200 demonstrate that CRAFT outperforms 33 existing methods, particularly in the more challenging near-OOD detection, as well as in overall OOD detection consistency and ID classification accuracy.

1. Introduction

The open world is full of unknowns, posing challenges in safely operating deep neural networks (DNNs) designed to predominantly handle known inputs. These unfamiliar inputs, which notably deviate from the training distribution, are collectively referred to as out-of-distribution (OOD) data. DNNs often exhibit high confidence in such unknown inputs, leading to inaccurate predictions that can degrade their performance and pose serious risks in safety-critical applications such as autonomous driving [9] and healthcare [30]. Therefore, to ensure their reliable operation in the open world, DNNs should be able to identify and avoid making predictions on OOD inputs.

Among the various OOD detection methods, some approaches focus on extracting information from pre-trained models (i.e., post-hoc methods), leveraging their inherent knowledge [13, 22–24]. Other techniques modify the training process to improve differentiation between in-distribution (ID) and OOD data. These methods can be categorised based on whether they use outliers during training (i.e., training methods with outliers) [14, 42, 43, 45] or not (i.e., training methods without outliers) [7, 15, 40]. Each approach has its own strengths and weaknesses, making them effective for OOD detection in specific contexts. Post-hoc methods, for example, are known for their easy implementation and high performance on ID data. In contrast, training methods generally achieve better OOD detection by incorporating OOD awareness during training, though this often comes at the cost of reduced performance on ID data. Additionally, methods that involve outlier exposure [14, 43] are

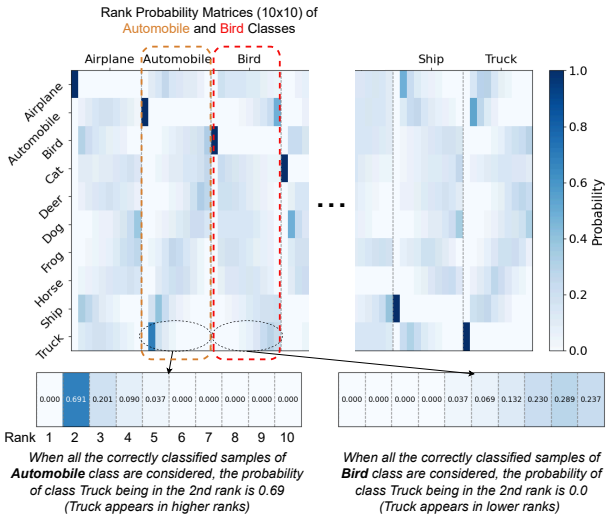


Figure 1. Ranking patterns observed in CIFAR-10 classes. *DNNs implicitly learn to rank ID classes such that semantically related classes are ranked higher and distant classes are ranked lower. CRAFT leverages this observation by performing a fine-tuning on a pre-trained model to refine the class rank order, enhancing its sensitivity to OOD inputs.*

prone to overfitting to the seen outliers during training, resulting in poor performance on unseen outliers.

In this paper, we propose a novel method called ***C**lass **R**anking **A**ware **F**ine-**T**uning* (CRAFT) for OOD detection in image classification settings. The core idea behind CRAFT is that, despite being trained with one-hot encoded ground truths, the high expressivity of DNNs allows them to capture high-level semantic similarities between ID classes [1]. As a result, predicted class ranking patterns are more deterministic for ID data. This behaviour is visualised in Fig. 1 for CIFAR-10 classes. Each ID class is represented as a column in the figure, with each sub-column showing the occurrence of other classes across different ranks. For instance, for the *Automobile* class, the probability of *Truck* appearing in the second rank is 0.69. Given the semantic similarity between *Truck* and *Automobile*, *Truck* frequently appears within the higher ranks for samples from the *Automobile* class. In contrast, for the *Bird* class, the probability of *Truck* appearing in higher ranks is zero. Since *Bird* and *Truck* are not semantically related, *Truck* consistently ranks lower in predictions for the *Bird* class. In CRAFT, we fine-tune the pre-trained model to capture these class-specific ranking patterns and use this information to differentiate between OOD and ID inputs.

Among existing OOD detection methods, CRAFT stands between *post-hoc* and *training methods with outliers*. It leans more towards post-hoc methods since it only requires additional fine-tuning rather than extensive changes to the training process. Furthermore, unlike other training methods—whether they use outliers or not—that often affect classification accuracy on ID classes, CRAFT improves this accuracy while still effectively detecting OOD samples. To summarise, we make the following contributions.

- We propose CRAFT, a novel approach to fine-tune DNNs using class-specific patterns modelled as probability mass functions (PMFs) in subsequent ranks. Additionally, we modify the architecture of pre-trained models to accommodate CRAFT fine-tuning, ensuring that it does not compromise performance on ID data.
- We leverage the fine-tuned model to formulate an OOD score based on the degree of alignment between the predicted PMFs for subsequent ranks and the reference PMFs for those ranks, measured by KL divergence. This score is then used to detect OOD inputs.
- We validate CRAFT through extensive experiments conducted in the OpenOOD environment [46]. Compared to 33 existing OOD detectors, our method consistently ranks among the top five in OOD detection across various scenarios, demonstrating the best overall consistency among the evaluated methods. Additionally, CRAFT excels in near-OOD detection, which is more challenging than far-OOD detection, by achieving an average FPR95 of 46.76%, surpassing all methods that do not require outliers by at least 3%.

The rest of the paper is organised as follows. We present the related work in Sec. 2 and provide the overview of our methodology in Sec. 3. In Sec. 4, we discuss the experimental setup, followed by the results of our experiments in Sec. 5. Finally, Sec. 6 concludes the paper.

2. Related work

Existing OOD detection work can be categorised into three approaches [46] as follows.

2.1. Post-hoc methods

Post-hoc methods [1, 10, 23, 33, 34] use features derived from pre-trained DNNs, such as maximum output probabilities [12,13] and variants of energy [24,44] to assess whether an input is OOD. Some methods transform intermediate features of different DNN layers to capture complex patterns that distinguish ID from OOD [22, 31, 35]. Recently, ExCeL [18] showed that the predicted class rank order could serve as an indicator of OOD-ness, extracting class ranking information from pre-trained models to define a *post-hoc* OOD scoring function. *Building on this, CRAFT leverages the same ranking information differently, modifying the pre-trained model to incorporate this ranking information into its optimisation process. This class ranking-aware fine-tuning enhances OOD detection performance, overcoming ExCeL's limitations.*

2.2. Training methods without outliers

Training methods without outliers include ConfBranch [7], which estimates prediction confidence via an auxiliary branch, and LogitNorm [40], which enforces a constant norm on logits. Other methods like MOS [16] use group-based learning to simplify ID-OOD separation, while CIDER [27] and CSI [36] enhance ID-OOD separability through loss optimisation and contrastive learning. *Unlike these, CRAFT only requires fine-tuning without extensive modifications to the training process, thereby avoiding the drop in ID accuracy often seen with such changes.*

2.3. Training methods with outliers

These methods use auxiliary OOD data during training [14,42,43,45]. For instance, Outlier Exposure (OE) [14] adds a loss term to encourage outlier predictions to follow a uniform distribution, while Mixture Outlier Exposure (MixOE) [45] trains on virtual outliers created by interpolating between ID and outlier data, aiming for a smooth decay of the confidence as the inputs transition from ID to OOD. The labels for these virtual outliers are determined by interpolating between the true ID label with a uniform distribution. While these methods may show strong performance, they often overfit to seen outliers, limiting their

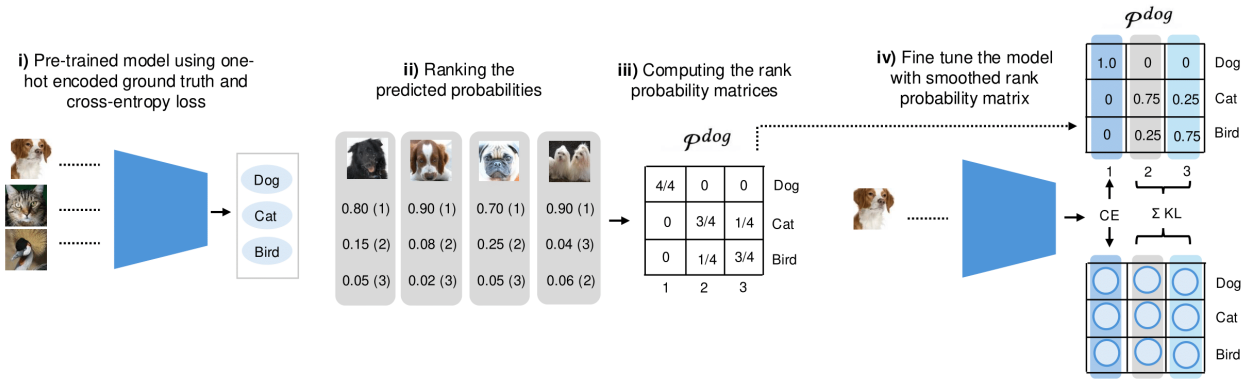


Figure 2. Methodology: (i) CRAFT takes an existing pre-trained model and (ii) extracts the probabilities of training samples to (iii) compute a per-class rank probability matrix (Eq. 1). (iv) This matrix is smoothed to remove ranking noise and used as ground truth to fine-tune the pre-trained model with a loss term that captures both original labelling and rank-derived probability mass functions (Eq. 2). *Note that the model architecture is modified by adding separate neurons to learn the rank PMFs, without disrupting the learning from original labels, thereby preserving ID classification accuracy during fine-tuning.*

generalisability to new OOD types and reducing ID classification accuracy. *In contrast, CRAFT fine-tunes the model without relying on outliers, avoiding such issues.*

3. Methodology

In this section, we explain the methodology behind CRAFT, which consists of two main steps. First, similar to ExCeL [18], we extract class-specific ranking information from a pre-trained model. Second, we modify the architecture of the pre-trained model and fine-tune the model based on the extracted ranking information. The overall methodology of CRAFT is illustrated in Fig. 2.

3.1. Extracting class ranking information

Using the pre-trained model, we filter out the correctly classified training samples for each class and retrieve their corresponding logit vectors. We observe that class prediction probabilities are more distinct in the unnormalised logit space compared to the softmax space. Based on these logit values, we then rank the predicted classes for each training sample. For a given class, we compute the probability mass function (PMF) over the ID classes for each rank, using the rank sequences of samples where the chosen class is the top prediction. We refer to the matrix containing the PMFs of all ranks for a given class as the *rank probability matrix* (RPM), as defined in Eq. 1. Accordingly, for each class c, an element p^c_{ij} in RPM $\mathcal{P}^c \in \mathbb{R}^{C\times C}$ specifies the probability that class i is ranked j^{th} when an input is classified as c. Moreover, each column j corresponds to a PMF over ID classes at rank j.

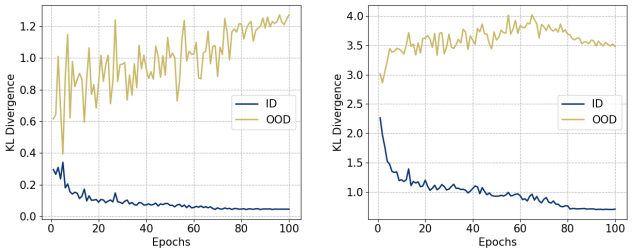
$$\mathcal{P}^c = \begin{pmatrix} p^c_{11} & p^c_{12} & \cdots & p^c_{1C} \\ \vdots & \vdots & \ddots & \vdots \\ p^c_{C1} & p^c_{C2} & \cdots & p^c_{CC} \end{pmatrix}, \quad p^c_{ij} = \frac{n^c_{ij}}{N_c} \qquad (1)$$

Here, n^c_{ij} denotes the number of instances class i is ranked j^{th} among correctly classified samples of class c. N_c represents the total number of correctly classified samples in class c, and C is the total number of ID classes. For any class c, since only correctly classified samples of class c are considered for computing the RPM, the first column of $\mathcal{P}^c$ will be the one-hot encoded vector for class c.

3.2. Fine-tuning the modified model based on class ranking information

Prior to fine-tuning, we adopt a smoothing strategy since the original RPMs may contain noise, and we are primarily interested in significant patterns at subsequent ranks. Therefore, we compare the probabilities of a class appearing at a specific rank with a predefined threshold, τ. If the probability exceeds the threshold, we retain the original values; otherwise, we set them to zero. Next, we renormalise each column in the smoothed RPM to ensure it is a valid PMF again. This is done by dividing each element by the corresponding column sum. For columns where the sum is zero (i.e., no significant classes appearing at that particular rank), we set the distribution as uniform ($\mathcal{U}$). This approach yields a smoother version of the RPM, which we denote as $\hat{\mathcal{P}}$. From this point onward, we will use the term RPM to refer to the smoothed version of the original RPM.

Next, our goal is to leverage the ranking information in RPM to fine-tune the pre-trained model. It is well-known that integrating a secondary objective into the model's loss function to improve the model's OOD awareness often compromises the ID classification accuracy [14]. To mitigate this, we modify the architecture so that the model has separate paths (i.e., neurons) to learn from the one-hot encoded ground truth and the class ranking information. Thus, the model is fine-tuned on a *two-dimensional ground truth*, in



(a) CIFAR-10 (b) CIFAR-100

Figure 3. The behaviour of KL loss on ID and OOD validation sets. Note that the model is fine-tuned only on ID data, with OOD loss included solely to illustrate how KL loss on OOD inputs changes during fine-tuning. *Fine-tuning pushes the KL loss for ID inputs toward zero, while for OOD inputs, it behaves randomly and diverges further from ID data.*

contrast to traditional model training on a one-dimensional one-hot encoded ground truth.

More specifically, we alter the final layer of the model to produce a two-dimensional output where rows represent the ID classes and columns represent prediction ranks. If there are C classes, the modified output layer will be a (C, C) matrix. For example, for CIFAR-10, the normal output layer is of shape (10,1), and our modification changes it to a (10,10) matrix. For fine-tuning, the RPMs are used as the ground truth, with each training sample assigned a class-specific RPM based on its true label. This process is guided by a dual objective, as shown in Eq. 2.

$$\min_{\theta} \quad (1-\alpha)\cdot \mathbb{E}_{(x,y)\sim D_{\text{train}}}\left[\mathcal{L}_{\text{CE}}\left(f_\theta(x)_1, \hat{\mathcal{P}}_1^y\right)\right] + \alpha\cdot \mathbb{E}_{(x,y)\sim D_{\text{train}}}\left[\sum_{k=2}^{C} \text{KL}\left(f_\theta(x)_k \parallel \hat{\mathcal{P}}_k^y\right)\right] \tag{2}$$

Here, $f_\theta(x)_k$ and $\hat{\mathcal{P}}_k^y$ represent the k-th columns of the model output for input x (i.e., $f_\theta(x)$) and the RPM corresponding to the true label y of x (i.e., $\hat{\mathcal{P}}^y$), respectively. The first term minimises the categorical cross-entropy between the first columns of the predicted outputs and the corresponding RPMs. Since the first columns of the RPMs are the one-hot encoded vectors of the true labels (i.e., $\hat{\mathcal{P}}_1^y$), this term continues the standard cross-entropy-based training initially used to train the pre-trained model. Importantly, the second term minimises the Kullback-Leibler (KL) divergence between the predicted PMFs and the true PMFs for the subsequent ranks as specified by the RPMs. The dual objective function ensures that the model learns the true labels of the training samples while preserving the class ranking patterns observed in the subsequent ranks. The hyperparameter α balances the trade-off between the two objectives. Since our primary focus is on learning the class ranking patterns, we set $\alpha = 0.9$ to prioritise the second term. However, to ensure that the model remains aligned with the actual ground truth and does not deviate, we assign a smaller weight to the first term rather than ignoring it.

In Fig. 3, we illustrate the minimisation of KL loss between the predicted rank probabilities and the ground-truth RPMs throughout the epochs of CRAFT fine-tuning. Additionally, to track the performance of the fine-tuned model on OOD inputs, we evaluate and report the KL loss on an OOD validation set. As fine-tuning progresses, an increasing gap between the KL divergence for ID and OOD inputs becomes evident for both CIFAR-10 (Fig. 3a) and CIFAR-100 (Fig. 3b) datasets. This property is leveraged in the CRAFT score to differentiate between ID and OOD inputs.

3.3. Detecting OOD inputs using CRAFT score

Once fine-tuning is completed, we use the updated model for OOD detection based on the premise that the class ranking patterns of ID data will align more closely with the patterns in RPMs compared to OOD data. Therefore, we define the *CRAFT* score to quantify how close the predicted PMFs for subsequent ranks are to the corresponding RPMs. Since the rank predictions for OOD data are highly likely to follow a uniform distribution, in CRAFT, we focus only on the ranks where a prominent class ranking pattern (i.e., non-uniform PMFs in an RPM) exists. Accordingly, for a given test input $\bar{x}$, if the predicted label (i.e., the highest probable class in the first column of the output matrix) is $\bar{y}$, we first extract the columns of the RPM for class $\bar{y}$ with non-uniform PMFs, denoting this set of ranks as $\mathcal{R}^{\bar{y}}$. The CRAFT score is then computed as the mean KL divergence between these non-uniform columns of the RPM and the corresponding columns of the predicted output matrix. Since we only consider the class ranking information from the subsequent ranks, the KL divergence between the first columns of the two matrices is ignored in the CRAFT score computation. We define the CRAFT score in Eq. 3.

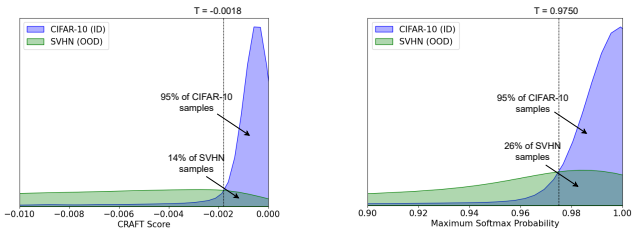
$$CRAFT(\bar{x}, \bar{y}) = -\frac{1}{|\mathcal{R}^{\bar{y}}|}\sum_{k\in\mathcal{R}^{\bar{y}}} \text{KL}\left(f_\theta(\bar{x})_k \| \hat{\mathcal{P}}_k^{\bar{y}}\right) \tag{3}$$

Given that the KL divergences between ID data and RPMs are expected to be lower than those for OOD data, we define the CRAFT score as the negative mean of KL divergences. This aligns with the standard convention of assigning higher scores to ID data.

3.4. Qualitative analysis of CRAFT score

Having defined the CRAFT score as the KL divergence per non-uniform rank (**cf.** Eq. 3), we now analyse its behaviour. To facilitate a better understanding, we compare the CRAFT score with maximum softmax probability (MSP) [13], a well-known, simple baseline for OOD detection. Later, we quantitatively compare CRAFT with more recent and advanced OOD detection methods (**cf.** Sec. 5.1).

First, we present the distribution of CRAFT and MSP scores for the CIFAR-10 (ID) and SVHN [28] (OOD)



(a) CRAFT (b) MSP

Figure 4. The distribution of a) CRAFT and b) MSP scores for CIFAR-10 (ID) and SVHN (OOD) datasets. *The OOD detection thresholds are approximately -0.0018 for CRAFT and 0.9750 for MSP, with FPR95 for SVHN at 14% and 26%, respectively.*

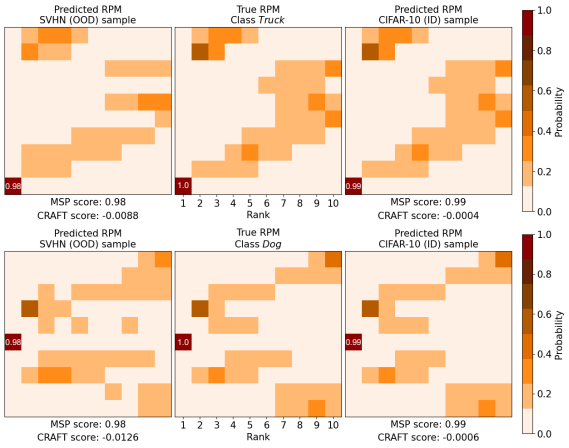


Figure 5. Visualisation of predicted RPMs for OOD samples (left) and ID samples (right) compared to the true RPMs (middle) for the *Truck* (1st row) and *Dog* (2nd row) ID classes. *The two OOD samples have high MSPs of 0.98, making them indistinguishable from ID samples (MSP - 0.99) according to MSP. However, the predicted rank patterns differ for subsequent ranks, enabling CRAFT to effectively differentiate between these ID and OOD samples.*

datasets in Fig. 4. As the KL divergence in Eq. 2 is minimised during fine-tuning (**cf.** Fig. 3), the CRAFT scores for CIFAR-10 samples shift closer to zero compared to those for SVHN samples, as shown in Fig. 4a. The OOD detection thresholds are approximately -0.0018 for CRAFT and 0.9750 for MSP, with FPR95 for SVHN at 14% and 26%, respectively. Next, we analyse the instances where MSP fails, but CRAFT succeeds in OOD detection.

In Fig. 5, we show the predicted RPMs for two SVHN OOD samples misclassified as a Truck (1st row) and a Dog (2nd row) by a model trained on CIFAR-10, each with a high softmax probability of 0.98. Since the OOD detection threshold for MSP is 0.9750 (**cf.** Fig. 4b), these two OOD samples are incorrectly identified as ID by the MSP detector. In contrast, the CRAFT score applies a more rigorous criterion by evaluating the alignment between the predicted PMFs for subsequent ranks, leading to more accurate OOD detection. We observe in Fig. 5 that the predicted RPMs for OOD samples (left) show greater deviation from the true RPMs (middle) than the predicted RPMs for ID samples (right). This difference is reflected in the CRAFT score, where a higher value (i.e., -0.0004 and -0.0006) is obtained for ID samples, while a much lower value (-0.0088 and -0.0126) is observed for OOD samples. In essence, CRAFT assesses whether an input predicted as an ID class conforms to the class ranking patterns observed in subsequent ranks to determine if the input is OOD.

4. Experimental setup

Next, we provide an overview of the datasets, models, and evaluation metrics employed to assess CRAFT's efficacy on common OOD detection benchmarks. We conduct our experiments in OpenOOD[1] environment [46].

4.1. OpenOOD environment

We select OpenOOD because Zhang et al. highlighted that existing OOD detection methods, although they outperform established baselines such as Maximum Softmax Probability (MSP) [13] and Energy [24] in their respective experimental setups, most of them underperform when tested under a standardised benchmarking setup [46]. For instance, methods such as ODIN [23], GradNorm [17], and KLM [12] perform worse than MSP, a naive baseline, when evaluated outside their experimental setup. Therefore, to ensure a fair comparison, we assess CRAFT in the OpenOOD benchmarking framework.

4.2. Datasets

We use CIFAR10, CIFAR100 [20], ImageNet-200 (a.k.a., TinyImageNet) [21] as ID data in our experiments. Each ID dataset is evaluated against both near-OOD and far-OOD datasets. Near-OOD data follow a distribution closer to the ID dataset compared to far-OOD data, making near-OOD detection more challenging than far-OOD detection. For CIFAR-10, the near-OOD datasets are CIFAR-100 and TinyImageNet, whereas for CIFAR-100, CIFAR-10 and TinyImageNet serve as near-OOD. For both CIFAR-10 and CIFAR-100, MNIST [6], SVHN [28], Textures [5], and Places365 [47] are considered as far-OOD. Similarly, for ImageNet-200, SSB-hard [38] and NINCO [2] datasets are used as near-OOD, while iNaturalist [37], Textures [5], and OpenImage-O [39] datasets are used as far-OOD. For consistency, we adopt the same train, validation, and test splits used by the OpenOOD benchmark.

4.3. Comparison with baselines

We evaluate CRAFT against 33 existing OOD detectors, spanning various approaches: *post-hoc methods*, *training methods without outliers*, and *training methods with outliers*. Among the post-hoc methods, we include early base-

[1] https://github.com/Jingkang50/OpenOOD

Table 1. Performance comparison in *near-OOD detection*. For each column, the top five methods are marked in **bold**.

Method	CIFAR-10		CIFAR-100		ImageNet-200		Average	
	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓
Post-hoc inference methods								
MSP [13]	88.03 ± 0.25	48.17 ± 3.92	80.27 ± 0.11	**54.80 ± 0.33**	83.34 ± 0.06	**54.82 ± 0.35**	83.88	52.60
TempScale [10]	88.09 ± 0.31	50.96 ± 4.32	80.90 ± 0.07	**54.49 ± 0.48**	**83.69 ± 0.04**	**54.82 ± 0.23**	**84.23**	53.42
RMDS [29]	89.80 ± 0.28	38.89 ± 2.39	80.15 ± 0.11	55.46 ± 0.41	82.57 ± 0.25	**54.02 ± 0.58**	84.17	**49.46**
EBO [24]	87.58 ± 0.46	61.34 ± 4.63	**80.91 ± 0.08**	55.62 ± 0.61	82.50 ± 0.05	60.24 ± 0.57	83.66	59.07
ReAct [33]	87.11 ± 0.61	63.56 ± 7.33	80.77 ± 0.05	56.39 ± 0.34	81.87 ± 0.98	62.49 ± 2.19	83.25	60.81
MLS [12]	87.52 ± 0.47	61.32 ± 4.62	**81.05 ± 0.07**	55.47 ± 0.66	82.90 ± 0.04	59.76 ± 0.59	83.82	58.85
VIM [39]	88.68 ± 0.28	44.84 ± 2.31	74.98 ± 0.13	62.63 ± 0.27	78.68 ± 0.24	59.19 ± 0.71	80.78	55.55
KNN [35]	90.64 ± 0.20	34.01 ± 0.38	80.18 ± 0.15	61.22 ± 0.14	81.57 ± 0.17	60.18 ± 0.52	84.13	51.80
ASH [8]	75.27 ± 1.04	86.78 ± 1.82	78.20 ± 0.15	65.71 ± 0.24	82.38 ± 0.19	64.89 ± 0.90	78.62	72.46
GEN [25]	88.20 ± 0.30	53.67 ± 3.14	**81.31 ± 0.08**	**54.42 ± 0.33**	**83.68 ± 0.06**	55.20 ± 0.20	**84.40**	54.43
ExCeL [18]	86.89 ± 0.23	66.55 ± 0.43	80.70 ± 0.06	55.21 ± 0.56	82.40 ± 0.04	57.90 ± 0.40	83.33	59.89
Training methods without outliers								
CRAFT (Ours)	**91.11 ± 0.04**	**31.94 ± 1.41**	80.90 ± 0.33	**53.73 ± 0.62**	**83.65 ± 0.41**	**54.62 ± 0.57**	**85.22**	**46.76**
ConfBranch [7]	89.84 ± 0.24	**31.28 ± 0.66**	71.60 ± 0.62	70.21 ± 0.83	79.10 ± 0.24	61.44 ± 0.34	80.18	54.31
G-ODIN [15]	89.12 ± 0.57	45.54 ± 2.52	77.15 ± 0.28	67.58 ± 0.98	77.28 ± 0.10	69.87 ± 0.46	81.18	61.00
LogitNorm [40]	**92.33 ± 0.08**	**29.34 ± 0.81**	78.47 ± 0.31	62.89 ± 0.57	82.66 ± 0.15	56.46 ± 0.37	**84.49**	**49.56**
CIDER [27]	**90.71 ± 0.16**	32.11 ± 0.94	73.10 ± 0.39	72.02 ± 0.31	80.58 ± 1.75	60.10 ± 0.73	81.46	54.74
Training methods with outliers								
OE [14]	**94.82 ± 0.21**	**19.84 ± 0.95**	**88.30 ± 0.10**	**30.73 ± 0.11**	**84.84 ± 0.16**	**52.30 ± 0.67**	**89.32**	**34.29**
MCD [43]	**91.03 ± 0.12**	**30.17 ± 0.06**	77.07 ± 0.32	55.88 ± 0.85	**83.62 ± 0.09**	**54.71 ± 0.83**	83.91	**46.92**
UDG [42]	89.91 ± 0.25	35.34 ± 0.95	78.02 ± 0.10	61.42 ± 0.48	74.30 ± 1.63	68.89 ± 1.72	80.74	55.22
MixOE [45]	88.73 ± 0.82	51.45 ± 7.78	**80.95 ± 0.20**	55.22 ± 0.49	82.62 ± 0.03	57.97 ± 0.40	84.10	54.88

Table 2. Performance comparison in *far-OOD detection*. For each column, the top five methods are marked in **bold**.

Method	CIFAR-10		CIFAR-100		ImageNet-200		Average	
	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓
Post-hoc inference methods								
MSP [13]	90.73 ± 0.43	31.72 ± 1.84	77.76 ± 0.44	58.70 ± 1.06	90.13 ± 0.09	35.43 ± 0.38	86.21	41.95
TempScale [10]	90.97 ± 0.52	33.48 ± 2.39	78.74 ± 0.51	57.94 ± 1.14	90.82 ± 0.09	34.00 ± 0.37	86.84	41.81
RMDS [29]	92.20 ± 0.21	25.35 ± 0.73	**82.92 ± 0.42**	**52.81 ± 0.63**	88.06 ± 0.34	32.45 ± 0.79	87.73	36.87
EBO [24]	91.21 ± 0.92	41.69 ± 5.32	79.77 ± 0.61	56.59 ± 1.38	90.86 ± 0.21	34.86 ± 1.30	87.28	44.38
ReAct [33]	90.42 ± 1.41	44.90 ± 8.37	80.39 ± 0.49	54.20 ± 1.56	**92.31 ± 0.56**	28.50 ± 0.95	87.71	42.53
MLS [12]	91.10 ± 0.89	41.68 ± 5.27	79.67 ± 0.57	56.73 ± 1.33	91.11 ± 0.19	34.03 ± 1.21	87.29	44.15
VIM [39]	93.48 ± 0.24	25.05 ± 0.52	81.70 ± 0.62	**50.74 ± 1.00**	91.26 ± 0.19	**27.20 ± 0.30**	**88.81**	**34.33**
KNN [35]	92.96 ± 0.14	24.27 ± 0.40	**82.40 ± 0.17**	53.65 ± 0.28	**93.16 ± 0.22**	**27.27 ± 0.75**	**89.51**	35.06
ASH [8]	78.49 ± 2.58	79.03 ± 4.22	80.58 ± 0.66	59.20 ± 2.46	**93.90 ± 0.27**	**27.29 ± 1.12**	84.32	55.17
GEN [25]	91.35 ± 0.69	34.73 ± 1.58	79.68 ± 0.75	56.71 ± 1.59	91.36 ± 0.10	32.10 ± 0.59	87.46	41.18
ExCeL [18]	91.69 ± 0.18	40.03 ± 0.84	**82.04 ± 0.90**	**52.24 ± 1.90**	91.97 ± 0.27	**28.45 ± 0.80**	88.57	40.24
Training methods without outliers								
CRAFT (Ours)	93.94 ± 0.20	**19.40 ± 0.88**	**82.03 ± 0.34**	**51.86 ± 0.49**	90.88 ± 0.89	32.67 ± 1.13	**88.95**	**34.64**
ConfBranch [7]	92.85 ± 0.29	21.48 ± 0.94	68.90 ± 1.83	71.82 ± 3.39	90.43 ± 0.18	34.75 ± 0.63	84.06	42.68
G-ODIN [15]	**95.51 ± 0.31**	21.45 ± 1.91	**85.67 ± 1.58**	**42.68 ± 3.19**	**92.33 ± 0.11**	30.18 ± 0.49	**91.17**	**31.44**
LogitNorm [40]	**96.74 ± 0.06**	**13.81 ± 0.20**	81.53 ± 1.26	53.61 ± 3.45	**93.04 ± 0.21**	**26.11 ± 0.52**	**90.44**	**31.18**
CIDER [27]	**94.71 ± 0.36**	**20.72 ± 0.85**	80.49 ± 0.68	54.22 ± 1.24	90.66 ± 1.68	30.17 ± 2.75	88.62	35.04
Training methods with outliers								
OE [14]	**96.00 ± 0.13**	**13.13 ± 0.53**	81.41 ± 1.49	54.82 ± 2.79	89.02 ± 0.18	34.17 ± 0.56	**88.81**	**34.04**
MCD [43]	91.00 ± 1.10	32.03 ± 4.21	74.72 ± 0.78	54.39 ± 1.34	88.94 ± 0.10	29.93 ± 0.30	84.89	38.78
UDG [42]	**94.06 ± 0.90**	**20.35 ± 2.41**	79.59 ± 1.77	59.00 ± 3.35	82.09 ± 2.78	62.04 ± 5.99	85.25	47.13
MixOE [45]	91.93 ± 0.69	33.84 ± 4.77	76.40 ± 1.44	63.88 ± 2.48	88.27 ± 0.41	40.93 ± 0.29	85.53	46.22

lines such as MSP [13] and Energy-based OOD detection (EBO) [24], as well as more recent approaches like ReAct [33], ASH [8], and GEN [25]. Training methods that do not utilise auxiliary outlier data include ConfBranch [7], G-ODIN [15], and LogitNorm [40]. Conversely, methods that leverage outlier data for training are Outlier Exposure (OE) [14], MCD [43], UDG [42], and MixOE [45].

4.4. Models

We use pre-trained ResNet-18 [11] models as the backbone for our experiments. Each model was trained for 100 epochs using standard cross-entropy loss with the SGD optimiser having a momentum of 0.9, a learning rate of 0.1, and a cosine annealing decay schedule [26]. We use the same training configuration when fine-tuning with CRAFT.

4.5. Evaluation metrics

We use two metrics to assess the performance of OOD detectors: i) False Positive Rate at 95% True Positive Rate (FPR95), which quantifies the rate at which OOD samples are incorrectly classified as ID when the true positive rate for ID samples is at 95%; and ii) Area Under the Receiver Operating Characteristic Curve (AUROC), which measures the classifier's ability to distinguish between ID and OOD samples across various thresholds. A good OOD detector should demonstrate both a low FPR95 and a high AUROC. Moreover, we report the mean and the standard deviation of these metrics computed over three independent runs.

5. Results and analysis

We compare the performance of various methods on near-OOD detection in Tab. 1 and far-OOD detection in Tab. 2. Due to space constraints, we provide the average performance for both near and far-OOD across all benchmarks for each ID dataset (**cf.** Sec. 4.2), along with the overall OOD detection performance. Also, in the main text, we report results only for baselines that appear at least once among the top five in an experiment scenario. Results for all 33 methods, along with the dataset-wise results, are available in the Appendix in Supplementary Materials.

5.1. Comparison of OOD detection methods

As observed in Tab. 1, CRAFT exhibits the second-best performance in near-OOD detection, achieving an average AUROC of 85.22 and an FPR95 of 46.76 across three ID datasets. The only method that surpasses CRAFT is Outlier Exposure (OE), which leverages the knowledge of outliers during training. However, we later show that outlier-based methods are highly sensitive to the specific outliers encountered during training, and the high performance of OE in near-OOD detection is, in fact, a result of a bias towards these seen outliers (**cf.** Sec. 5.4). Consequently, among the methods that do not rely on outliers, CRAFT achieves the best performance in near-OOD detection, showing a 3% improvement in FPR95 on average compared to the next best-performing method, RMDS [29].

In far-OOD detection, CRAFT ranks fourth in AUROC with an average of 88.95 and fifth in FPR95 with an average of 34.64, as shown in Tab. 2. LogitNorm [40] and G-ODIN [15] have the highest performance here, surpassing CRAFT by approximately 3% in terms of FPR95. Despite this, CRAFT outperforms LogitNorm on CIFAR-100 and G-ODIN on CIFAR-10. However, we highlight that both LogitNorm and G-ODIN perform worse than CRAFT in near-OOD detection, which is more challenging due to the closer semantic similarity of near-OOD samples to ID data. Notably, on CIFAR-100 and ImageNet-200, CRAFT outperforms OE, which uses outliers for training.

In terms of overall consistency, as indicated by the number of times each method appears in the top five, CRAFT exhibits the shared best performance with OE with 12 out of 16 appearances (marked in **bold** in Tab. 1 and 2). LogitNorm follows, with 10 out of 16 appearances, while G-ODIN ranks third with 6 out of 16 appearances.

5.2. Impact on ID accuracy

Another strength of CRAFT is its ability to improve ID classification performance even during fine-tuning, unlike other training methods (with or without outliers), where ID performance tends to degrade. This advantage stems from CRAFT's architectural modifications prior to fine-tuning that aligns the rank patterns while keeping the top prediction intact. Specifically, CRAFT uses a separate set of neurons in the output layer to learn from class ranking patterns without interfering with the learning from the one-hot encoded true labels (**cf.** Fig. 2). This is confirmed by the results in Tab. 3, which show the ID classification performance of various methods across three ID datasets. Since all post-hoc methods rely on pre-trained models for OOD detection, we present their ID performance collectively.

Table 3. ID classification accuracy (%). *CRAFT achieves the highest accuracy in all datasets, whereas other methods show a performance drop compared to the baseline post-hoc accuracy.* Note that N/A indicates that results are not reported in OpenOOD.

Method	CIFAR-10	CIFAR-100	ImageNet-200
Post-hoc	95.06 ± 0.30	77.25 ± 0.10	86.37 ± 0.08
ConfBranch [7]	94.88 ± 0.05	76.59 ± 0.27	85.92 ± 0.07
G-ODIN [15]	94.70 ± 0.25	74.46 ± 0.04	84.56 ± 0.28
CSI [36]	91.16 ± 0.14	61.60 ± 0.46	N/A
ARPL [4]	93.66 ± 0.11	70.70 ± 1.08	83.95 ± 0.32
MOS [16]	94.83 ± 0.37	76.98 ± 0.20	85.60 ± 0.20
LogitNorm [40]	94.30 ± 0.25	76.34 ± 0.17	86.04 ± 0.15
OE [14]	94.63 ± 0.26	76.84 ± 0.42	85.82 ± 0.21
MCD [43]	94.95 ± 0.04	75.83 ± 0.04	86.12 ± 0.17
UDG [42]	92.36 ± 0.84	71.54 ± 0.64	68.11 ± 1.24
MixOE [45]	94.55 ± 0.32	75.13 ± 0.06	85.71 ± 0.07
CRAFT (Ours)	**95.80 ± 0.13**	**78.43 ± 0.09**	**86.71 ± 0.08**

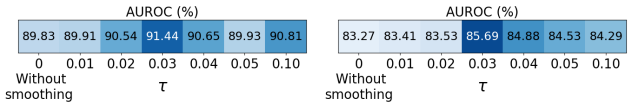
As can be seen from Tab. 3, CRAFT achieves the highest ID accuracy across all three datasets, with 95.8% on CIFAR-10, 78.43% on CIFAR-100, and 86.71% on ImageNet-200. In contrast, other training methods exhibit at least a slight drop in ID accuracy compared to post-hoc methods, with some, such as CSI [36] and UDG [42], demonstrating a more significant drop.

5.3. The ablation study on smoothing

Smoothing the RPM to eliminate noisy class occurrences in subsequent ranks is a key step in CRAFT (**cf.** Sec. 3.2). Here, we analyse the impact of the smoothing threshold, τ.

In prior OOD detection work, hyperparameters are often tuned on a validation set [14, 24, 35]. Following this approach, we determine CRAFT's smoothing threshold, τ, using the validation set provided by OpenOOD for each ID dataset. Therefore, through a grid search, we identified the best-performing value for τ as 0.03 for both CIFAR-10 and CIFAR-100. For ImageNet-200, the best value for τ was 0.025. The change of AUROC across different τ values for CIFAR-10 and CIFAR-100 is illustrated in Fig. 6.

Incorporating smoothing (i.e., $\tau > 0$) improves OOD detection performance, as shown in both figures. For instance, in CIFAR-10, the AUROC on the validation set without smoothing (i.e., $\tau = 0$) is 89.8%, while in CIFAR-100, it is 83.3%. By adopting a smoothing strategy, the OOD detection performance can improve up to 91.4% in CIFAR-10 and 85.7% in CIFAR-100 (with $\tau = 0.03$). A similar trend can be observed in ImageNet-200 as well. As τ increases, OOD detection performance initially improves due to the removal of noisy class occurrences. However, after a certain point, further increases in τ lead to a drop in performance



(a) CIFAR-10 (b) CIFAR-100

Figure 6. Ablation study on smoothing. *Smoothing enhances performance by removing noisy class occurrences in subsequent ranks up to a certain point. However, further increasing τ eliminates important rank patterns, dropping the performance.*

because important rank occurrences in the RPMs are also removed, resulting in information loss. Consequently, the AUROC decreases once τ surpasses this threshold.

5.4. Impact of auxiliary training outliers

In Tab. 1, we observed that the outlier exposure (OE) method outperformed all its competitors in near-OOD detection. However, we attribute this performance advantage as an artefact of bias towards the outliers seen during training. For all outlier-based methods (i.e., OE, MCD, UDG, and MixOE), OpenOOD employs an auxiliary outlier training set derived from non-overlapping classes within ImageNet-1K. Since ImageNet-200 comprises the first 200 classes of ImageNet-1K, the outliers are drawn from the remaining 800 classes. Moreover, due to class overlap with CIFAR-10 and CIFAR-100, this number further reduces to 597 classes, forming the tin597 [46] outlier dataset.

Consequently, during OE [14] training for both CIFAR-10 and CIFAR-100, the tin597 dataset is used as auxiliary outliers. This results in a bias towards the rest of the ImageNet data (e.g. ImageNet-200) due to high visual similarity. Therefore, OE performs significantly well on ImageNet-200 (refer to the *OE* row in Tab. 7 and 9 in the Appendix) compared to other OOD datasets, reflecting a significant improvement in near OOD detection in Tab. 1.

To demonstrate this, we tested the OE method using NINCO [2], notMNIST [3], and FashionMNIST [41] as training outliers, which are visually distinct from tin597 dataset. We show the results in Tab. 4. Due to space constraints, we report only the average near and far-OOD FPR95. Full results are provided in Tab. 13 in the Appendix.

In Tab. 4, we observe that FPR95 has worsened across all auxiliary outlier datasets compared to when tin597 is used as outliers. Notably, when visually distinct datasets such as notMNIST and FashionMNIST are used as outliers for training, the performance on near-OOD detection drops significantly, with FPR95 increasing from 19% to as high as 48% on CIFAR-10 and from 30% to 56% on CIFAR-100. However, in this scenario, the performance on far-OOD detection is either slightly affected or even shows improvement. This can be attributed to a similar bias, this time with the MNIST dataset, which is considered far-OOD for both CIFAR-10 and CIFAR-100. For example, when notMNIST or FashionMNIST is used as outliers to train an OE model on CIFAR-10 or CIFAR-100, the FPR95 on MNIST drops to 0% (Tab. 13 in the Appendix), improving the overall performance on far-OOD detection. *Therefore, although OE-based methods show improved performance, they lack generalisability due to their bias towards outliers seen during training. In contrast, CRAFT's OOD detection performance does not rely on outliers during fine-tuning.*

Table 4. FPR95 variation of Outlier Exposure (OE) method across various auxiliary outlier training data. *Here, we include CRAFT's performance to show that it does not rely on outliers during training. Furthermore, CRAFT represents all methods whose OOD detection performance is independent of training outliers.*

ID dataset	Auxiliary outlier data	OE		CRAFT	
		Near	Far	Near	Far
CIFAR-10	tin597 [46]	19.84	13.13	31.94	19.40
	NINCO [2]	30.56	19.94	31.94	19.40
	FashionMNIST [41]	38.57	22.88	31.94	19.40
	notMNIST [3]	48.69	27.22	31.94	19.40
CIFAR-100	tin597 [46]	30.73	54.82	53.73	51.86
	NINCO [2]	49.74	40.88	53.73	51.86
	FashionMNIST [41]	56.24	41.38	53.73	51.86
	notMNIST [3]	56.36	42.33	53.73	51.86

5.5. Comparison with ExCeL

Using class rank information for OOD detection was initially explored in ExCeL [18]. However, as a post-hoc method, ExCeL doesn't perform well in near-OOD detection and in cases with limited class diversity, such as the CIFAR-10 ID setting. *CRAFT leverages class ranking information more effectively using fine-tuning to achieve improved performance, addressing both limitations of ExCeL.* In terms of FPR95, CRAFT shows an average improvement of 13% in near-OOD detection and 6% in far-OOD detection compared to ExCeL. Additionally, in CIFAR-10 ID setting, CRAFT achieves a significant performance boost of 34% and 20% for near- and far-OOD detection over ExCeL.

6. Conclusion

We proposed a novel fine-tuning approach for OOD detection, using class ranking patterns implicitly learned by DNNs during pre-training. For each class, we modelled ranks as PMFs over ID classes, creating a unique RPM that serves as ground truth for fine-tuning. We then defined the CRAFT score, which quantifies the alignment between predicted and reference PMFs, and used this score to detect OOD inputs. Experiments on CIFAR-10, CIFAR-100, and ImageNet-200 showed that CRAFT outperforms 33 baselines, particularly in near-OOD detection, overall detection consistency, and ID classification accuracy.

Acknowledgment

This research was supported by the Australian Government through the Australian Research Council's Discovery Projects funding scheme (Project ID - DP220102520).

References

[1] Abhijit Bendale et al. Towards open set deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1563–1572, 2016. 2, 12, 13, 14, 15, 16, 17, 18, 19

[2] Julian Bitterwolf, Maximilian Müller, and Matthias Hein. In or out? fixing imagenet out-of-distribution detection evaluation. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023. 5, 8, 19

[3] Yaroslav Bulatov. Notmnist dataset. *Google (Books/OCR), Tech. Rep.[Online]. Available: http://yaroslavvb. blogspot. it/2011/09/notmnist-dataset. html*, 2, 2011. 8, 19

[4] Guangyao Chen, Peixi Peng, Xiangqian Wang, and Yonghong Tian. Adversarial reciprocal points learning for open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):8065–8081, 2021. 7, 12, 13, 14, 15, 16, 17, 18, 19

[5] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3606–3613, 2014. 5

[6] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. 5

[7] Terrance DeVries et al. Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint arXiv:1802.04865*, 2018. 1, 2, 6, 7, 12, 13, 14, 15, 16, 17, 18, 19

[8] Andrija Djurisic, Nebojsa Bozanic, Arjun Ashok, and Rosanne Liu. Extremely simple activation shaping for out-of-distribution detection. In *The Eleventh International Conference on Learning Representations*, 2023. 6, 12, 13, 14, 15, 16, 17, 18, 19

[9] Angelos Filos, Panagiotis Tigkas, Rowan McAllister, Nicholas Rhinehart, Sergey Levine, and Yarin Gal. Can autonomous vehicles identify, recover from, and adapt to distribution shifts? In *International Conference on Machine Learning*, pages 3145–3153. PMLR, 2020. 1

[10] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1321–1330. JMLR.org, 2017. 2, 6, 12, 13, 14, 15, 16, 17, 18, 19

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 6

[12] Dan Hendrycks, Steven Basart, Mantas Mazeika, Andy Zou, Joseph Kwon, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. Scaling out-of-distribution detection for real-world settings. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 8759–8773. PMLR, 2022. 2, 5, 6, 12, 13, 14, 15, 16, 17, 18, 19

[13] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2017. 1, 2, 4, 5, 6, 12, 13, 14, 15, 16, 17, 18, 19

[14] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations*, 2019. 1, 2, 3, 6, 7, 8, 11, 12, 13, 14, 15, 16, 17, 18, 19

[15] Yen-Chang Hsu, Yilin Shen, Hongxia Jin, and Zsolt Kira. Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10951–10960, 2020. 1, 6, 7, 12, 13, 14, 15, 16, 17, 18, 19

[16] Rui Huang et al. Mos: Towards scaling out-of-distribution detection for large semantic space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8710–8719, 2021. 2, 7, 12, 13, 14, 15, 16, 17, 18, 19

[17] Rui Huang et al. On the importance of gradients for detecting distributional shifts in the wild. *Advances in Neural Information Processing Systems*, 34:677–689, 2021. 5, 12, 13, 14, 15, 16, 17, 18, 19

[18] Naveen Karunanayake, Suranga Seneviratne, and Sanjay Chawla. Excel: Combined extreme and collective logit information for enhancing out-of-distribution detection. *arXiv preprint arXiv:2311.14754*, 2023. 2, 3, 6, 8, 12, 13, 14, 15, 16, 17, 18, 19

[19] Shu Kong et al. Opengan: Open-set recognition via open data generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 813–822, 2021. 12, 13, 14, 15, 16, 17, 18, 19

[20] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[21] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 5

[22] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31, 2018. 1, 2, 12, 13, 14, 15, 16, 17, 18, 19

[23] Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*, 2018. 1, 2, 5, 12, 13, 14, 15, 16, 17, 18, 19

[24] Weitang Liu, Xiaoyun Wang, John D. Owens, and Yixuan Li. Energy-based out-of-distribution detection. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. 1, 2, 5, 6, 7, 12, 13, 14, 15, 16, 17, 18, 19

[25] Xixi Liu et al. Gen: Pushing the limits of softmax-based out-of-distribution detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23946–23955, 2023. 6, 12, 13, 14, 15, 16, 17, 18, 19

[26] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017. 6

[27] Yifei Ming, Yiyou Sun, Ousmane Dia, and Yixuan Li. How to exploit hyperspherical embeddings for out-of-distribution detection? In *The Eleventh International Conference on Learning Representations*, 2023. 2, 6, 12, 13, 14, 15, 16, 17, 18, 19

[28] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. 4, 5

[29] Jie Ren, Stanislav Fort, Jeremiah Liu, Abhijit Guha Roy, Shreyas Padhy, and Balaji Lakshminarayanan. A simple fix to mahalanobis distance for improving near-ood detection. *arXiv preprint arXiv:2106.09022*, 2021. 6, 7, 12, 13, 14, 15, 16, 17, 18, 19

[30] Abhijit Guha Roy, Jie Ren, Shekoofeh Azizi, Aaron Loh, Vivek Natarajan, Basil Mustafa, Nick Pawlowski, Jan Freyberg, Yuan Liu, Zach Beaver, 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. 1

[31] Chandramouli Shama Sastry et al. Detecting out-of-distribution examples with gram matrices. In *International Conference on Machine Learning*, pages 8491–8501. PMLR, 2020. 2, 12, 14, 15, 16, 17, 18, 19

[32] Yue Song et al. Rankfeat: Rank-1 feature removal for out-of-distribution detection. *Advances in Neural Information Processing Systems*, 35:17885–17898, 2022. 12, 13, 14, 15, 16, 17, 18, 19

[33] Yiyou Sun et al. React: Out-of-distribution detection with rectified activations. *Advances in Neural Information Processing Systems*, 34:144–157, 2021. 2, 6, 12, 13, 14, 15, 16, 17, 18, 19

[34] Yiyou Sun et al. Dice: Leveraging sparsification for out-of-distribution detection. In *European Conference on Computer Vision*, pages 691–708. Springer, 2022. 2, 12, 13, 14, 15, 16, 17, 18, 19

[35] Yiyou Sun, Yifei Ming, Xiaojin Zhu, and Yixuan Li. Out-of-distribution detection with deep nearest neighbors. In *International Conference on Machine Learning*, pages 20827–20840. PMLR, 2022. 2, 6, 7, 12, 13, 14, 15, 16, 17, 18, 19

[36] Jihoon Tack, Sangwoo Mo, Jongheon Jeong, and Jinwoo Shin. Csi: Novelty detection via contrastive learning on distributionally shifted instances. *Advances in Neural Information Processing Systems*, 33:11839–11852, 2020. 2, 7, 12, 13, 14, 15, 16, 17, 18, 19

[37] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8769–8778, 2018. 5

[38] Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Open-set recognition: A good closed-set classifier is all you need. In *International Conference on Learning Representations*, 2022. 5

[39] Haoqi Wang, Zhizhong Li, Litong Feng, and Wayne Zhang. Vim: Out-of-distribution with virtual-logit matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4921–4930, 2022. 5, 6, 12, 13, 14, 15, 16, 17, 18, 19

[40] Hongxin Wei, Renchunzi Xie, Hao Cheng, Lei Feng, Bo An, and Yixuan Li. Mitigating neural network overconfidence with logit normalization. In *International Conference on Machine Learning*, pages 23631–23644. PMLR, 2022. 1, 2, 6, 7, 12, 13, 14, 15, 16, 17, 18, 19

[41] Han Xiao et al. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017. 8, 19

[42] Jingkang Yang, Haoqi Wang, Litong Feng, Xiaopeng Yan, Huabin Zheng, Wayne Zhang, and Ziwei Liu. Semantically coherent out-of-distribution detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8301–8309, 2021. 1, 2, 6, 7, 12, 13, 14, 15, 16, 17, 18, 19

[43] Qing Yu et al. Unsupervised out-of-distribution detection by maximum classifier discrepancy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9518–9526, 2019. 1, 2, 6, 7, 12, 13, 14, 15, 16, 17, 18, 19

[44] Jinsong Zhang, Qiang Fu, Xu Chen, Lun Du, Zelin Li, Gang Wang, Shi Han, Dongmei Zhang, et al. Out-of-distribution detection based on in-distribution data patterns memorization with modern hopfield energy. In *The Eleventh International Conference on Learning Representations*, 2022. 2, 12, 13, 14, 15, 16, 17, 18, 19

[45] Jingyang Zhang, Nathan Inkawhich, Randolph Linderman, Yiran Chen, and Hai Li. Mixture outlier exposure: Towards out-of-distribution detection in fine-grained environments. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5531–5540, 2023. 1, 2, 6, 7, 12, 13, 14, 15, 16, 17, 18, 19

[46] Jingyang Zhang, Jingkang Yang, Pengyun Wang, Haoqi Wang, Yueqian Lin, Haoran Zhang, Yiyou Sun, Xuefeng Du, Kaiyang Zhou, Wayne Zhang, et al. Openood v1. 5: Enhanced benchmark for out-of-distribution detection. *arXiv preprint arXiv:2306.09301*, 2023. 2, 5, 8, 19

[47] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2017. 5