Out-of-Distribution Detection with Prototypical Outlier Proxy

Mingrong Gong[1], Chaoqi Chen[1*], Qingqiang Sun[2], Yue Wang[3], Hui Huang[1]

[1]College of Computer Science and Software Engineering, Shenzhen University
[2]School of Engineering, Great Bay University
[3]Department of Computer Science, University College London
{gmr52333, cqchen1994, hhzhiyan}@gmail.com, qqsun@gbu.edu.cn, zcabwaa@ucl.ac.uk

Abstract

Out-of-distribution (OOD) detection is a crucial task for deploying deep learning models in the wild. One of the major challenges is that well-trained deep models tend to perform over-confidence on unseen test data. Recent research attempts to leverage real or synthetic outliers to mitigate the issue, which may significantly increase computational costs and be biased toward specific outlier characteristics. In this paper, we propose a simple yet effective framework, *Prototypical Outlier Proxy* (POP), which introduces virtual OOD prototypes to reshape the decision boundaries between ID and OOD data. Specifically, we transform the learnable classifier into a fixed one and augment it with a set of prototypical weight vectors. Then, we introduce a hierarchical similarity boundary loss to impose adaptive penalties depending on the degree of misclassification. Extensive experiments across various benchmarks demonstrate the effectiveness of POP. Notably, POP achieves average FPR95 reductions of 7.70%, 6.30%, and 5.42% over the second-best methods on CIFAR-10, CIFAR-100, and ImageNet-200, respectively. Moreover, compared to the recent method NPOS, which relies on outlier synthesis, POP trains 7.2 times faster and performs inference 19.5 times faster.

Introduction

Deep learning models have achieved remarkable success across various tasks such as image classification (He et al. 2016), face recognition (Deng et al. 2019), and object detection (He et al. 2017). However, the safety requirements of these models pose significant challenges when deployed in real-world applications, such as autonomous driving (Chen et al. 2023b), robotics (Levine et al. 2016), and medical diagnostics (Amodei et al. 2016). Albeit the extraordinary performance on in-distribution (ID) data, such models struggle to deal with out-of-distribution (OOD) data, which may result in misclassifications, misguided decisions, and even catastrophe. As shown in Fig. 1 (left), to achieve higher training accuracy, deep models tend to make overconfident predictions (Guo et al. 2017), even in the low-density regions. To solve this issue, many existing methods strive to directly introduce outliers to enhance the unknown-aware ability during the training phase, using either real outlier data, *i.e.,* outlier exposure (OE) (Hendrycks, Mazeika, and Dietterich 2019; Yu and Aizawa 2019; Yang et al. 2021; Ming, Fan, and Li 2022; Zhang et al. 2023b), or feature-based outlier synthesis (Pei et al. 2022; Du et al. 2022; Tao et al. 2023). Fig. 1 (middle) shows that training with outliers will create a specific region to accommodate potential OOD data. Despite the promise, these methods may still be constrained by two bottlenecks: (i) Incorporating extra outliers in the training phase can be time-consuming and resource-intensive. For example, synthesizing outliers requires density estimation (parametric (Du et al. 2022) or non-parametric (Tao et al. 2023)) of ID data first. (ii) In practice, OOD data are diverse and typically distribution-free (Fang et al. 2022). Thus, OE methods may only be effective in certain specific domains because it is impossible to cover all potential scenarios. These methods may cause the model to be biased towards specific outlier characteristics, leading to a loss of generality. For instance, the model might learn spurious correlations between the data and binary labels (Ming, Yin, and Li 2022). To this premise, we raise an open question:

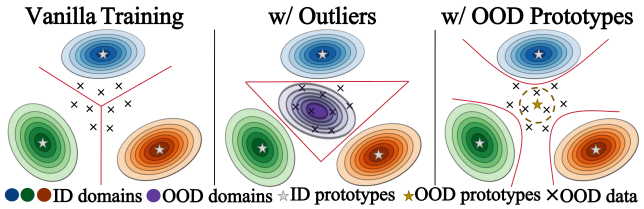


Figure 1: Illustration of our motivation. *Left*: Vanilla training. *Middle*: Training with the mixture of ID data and outliers. *Right*: Training with prototypical outlier proxies.

Can we enable deep models to perceive unseen data without introducing any specific outliers?

In this paper, we introduce a novel framework, Prototypical Outlier Proxy (POP), which enables the model to learn about unknowns without exposing it to real or synthesized outliers. As shown in Fig. 1 (right), POP, which acts as a virtual class center, can attract nearby OOD data and thereby compress the decision boundaries to mitigate the over-confidence of the deep model. First, we transform the learnable classifier into a fixed one by using the hierarchical structure of ID data. Then, we add prototypical outlier proxies to the fixed

*Corresponding author.

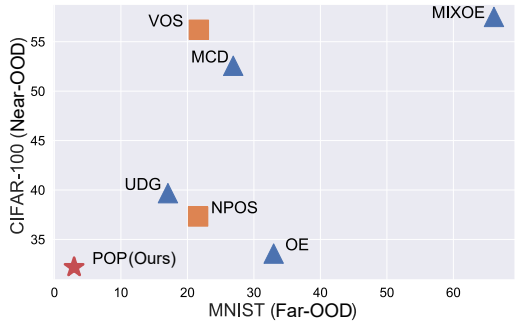


Figure 2: FPR95 (%) of six OOD detection baselines and our POP, using ResNet-18 trained on CIFAR-10, tested on CIFAR-100 and MNIST. Lower FPR95 values indicate better performance. Blue △ denotes real outliers, orange □ denotes synthetic outliers, and red ⋆ is our POP. 'near' and 'far' indicate the degree of difference between ID and OOD data.

classifier to form an OOD-aware deep model. On the other hand, outlier proxies only cover the inter-class regions. For OOD data that are substantially different and easier to detect, we introduce adaptive penalties according to the severity of misclassification. Specifically, we propose a hierarchical similarity boundary loss (HSBL) which enables the deep model to classify data with significantly different features using the semantic hierarchical prior knowledge. As shown in Fig. 2, POP achieves balanced, excellent results in both near-OOD and far-OOD cases. Conversely, OE methods like OE (Hendrycks, Mazeika, and Dietterich 2019) and NPOS (Tao et al. 2023) perform well on near-OOD data but fail on the simpler MNIST dataset. This occurs because OE methods may force the feature extractor to overemphasize local feature discrimination between ID and OOD while lacking global data manifold understanding, hindering distant data perception.

In experiments, POP surpasses state-of-the-art methods, including both OE and post-hoc OOD detection, across two small-scale and one large-scale benchmarks. We also test two of the latest challenging OOD datasets, SSB-hard (Vaze et al. 2022) and NINCO (Bitterwolf, Müller, and Hein 2023). Notably, POP achieves average FPR95 reductions of 7.70%, 6.30%, and 5.42% over second-best methods on CIFAR-10, CIFAR-100, and ImageNet-200, respectively. Compared to recent outlier synthesis method NPOS (Tao et al. 2023), POP trains 7.2× faster and performs inference 19.5× faster.

In summary, this paper makes the following contributions:

- We first identify the efficiency and generality problems of existing OE methods. To solve them, we introduce a new perspective - Prototypical Outlier Proxy (POP) - to serve as a general surrogate for OOD data.
- We introduce a non-learnable classifier to mitigate the mutual influence between ID and OOD prototypes, and a similarity-based optimization objective to adaptively penalize misclassification.
- We conduct extensive experiments to understand the efficacy and efficiency of POP and also verify its scalability on the large-scale ImageNet dataset.

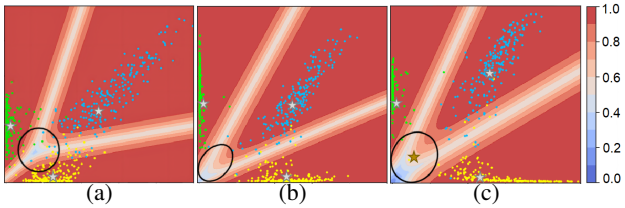


Figure 3: **Toy example.** Use a ResNet-18 with a feature layer size of 2 for three CIFAR-10 classes. The x- and y-axes represent the feature values in the square region. We evaluate prediction confidence for each point in these classes. Yellow, green, and blue points represent deer, horse, and ship, respectively. (a) Vanilla ResNet-18. (b) Fixed ResNet-18. (c) Fixed ResNet-18 with one outlier proxy (brown star marker). Their accuracy is both 99.7%.

Motivation of Algorithm Design

We address the challenges of appending outliers during training through the use of outlier proxies. This section outlines our motivation for this approach. Our approach is grounded in the concept of neural collapse (Papyan, Han, and Donoho 2020), observed during deep model training. As training progresses, features for each class converge around their mean, forming symmetrically distributed clusters. Concurrently, the classifier's weights align with these means, effectively matching well-trained features to their class prototypes. These prototypes represent the domain center of each corresponding class. Building on this, we incorporate additional prototypes as outlier proxies to create a virtual OOD domain, thereby enhancing the model's ability to recognize OOD data without being biased toward specific outlier characteristics. However, accurately positioning outlier proxies in high-dimensional space is non-trivial. They must maintain a suitable distance from ID prototypes—neither too distant nor too close. Additionally, since ID prototypes continuously change during training, determining exact outlier proxies becomes intractable. To address this, we propose pre-defining ID prototypes by fixing the final classifier's weights, making them non-learnable. This approach simplifies the determination of suitable outlier proxies, which will be detailed in the next section.

To validate the feasibility of this intuitive idea, we conducted a toy experiment. For detailed settings of the toy experiment, please refer to Appendix A. First, we train a baseline vanilla ResNet-18 (He et al. 2016). As shown in Fig 3 (a), this model exhibits extensive high-confidence regions (red) across the feature space, except at the decision boundaries. Even within the intersection of the three classes (black circle), the prediction confidence remains around 60%, highlighting the prevalent issue of overconfidence in deep neural networks. Next, we conduct an experiment using a model with pre-defined ID prototypes based on a simple semantic hierarchy before fixing the classifier of the vanilla ResNet-18. The results in Fig. 3 (b) show that the fixed model has tighter compression at decision boundaries, improving feature separation. However, the confidence levels in the high-confidence regions (red) and the intersection of the three classes (black

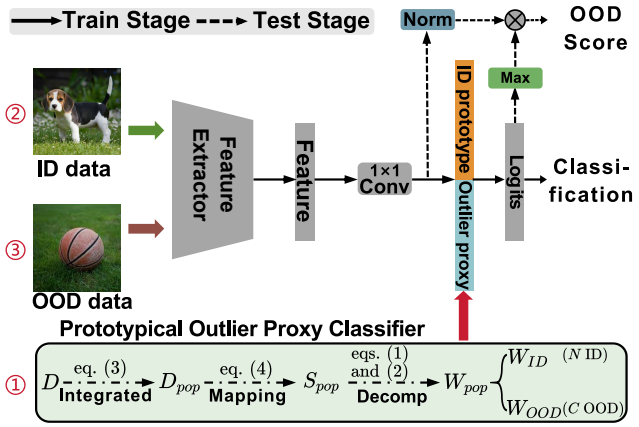


Figure 4: The overview of POP. The contributions module in POP is colored. Before training, in the green module at the bottom (①), integrate prototypical outlier proxies into the fixed classifier. Then, ID data is fed into the model for learning (②). Finally, during the test phase (③), OOD data is fed into the model, and the OOD score is calculated using the feature norm and logits.

circle) remain unchanged, still demonstrating overconfidence. Finally, we blend a single prototypical outlier proxy at the center of three ID prototypes. The results, depicted in Fig. 3 (c), show significant changes. The decision boundaries have widened, improving feature separation and leading to a more spread-out distribution. Notably, at the intersections (black circles), the confidence color shifts to light blue, indicating a decrease in prediction confidence to around 30%.

Proposed Method

In this section, we first present background knowledge on fixed classifiers, then introduce our novel approach, **POP**, a prototypical outlier proxy framework, and a hierarchical similarity boundary loss (HSBL) that imposes penalties based on misclassification severity. Finally, we explain the score function used for OOD detection. The overview framework is illustrated in Fig. 4.

Preliminary: Hierarchy-Aware Frame

HAFrame (Garg, Sani, and Anand 2022; Liang and Davis 2023) is introduced to fix the classifier utilizing the semantic hierarchical prior of ID data. Building on the fixed ID prototypes, we can easily mix prototypical outlier proxies to determine their positions. Common datasets and wild world data often follow a hierarchical label structure similar to WordNet (Fellbaum 1998), forming a weighted tree with all class labels as leaf nodes. The semantic distance between two classes, y_i and y_j, is measured by the height of their lowest common ancestor (LCA) in the tree, denoted as $d_{ij} = \mathrm{H}(\mathrm{LCA}(y_i, y_j))$, where $\mathrm{H}(\cdot)$ is the height function, $i, j \in \{1, 2, \ldots, N\}$, and N is the total number of leaf nodes. Next, we apply a monotonically decreasing function ϕ to transform d_{ij} into a similarity measure. This function maps d_{ij} to the interval $[0, 1]$, defining the similarity between y_i and y_j as $s_{ij} = \phi(d_{ij})$. Using these similarity values, we can construct a symmetric matrix $\boldsymbol{S} \in \mathbb{R}^{N\times N}$, where each element $\boldsymbol{S}_{ij} = \boldsymbol{S}_{ji} = s_{ij}$ represents the pairwise similarity between samples. HAFrame utilizes this similarity matrix $\boldsymbol{S}$ and introduces a set of unit vectors $\{\boldsymbol{w_i}\}_{i=1}^{N} \in \mathbb{R}^N$, where each $\boldsymbol{w_i}$ has a magnitude of 1 (i.e., $||\boldsymbol{w_i}|| = 1$). Their cosine similarity satisfies:

$$\cos(\theta_{ij}) = \frac{\boldsymbol{w}_i^T \boldsymbol{w}_j}{\|\boldsymbol{w}_i\|\|\boldsymbol{w}_j\|} = \boldsymbol{w}_i^T \boldsymbol{w}_j = s_{ij},\ \forall 1 \le i \le j \le N.$$

Finally, let $\boldsymbol{W} = [\boldsymbol{w_1}, \boldsymbol{w_2}, ..., \boldsymbol{w_N}]$ represent the classifier's weight vectors, which we consider as ID prototypes. The bias terms $\boldsymbol{b}$ of the linear layer were removed. Employing spectral decomposition and QR decomposition, we obtain:

$$\boldsymbol{S} = \boldsymbol{QPQ^T} = (\boldsymbol{QP^{\frac{1}{2}}U^T})(\boldsymbol{UP^{\frac{1}{2}}Q^T}) = \boldsymbol{W^TW}, \quad (1)$$

where $\boldsymbol{Q}$ and $\boldsymbol{P}$ come from the eigenvalue decomposition of $\boldsymbol{S}$, and $\boldsymbol{U}$ is an orthogonal matrix obtained through QR decomposition from $\boldsymbol{P}$. The ID prototypes $\boldsymbol{W}$ are given by:

$$\boldsymbol{W} = \boldsymbol{UP^{\frac{1}{2}}Q^T}. \quad (2)$$

Prototypical Outlier Proxy Classifier

Using HAFrame, we obtain ID prototypes and incorporate prototypical outlier proxies into the fixed classifier. The overall process of appending outlier proxies is illustrated at the bottom of Fig. 4. This is achieved by augmenting the ID distance matrix $\boldsymbol{D} \in \mathbb{R}^{N\times N}$ with distances greater than $d_{max} = \max(\boldsymbol{D})$, where $\boldsymbol{D}_{ij} = \boldsymbol{D}_{ji} = d_{ij}$. To accommodate C outlier proxies, we expand $\boldsymbol{D}$ to form $\boldsymbol{D}_{pop} \in \mathbb{R}^{(N+C)\times(N+C)}$ by interpolating OOD distances d. The structure of $\boldsymbol{D}_{pop}$ is illustrated below (for simplicity, only two OOD prototypes are shown in this example):

$$\boldsymbol{D}_{pop} = \begin{pmatrix} \overbrace{\begin{matrix} 0 & d_{12} & \cdots & d_{1N} \\ d_{21} & 0 & \cdots & d_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ d_{N1} & d_{N2} & \cdots & 0 \\ d & d & \cdots & d \\ d & d & \cdots & d \end{matrix}}^{N\text{ ID}} & \overbrace{\begin{matrix} d & d \\ d & d \\ \vdots & \vdots \\ d & d \\ 0 & d \\ d & 0 \end{matrix}}^{C\text{ OOD}} \end{pmatrix} \quad (3)$$

Then, We transform $\boldsymbol{D}_{pop}$ into a similarity matrix $\boldsymbol{S}_{pop}$ using an inverse mapping function ϕ: The formula for ϕ is:

$$\phi(d_{ij}) = \frac{1}{d_{ij} + 1}, \quad (4)$$

where d_{ij} is an element of the $\boldsymbol{D}_{pop}$. This function maps $\boldsymbol{D}_{pop}$ to the interval $[0, 1]$, resulting in $\boldsymbol{S}_{pop} = \phi(\boldsymbol{D}_{pop})$, represents the similarity between mixed ID prototypes and outlier proxies. Subsequently, utilizing the matrix decomposition from Eqs. (1) and (2), we derive the classifier $\boldsymbol{W}_{pop}$ by combining ID prototypes $\boldsymbol{W}_{ID}$ and outlier proxies $\boldsymbol{W}_{OOD}$:

$$\begin{aligned} \boldsymbol{W}_{pop} &= [\boldsymbol{W}_{ID}, \boldsymbol{W}_{OOD}] \\ &= [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_N, \boldsymbol{w}_{N+1}, \boldsymbol{w}_{N+2}, \ldots, \boldsymbol{w}_{N+C}]. \end{aligned} \quad (5)$$

Hierarchical Similarity Boundary Loss

Due to the removal of the classifier's bias $\boldsymbol{b}$ and normalization of $\boldsymbol{W}_{pop}$ (ensuring $||\boldsymbol{w}_i|| = 1$), we also normalize the feature

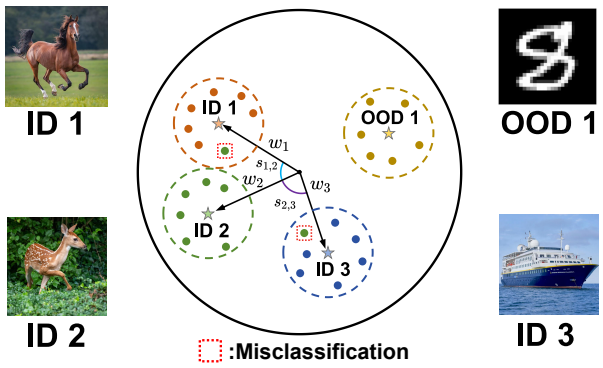


Figure 5: The principles of HSBL

$\boldsymbol{x}$ such that $||\boldsymbol{x}|| = 1$, where $\boldsymbol{x} \in \mathbb{R}^M$ denotes the feature vector of the ID data. For the i^{th} ID data's feature $\boldsymbol{x}_i$ with ground-truth label y_i and predicted label $\hat{y}_i$, the cross-entropy (CE) loss $\mathcal{L}_{ce}$ in cosine space is:

$$\begin{aligned}\mathcal{L}_{ce} &= -\sum_{i}^{N+C} \log \frac{e^{\boldsymbol{w}_{y_i}^T \boldsymbol{x}_i}}{\sum_{j=1}^{N+C} e^{\boldsymbol{w}_j^T \boldsymbol{x}_i}} \\ &= -\sum_{i}^{N+C} \log \frac{e^{\cos(\theta_{y_i,i})}}{\sum_{j=1}^{N+C} e^{\cos(\theta_{j,i})}},\end{aligned} \tag{6}$$

where $\theta_{i,j}$ denotes the angle between $\boldsymbol{w}_i$ and $\boldsymbol{w}_j$. CE loss treats all misclassifications equally, however, in a hierarchical structure, misclassifying different species is more severe than misclassifying different objects within the same species (e.g., in autonomous driving, misclassifying a person as a sedan is far more dangerous than misclassifying a truck as a sedan, so the penalty for the former should be higher). As shown in Fig. 5, if a sample from ID 2 is misclassified as ID 1 (red dashed box), the penalty is $m_{12} = 1 - s_{12}$, which is smaller due to their high similarity s_{12}. Conversely, misclassifying it as ID 3 incurs a larger penalty m_{23} because of the lower similarity s_{23}. We utilize the hierarchical similarity s_{ij} between classes, combining it with the CE loss to improve the model's discrimination of significantly different OOD data. We integrate s_{ij} into Eq. (6) to derive our new hierarchical similarity boundary loss:

$$\begin{aligned}\mathcal{L}_{hsbl} &= -\sum_{i}^{N+C} \log \frac{e^{\beta(\boldsymbol{w}_{y_i}^T \boldsymbol{x}_i - m_{\hat{y}_i y_i})}}{e^{\beta(\boldsymbol{w}_{y_i}^T \boldsymbol{x}_i - m_{\hat{y}_i y_i})} + \sum_{j=1, j\neq y_i}^{N+C} e^{\beta \boldsymbol{w}_j^T \boldsymbol{x}_i}} \\ &= -\sum_{i}^{N+C} \log \frac{e^{\beta(\cos(\theta_{y_i,i}) - m_{\hat{y}_i y_i})}}{e^{\beta(\cos(\theta_{y_i,i}) - m_{\hat{y}_i y_i})} + \sum_{j=1}^{N+C} e^{\beta \cos(\theta_{i,j})}},\end{aligned} \tag{7}$$

where:

$$m_{\hat{y}_i y_i} = 1 - s_{\hat{y}_i y_i} \quad = \begin{cases} 0, & \hat{y}_i = y_i \\ 1 - s_{\hat{y}_i y_i} & \hat{y}_i \neq y_i \end{cases}. \tag{8}$$

The penalty $m_{\hat{y}_i y_i}$ is inversely proportional to the similarity between the predicted and true classes. β is a scaling factor to enhance learning performance.

Algorithm 1: The algorithm of POP

 # The Training Stage

Input: Initial parameters θ for feature extractor $h(\cdot\,;\theta)$, hierarchical distance matrix $\boldsymbol{D}$, the number of outlier proxies C, OOD distance d

1: Insert C^{th} rows and columns d into $\boldsymbol{D}$ to construct $\boldsymbol{D}_{pop}$ following Eq. (3)
2: Map $\boldsymbol{D}_{pop}$ through ϕ to get $\boldsymbol{S}_{pop}$ using Eq. (4)
3: Decompose $\boldsymbol{S}_{pop}$ through matrix decomposition to obtain $\boldsymbol{W}_{pop}$ following Eqs. (1) and (2)
4: **for** *some training iterations* **do**
5: Optimize the parameters θ in a feature extractor $h(;\theta)$ using the HSBL following Eq. (7)
6: **end for**
7: **return** $\theta, \boldsymbol{W}_{pop}$

 # The Test Stage

Input: A trained feature extractor $h(\cdot\,;\theta)$, test sample X_i, threshold λ

1: Extract the feature $\boldsymbol{x}_i = h(X_i;\theta)$
2: Calculate OOD score S using Eq. (9)
3: **return** OOD detection decision $\mathbf{1}\{S \geq \lambda\}$

OOD Score at Test-Time

During the OOD detection phase, to avoid the distortion caused by softmax compression of the logits from the introduced prototypical outlier proxies, we use MaxLogit (Hendrycks et al. 2022) instead of the softmax-based MSP (Hendrycks and Gimpel 2017). Since the logits are in cosine space, we use the feature norm for scaling:

$$S(X_i) = ||\boldsymbol{x}_i|| \cdot \max(\boldsymbol{z}_i), \tag{9}$$

where $\boldsymbol{x}_i$ represents the feature vector of the i^{th} sample X_i, and $\boldsymbol{z}_i$ denotes the logit values. It is worth noting that the score function does not require access to ID data, making it both efficient and secure. The training and inference stages of POP are summarized in Algorithm 1.

Experiments

We first describe the experimental setup and then show that POP performs competitively compared to other state-of-the-art methods. Next, we perform extensive ablations to understand the impact of appending outlier proxies.

Experimental Setup

Datasets. For comprehensive experiments, we adopt the OpenOOD benchmark (Yang et al. 2022a; Zhang et al. 2023c), which provides an accurate, standardized, and unified evaluation for fair testing. We include small-scale datasets CIFAR-10 (Krizhevsky, Hinton et al. 2009) and CIFAR-100 (Krizhevsky, Hinton et al. 2009), and the large-scale ImageNet-200, which is a subset of ImageNet-1k (Deng et al. 2009) with the first 200 classes, as our ID datasets. Among them, (i) CIFAR-10 is a small dataset with 10 classes, including 50k training images and 10k test images. We establish OOD test dataset with CIFAR-100, Tiny ImageNet (TIN) (Torralba, Fergus, and Freeman 2008), MNIST (Xiao,

Methods	OOD Datasets						
	CIFAR-100	**TIN**	**MNIST**	**SVHN**	**Textures**	**Places365**	**Average**
	FPR95/AUROC	FPR95/AUROC	FPR95/AUROC	FPR95/AUROC	FPR95/AUROC	FPR95/AUROC	FPR95/AUROC
MSP (Hendrycks and Gimpel 2017)	59.89/86.73	47.21/88.64	19.22/93.95	24.22/91.57	40.42/89.13	41.83/89.35	38.79/89.90
Energy (Liu et al. 2020)	72.69/85.55	62.41/88.31	15.49/96.32	30.16/92.38	60.22/88.64	56.37/89.64	49.55/90.14
KNN (Sun et al. 2022)	37.90/89.75	31.18/91.65	20.61/94.41	20.88/92.89	24.50/93.02	29.50/92.10	27.43/92.30
MaxLogit (Hendrycks et al. 2022)	62.14/86.84	50.71/88.87	16.39/95.68	31.44/92.47	49.40/89.38	46.21/89.84	42.71/90.51
ViM (Wang et al. 2022)	53.61/87.44	42.49/89.57	18.04/94.25	18.71/94.39	21.79/94.76	44.48/89.14	33.18/91.59
VOS (Du et al. 2022)	56.21/87.42	47.18/89.17	21.72/94.06	59.16/83.49	42.84/89.46	44.14/89.89	45.20/88.92
NPOS (Tao et al. 2023)	37.32/88.87	30.48/91.50	21.61/94.82	2.54/99.30	23.37/94.34	30.07/91.86	<u>24.23</u>/<u>93.44</u>
POP (Ours)	32.19/91.77	21.18/94.76	2.96/99.43	7.72/98.45	16.59/96.50	18.56/95.65	**16.53**/**96.09**

Table 1: Experiment results on CIFAR-10. The utilized metrics include FPR95 (↓), aiming for lower values to indicate better performance; AUROC (↑), where higher values denote superior discriminative ability; and ID Accuracy, measuring the rate of correct classifications. The top-performing models are marked with **bold** for the best and <u>underline</u> for the second best.

Rasul, and Vollgraf 2017) (including <u>Fashion MNIST</u> (Deng 2012)), <u>Texture</u>(Cimpoi et al. 2014), and <u>Places365</u> (Zhou et al. 2016). (ii) CIFAR-100, another small dataset, consists of 50k training images and 10k test images, with 100 classes. The OOD test dataset includes <u>CIFAR-10</u>, with the remaining datasets configured identically to those in (i). (iii) For the large-scale dataset ImageNet-200, the OOD test dataset consist of <u>SSB</u> (Vaze et al. 2022) <u>NINCO</u> (Bitterwolf, Müller, and Hein 2023), <u>iNatruelist</u> (Van Horn et al. 2018), <u>Place365</u>, and <u>OpenImage-O</u> (Wang et al. 2022).

Baselines. We compare our POP with 7 baselines. They are mainly divided into two categories: (1) post-hoc inference methods: **MSP** (Hendrycks and Gimpel 2017), **Energy** (Liu et al. 2020), **ViM** (Wang et al. 2022), and **Maxlogit** (Hendrycks et al. 2022); (2) adding outliers methods: **VOS** (Du et al. 2022), **NPOS** (Tao et al. 2023).

Evaluation metrics. We evaluate our method using (1) the false positive rate (FPR95) at the threshold where the true positive rate for ID samples is 95% and (2) the area under the receiver operating characteristic curve (AUROC). Both metrics are reported as percentages. In ablation experiments, FPR95 and AUROC are averaged across the benchmark.

Training details. We train a ResNet-18 model (He et al. 2016) from scratch for 100 epochs on CIFAR-10 and CIFAR-100, and 90 epochs on ImageNet-200, using a single Nvidia 4090. Training is performed with the SGD optimizer, a learning rate of 0.1, momentum of 0.9, and weight decay of 0.0005. The complete experimental setup is provided in Appendix B.

Main Results

In the following, we present the performance of POP.

Results on CIFAR-10. On the CIFAR-10 dataset, the results in Tab. 1 show that POP outperforms other methods, leading by a significant margin in most OOD datasets. Particularly, on the challenging CIFAR-10, TIN, and Place365 datasets, where features closely resemble those in the ID dataset, POP achieves over 91% AUROC. POP also performs well on structured OOD datasets like MNIST, SVHN, and Textures, indicating that our method is effective across OOD data of varying difficulty. In contrast, other methods, whether post-hoc or mixing in outliers, fail to achieve an AUROC above 90% on any OOD dataset. Our average OOD AUROC performance surpasses the second-best by **2.05%**, while FPR95 is significantly reduced by **5.4%**. It is worth noting that POP not only excels in AUROC but also shows exceptional performance in FPR95, as observed in CIFAR-100 and ImageNet-200. This highlights POP's strong robustness across different evaluation metrics and datasets.

Results on CIFAR-100. On the CIFAR-100 dataset, as shown in Tab. 2. Compared to the second-best result, POP performs exceptionally well, improving by **6.3%** in FPR95 and **3.52%** in AUROC. POP exceeds 80% in AUROC across all datasets, except CIFAR-10. This is because CIFAR-100 is a more fine-grained dataset, and many of its labels overlap with those in CIFAR-10 due to the hierarchical structure shared between the two datasets. The performance of VOS and NPOS, which introduced feature-based synthetic outliers, is poor on the simple MNIST dataset. This suggests that using prototypical outlier proxies, rather than actual outliers, is more flexible and effective for handling OOD detection.

Results on ImageNet-200. On the large-scale ImageNet-200 dataset, as detailed in Tab. 3, POP maintains excellent generalization performance, achieving competitive results on the challenging SSB (Vaze et al. 2022) and NINCO (Bitterwolf, Müller, and Hein 2023) datasets. SSB only includes semantic shift, and NINCO ensures that none of the objects in its dataset have appeared in ImageNet (ID), but their features are very similar to ID. VOS and NPOS perform poorly on NINCO, even worse than post-hoc methods that require no training. This suggests that the unreliability of adding outliers is limited and may only be effective on certain OOD datasets. In contrast, compared to the second-best method, POP reduces FPR95 by **5.32%**, highlighting its exceptional ability to minimize false positives and enhance reliability. The average AUROC is also improved by **1.52%**, underscoring POP's strong performance across diverse OOD scenarios.

Ablation Study

To better understand POP, we conducted a thorough ablation study, detailed in Tab. 4. (1) (F + H) Using a fixed model with HSBL for parameter updates notably improves performance on CIFAR-100, but shows no significant improvement on CIFAR-10. We argue that the more complex hierarchical structure of CIFAR-100 provides more similarity

Methods	OOD Datasets						
	CIFAR-10	**TIN**	**MNIST**	**SVHN**	**Textures**	**Places365**	**Average**
	FPR95/AUROC	FPR95/AUROC	FPR95/AUROC	FPR95/AUROC	FPR95/AUROC	FPR95/AUROC	FPR95/AUROC
MSP (Hendrycks and Gimpel 2017)	59.10/78.54	50.36/82.30	63.47/73.54	56.08/79.10	61.37/78.06	55.41/79.62	57.63/78.52
Energy (Liu et al. 2020)	58.82/79.01	52.14/82.66	57.57/77.30	51.24/82.40	60.27/79.33	56.54/79.82	56.09/80.08
KNN (Sun et al. 2022)	72.41/76.76	49.64/83.18	44.21/83.67	56.14/82.65	51.92/83.86	61.46/78.79	55.96/81.48
MaxLogit (Hendrycks et al. 2022)	58.42/79.33	50.89/82.97	48.64/80.40	51.97/83.23	61.62/78.52	58.60/79.49	55.02/80.65
ViM (Wang et al. 2022)	71.17/71.73	54.71/77.94	46.87/81.76	44.52/84.25	46.99/86.28	60.49/76.17	54.12/79.68
VOS (Du et al. 2022)	59.79/78.69	54.29/82.00	43.98/84.34	75.66/73.30	66.58/76.66	58.37/79.29	59.77/79.04
NPOS (Tao et al. 2023)	70.97/75.72	54.17/81.60	77.73/70.96	31.40/91.72	50.90/84.19	59.80/78.53	57.49/80.45
POP (Ours)	66.92/76.74	51.74/82.46	31.38/91.29	30.91/89.76	53.20/83.11	52.80/80.64	**47.82/84.00**

Table 2: Experiment results on CIFAR-100.

Methods	OOD Datasets					
	SSB-hard	**NINCO**	**iNaturelist**	**Places365**	**OpenImage-O**	**Average**
	FPR95/AUROC	FPR95/AUROC	FPR95/AUROC	FPR95/AUROC	FPR95/AUROC	FPR95/AUROC
MSP (Hendrycks and Gimpel 2017)	65.76/79.92	43.59/85.91	26.87/92.67	41.61/88.51	35.80/88.89	42.73/87.17
Energy (Liu et al. 2020)	69.44/79.32	49.59/85.04	26.83/92.51	35.86/90.14	38.04/88.90	43.95/87.18
KNN (Sun et al. 2022)	72.48/77.24	48.41/85.36	28.44/92.57	45.56/85.47	37.30/88.66	46.44/85.85
MaxLogit (Hendrycks et al. 2022)	70.81/79.95	51.31/85.34	25.90/92.85	35.90/90.30	36.50/89.42	44.49/87.72
ViM (Wang et al. 2022)	69.78/75.49	45.81/82.91	30.00/88.96	39.99/85.10	36.66/86.70	44.45/83.83
VOS (Du et al. 2022)	70.86/78.91	52.00/84.21	26.96/92.82	51.57/82.93	38.01/88.98	47.88/85.57
NPOS (Tao et al. 2023)	73.61/74.19	48.53/84.67	20.67/94.75	46.99/88.08	29.39/91.57	43.84/86.65
POP (Ours)	66.71/78.09	43.48/86.82	15.84/96.09	29.24/91.78	31.30/90.72	**37.31/88.70**

Table 3: Experiment results on ImageNet-200.

ID dataset	F	O	H	FPR95 ↓	AUROC ↑
CIFAR-10	✓			27.45	92.75
	✓		✓	25.65	92.72
	✓	✓		23.90	93.67
	✓	✓	✓	**21.25**	**94.69**
CIFAR-100	✓			54.56	79.94
	✓		✓	53.28	81.10
	✓	✓		52.10	81.62
	✓	✓	✓	**47.82**	**84.00**

Table 4: The ablation study results for CIFAR-10 and CIFAR-100. The best performances in **bold**. F: fixed ResNet-18. O: fixed ResNet-18 with outlier proxies. H: update using HSBL.

Fix Method	CIFAR-10 FPR95/AUROC	CIFAR-100 FPR95/AUROC
Random	32.43/90.48	63.06/74.44
Hierarchy	**16.53/96.09**	**47.82/84.00**

Table 5: The results of various fixed methods on CIFAR-10 and CIFAR-100.

information between classes. (2) (F + O) Appending prototypical outlier proxies enhances generalization performance, showing improvements on both CIFAR-10 and CIFAR-100 compared to using only the fixed model. (3) (F + O + H) Appending prototypical outlier proxies with HSBL for parameter updates, there is a substantial improvement compared to previous methods. Outlier proxies build virtual OOD domains, preserving the model's semantic space and alleviating over-confidence in deep models. Meanwhile, HSBL helps the model classify samples by discriminative features, enhancing its ability to recognize distant OOD data. Thus, their combination improves the model's generalization.

Effects of Hierarchy

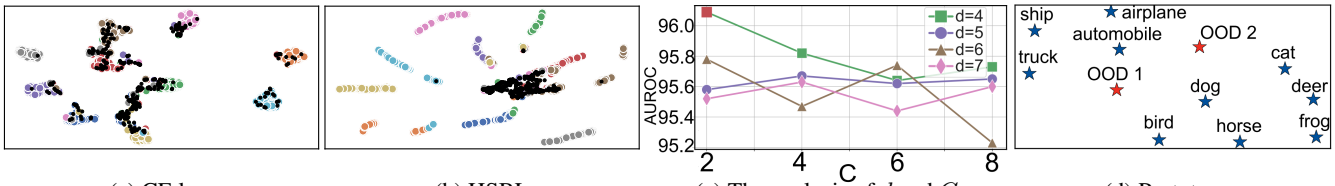
We compared prototypes that were randomly orthogonalized with those decomposed based on hierarchical distances, as shown in Tab. 5. The results indicate that using random prototypes, which lack hierarchical prior information, yields sub-optimal performance.

Analysis of the HSBL

The high-dimensional feature learned by CE loss and HSBL is visualized as shown in Figs. 6 (a) and (b). Clearly, using HSBL results in tighter intra-class compression and greater inter-class separation. HSBL is capable of compressing the features of OOD data into a smaller region. This helps the model better distinguish ID and OOD data. Also, recent work (Ma, Tsao, and Shum 2022; Chen et al. 2023a) indicates that models with better compression enhance generalization.

Impact of Prototypical Outlier Proxies

On CIFAR-10, with the maximum ID hierarchical distance $d_{max} = 3$, we set $d \in \{4, 5, 6, 7\}$ and $C \in \{2, 4, 6, 8\}$ in



(a) CE loss (b) HSBL (c) The analysis of d and C (d) Prototypes

Figure 6: The analysis of HSBL loss and POP. ID test data (CIFAR-10) and OOD data (SVHN) features are visualized using UMAP (McInnes, Healy, and Melville 2020), with ResNet-18 trained with CE loss and HSBL. Colored points represent ID data, and black points represent OOD data. The prototypes, the classifier's weight vectors, are visualized using UMAP.

Metric (s)	VOS	NPOS	POP (Ours)
Train time	$21.58_{\pm}0.42$	$65.03_{\pm}4.82$	$\mathbf{9.00_{\pm}1.12}$
Infer time	$5.84_{\pm}0.39$	$107.05_{\pm}3.48$	$\mathbf{5.49_{\pm}0.32}$

Table 6: Comparison of running times: training on CIFAR-10 and testing on Place365, conducted on an NVIDIA RTX 4090 (each method tested over 5 rounds on the full dataset).

Eq. (3) and conducted a grid search. The results are shown in Fig. 6 (c). As the number of outlier proxies increases, the overall performance shows a downward trend. We speculate that may be because, in a simple hierarchical structure like CIFAR-10, the inclusion of an excessive number of outlier proxies could prevent the model from effectively utilizing the hierarchical prior information. Therefore, we visualized prototypes for optimal parameters $d = 4$ and $C = 2$ in Fig. 6 (d). ID prototypes (blue stars) in the upper-left maintain the semantic structure as *tools*. Lower-right shows *animal* categories with deer and horse in proximity. Two outlier proxies (red stars) effectively separate these categories, helping the model capture intrinsic semantic information.

Analysis of Time Efficiency for Testing

Tab. 6 shows the comparison of running times between POP and other methods for integrating outliers during the training and inference phases. During training, POP is 2.40 times faster than VOS and 7.23 times faster than NPOS. In testing, POP achieves 1.06 times the speed of VOS and 19.50 times the speed of NPOS. This efficiency is due to POP's simple modifications to the vanilla model and because post-hoc methods avoid accessing ID data. In contrast, NPOS is slower due to KNN (Sun et al. 2022) for distance computation on ID data. The efficiency and effectiveness of POP pave the way for applications requiring high real-time performance.

Related Work

OOD Detection Methods

In OOD detection, one category of methods involves using post-processing techniques without training. Techniques like MSP (Hendrycks and Gimpel 2017) focus on the classifier's output probabilities. Methods such as MaxLogit (Hendrycks et al. 2022) and energy scores (Liu et al. 2020) use the logits. The Mahalanobis distance measure (Lee et al. 2018), relies on the classifier's feature representations. Another approach involves retraining a model by incorporating outliers. OE (Hendrycks and Gimpel 2017) directly trains with labeled real outliers. UDG (Yang et al. 2021) and MCD (Yu and Aizawa 2019) use unlabeled real outliers for unsupervised training. Dream-OOD (Du et al. 2023) employs a powerful diffusion model to generate synthetic outliers based on a text-conditioned latent space derived from ID data. VOS (Du et al. 2022) assumes a certain distribution in the feature space to generate outliers, while NPOS (Tao et al. 2023) extends VOS by generating outliers without a specific distribution. MODE (Zhang et al. 2023a) proposes a multi-scale framework that combines global and local features to improve out-of-distribution detection performance. VOso (Nie et al. 2024) proposes a novel approach to address DNN overconfidence in out-of-distribution detection by creating virtual outliers through semantic region perturbation of in-distribution samples. In contrast, our POP approach directly addresses OOD detection from the perspective of outlier proxies.

Fixed Classifier

Early research into optimizing memory and computational resources has explored fixing the classifier. (Hardt and Ma 2017) examines modifying the final layer of deep learning models, while (Hoffer, Hubara, and Soudry 2018) suggests fixing the classifier to a global scale constant and demonstrates the feasibility of starting the classifier from a Hadamard matrix. FRCR (Huang and Mo 2024) first fixes the classifier using a random matrix and then reorders the classifier for continual learning. (Yang et al. 2022b) proposed a simplex equiangular tight frame that has achieved promising results in fixing classifiers on long-tailed datasets. Meanwhile, HAFrame (Liang and Davis 2023) employs a hierarchical structure to fix the classifier, improving performance on fine-grained classification tasks. Building on HAFrame, we leverage hierarchical prior information and introduce outlier proxies to address OOD detection.

Conclusion

In this paper, we introduce a simple yet effective OOD detection framework, POP, which reshapes the decision boundary between ID and OOD data without exposing the model to real or synthetic outliers. By doing so, POP prevents the model from learning biased characteristics and eliminates the need for synthetic samples, improving both training and inference speeds. Experiments on multiple benchmark datasets demonstrate the superiority of the proposed POP.

Acknowledgments

This work was supported in parts by NSFC (U21B2023), ICFCRT (W2441020), Guangdong Basic and Applied Basic Research Foundation (2023B1515120026), and Scientific Development Funds from Shenzhen University.

References

Amodei, D.; Olah, C.; Steinhardt, J.; Christiano, P.; Schulman, J.; and Mané, D. 2016. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*.

Bitterwolf, J.; Müller, M.; and Hein, M. 2023. In or Out? Fixing ImageNet Out-of-Distribution Detection Evaluation. In *ICML*, volume 202, 2471–2506.

Chen, C.; Tang, L.; Huang, Y.; Han, X.; and Yu, Y. 2023a. CODA: Generalizing to Open and Unseen Domains with Compaction and Disambiguation. In *NeurIPS*, volume 36, 12746–12759.

Chen, L.; Wu, P.; Chitta, K.; Jaeger, B.; Geiger, A.; and Li, H. 2023b. End-to-end autonomous driving: Challenges and frontiers. *arXiv preprint arXiv:2306.16927*.

Cimpoi, M.; Maji, S.; Kokkinos, I.; Mohamed, S.; and Vedaldi, A. 2014. Describing textures in the wild. In *CVPR*, 3606–3613.

Deng, J.; Dong, W.; Socher, R.; Li, L.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A large-scale hierarchical image database. In *CVPR*, 248–255.

Deng, J.; Guo, J.; Xue, N.; and Zafeiriou, S. 2019. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, 4690–4699.

Deng, L. 2012. The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]. *IEEE Signal Processing Magazine*, 29(6): 141–142.

Du, X.; Sun, Y.; Zhu, J.; and Li, Y. 2023. Dream the Impossible: Outlier Imagination with Diffusion Models. In *NeurIPS*, volume 36, 60878–60901.

Du, X.; Wang, Z.; Cai, M.; and Li, Y. 2022. VOS: Learning What You Don't Know by Virtual Outlier Synthesis. In *ICLR*.

Fang, Z.; Li, Y.; Lu, J.; Dong, J.; Han, B.; and Liu, F. 2022. Is out-of-distribution detection learnable? In *NeurIPS*, volume 35, 37199–37213.

Fellbaum, C. 1998. *WordNet: An electronic lexical database*. MIT press.

Garg, A.; Sani, D.; and Anand, S. 2022. Learning Hierarchy Aware Features for Reducing Mistake Severity. In *ECCV*, volume 13684, 252–267.

Guo, C.; Pleiss, G.; Sun, Y.; and Weinberger, K. Q. 2017. On Calibration of Modern Neural Networks. In *ICML*, volume 70, 1321–1330.

Hardt, M.; and Ma, T. 2017. Identity Matters in Deep Learning. In *ICLR*.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. B. 2017. Mask R-CNN. In *ICCV*, 2980–2988.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *CVPR*, 770–778.

Hendrycks, D.; Basart, S.; Mazeika, M.; Mostajabi, M.; Steinhardt, J.; and Song, D. X. 2022. Scaling Out-of-Distribution Detection for Real-World Settings. In *ICML*, volume 162, 8759–8773.

Hendrycks, D.; and Gimpel, K. 2017. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. In *ICLR*.

Hendrycks, D.; Mazeika, M.; and Dietterich, T. G. 2019. Deep Anomaly Detection with Outlier Exposure. In *ICLR*.

Hoffer, E.; Hubara, I.; and Soudry, D. 2018. Fix your classifier: the marginal value of training the last weight layer. In *ICLR*.

Huang, S.; and Mo, J. 2024. Fixed Random Classifier Rearrangement for Continual Learning. arXiv:2402.15227.

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

Lee, K.; Lee, K.; Lee, H.; and Shin, J. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *NeurIPS*, volume 31, 7167–7177.

Levine, S.; Finn, C.; Darrell, T.; and Abbeel, P. 2016. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(1): 1334–1373.

Liang, T.; and Davis, J. 2023. Inducing Neural Collapse to a Fixed Hierarchy-Aware Frame for Reducing Mistake Severity. In *ICCV*, 1443–1452.

Liu, W.; Wang, X.; Owens, J. D.; and Li, Y. 2020. Energy-based Out-of-distribution Detection. In *NeurIPS*, volume 33, 21464–21475.

Ma, Y.; Tsao, D.; and Shum, H.-Y. 2022. On the principles of parsimony and self-consistency for the emergence of intelligence. *Frontiers of Information Technology & Electronic Engineering*, 23(9): 1298–1323.

McInnes, L.; Healy, J.; and Melville, J. 2020. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. arXiv:1802.03426.

Ming, Y.; Fan, Y.; and Li, Y. 2022. Poem: Out-of-distribution detection with posterior sampling. In *ICML*, volume 162, 15650–15665.

Ming, Y.; Yin, H.; and Li, Y. 2022. On the impact of spurious correlation for out-of-distribution detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, 10051–10059.

Nie, J.; Luo, Y.; Ye, S.; Zhang, Y.; Tian, X.; and Fang, Z. 2024. Out-of-distribution detection with virtual outlier smoothing. *International Journal of Computer Vision*, 1–18.

Papyan, V.; Han, X.; and Donoho, D. L. 2020. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40): 24652–24663.

Pei, S.; Zhang, X.; Fan, B.; and Meng, G. 2022. Out-of-distribution Detection with Boundary Aware Learning. In *ECCV*, volume 13684, 235–251.

Sun, Y.; Ming, Y.; Zhu, X.; and Li, Y. 2022. Out-of-Distribution Detection with Deep Nearest Neighbors. In *ICML*, volume 162, 20827–20840.

Tao, L.; Du, X.; Zhu, J.; and Li, Y. 2023. Non-parametric Outlier Synthesis. In *ICLR*.

Torralba, A.; Fergus, R.; and Freeman, W. T. 2008. 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.

Van Horn, G.; Mac Aodha, O.; Song, Y.; Cui, Y.; Sun, C.; Shepard, A.; Adam, H.; Perona, P.; and Belongie, S. 2018. The inaturalist species classification and detection dataset. In *CVPR*, 8769–8778.

Vaze, S.; Han, K.; Vedaldi, A.; and Zisserman, A. 2022. Open-Set Recognition: A Good Closed-Set Classifier is All You Need. In *ICLR*.

Wang, H.; Li, Z.; Feng, L.; and Zhang, W. 2022. Vim: Out-of-distribution with virtual-logit matching. In *CVPR*, 4921–4930.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Yang, J.; Wang, H.; Feng, L.; Yan, X.; Zheng, H.; Zhang, W.; and Liu, Z. 2021. Semantically Coherent Out-of-Distribution Detection. In *ICCV*, 8281–8289.

Yang, J.; Wang, P.; Zou, D.; Zhou, Z.; Ding, K.; Peng, W.; Wang, H.; Chen, G.; Li, B.; Sun, Y.; et al. 2022a. Openood: Benchmarking generalized out-of-distribution detection. In *NeurIPS*, volume 35, 32598–32611.

Yang, Y.; Chen, S.; Li, X.; Xie, L.; Lin, Z.; and Tao, D. 2022b. Inducing Neural Collapse in Imbalanced Learning: Do We Really Need a Learnable Classifier at the End of Deep Neural Network? In *NeurIPS*, volume 35, 37991–38002.

Yu, Q.; and Aizawa, K. 2019. Unsupervised Out-of-Distribution Detection by Maximum Classifier Discrepancy. In *ICCV*, 9517–9525.

Zhang, J.; Gao, L.; Hao, B.; Huang, H.; Song, J.; and Shen, H. 2023a. From Global to Local: Multi-Scale Out-of-Distribution Detection. *IEEE Trans. Image Process.*, 32: 6115–6128.

Zhang, J.; Inkawhich, N.; Linderman, R.; Chen, Y.; and Li, H. 2023b. Mixture Outlier Exposure: Towards Out-of-Distribution Detection in Fine-grained Environments. In *WACV*, 5520–5529.

Zhang, J.; Yang, J.; Wang, P.; Wang, H.; Lin, Y.; Zhang, H.; Sun, Y.; Du, X.; Zhou, K.; Zhang, W.; Li, Y.; Liu, Z.; Chen, Y.; and Li, H. 2023c. OpenOOD v1.5: Enhanced Benchmark for Out-of-Distribution Detection. *arXiv preprint arXiv:2306.09301*.

Zhou, B.; Khosla, A.; Lapedriza, A.; Torralba, A.; and Oliva, A. 2016. Places: An image database for deep scene understanding. *arXiv preprint arXiv:1610.02055*.