RankFeat: Rank-1 Feature Removal for Out-of-distribution Detection

Yue Song, Nicu Sebe, Wei Wang
Department of Information Engineering and Computer Science
University of Trento, Italy
yue.song@unitn.it

Abstract

The task of out-of-distribution (OOD) detection is crucial for deploying machine learning models in real-world settings. In this paper, we observe that the singular value distributions of the in-distribution (ID) and OOD features are quite different: the OOD feature matrix tends to have a larger dominant singular value than the ID feature, and the class predictions of OOD samples are largely determined by it. This observation motivates us to propose `RankFeat`, a simple yet effective `post hoc` approach for OOD detection by removing the rank-1 matrix composed of the largest singular value and the associated singular vectors from the high-level feature (*i.e.,* $\mathbf{X}-\mathbf{s}_1\mathbf{u}_1\mathbf{v}_1^T$). `RankFeat` achieves the *state-of-the-art* performance and reduces the average false positive rate (FPR95) by 17.90% compared with the previous best method. Extensive ablation studies and comprehensive theoretical analyses are presented to support the empirical results.

1 Introduction

In the real-world applications of deep learning, understanding whether a test sample belongs to the same distribution of training data is critical to the safe deployment of machine learning models. The main challenge stems from the fact that current deep learning models can easily give over-confident predictions for out-of-distribution (OOD) data [47]. Recently a rich line of literature has emerged to address the challenge of OOD detection [64, 31, 3, 7, 56, 16, 13, 58, 70, 41, 15, 72, 11, 18, 22].

Previous OOD detection approaches either rely on the feature distance [42], activation abnormality [56], or gradient norm [31]. In this paper, we tackle the problem of OOD detection from another perspective: by analyzing the spectrum of the high-level feature matrices (*e.g.,* the output of Block 3 or Block 4 of a typical ResNet [23] model), we observe that the feature matrices have quite different singular value distributions for the in-distribution (ID) and OOD data (see Fig. 1(a)): *the OOD feature tends to have a much larger dominant singular value than the ID feature, whereas the magnitudes of the rest singular values are very similar.* This peculiar behavior motivates us to remove the rank-1 matrix composed of the dominant singular value and singular vectors from the feature. As displayed in Fig. 1(b), removing the rank-1 feature drastically perturbs the class prediction of OOD samples; a majority of predictions have been changed. On the contrary, most ID samples have consistent classification results before and after removing the subspace. *This phenomenon indicates that the over-confident prediction of OOD samples might be largely determined by the dominant singular value and the corresponding singular vectors.*

Based on this observation, we **assume** that the first singular value of OOD feature tends to be much larger than that of ID feature. The intuition behind is that the OOD feature corresponds to a larger PCA explained variance ratio (being less informative), and the well-trained network weights might cause and amplify the difference (see Sec. E of the supplementary for the detailed illustration). Hence, we conjecture that leveraging this gap might help to better distinguish ID and OOD samples. To this

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

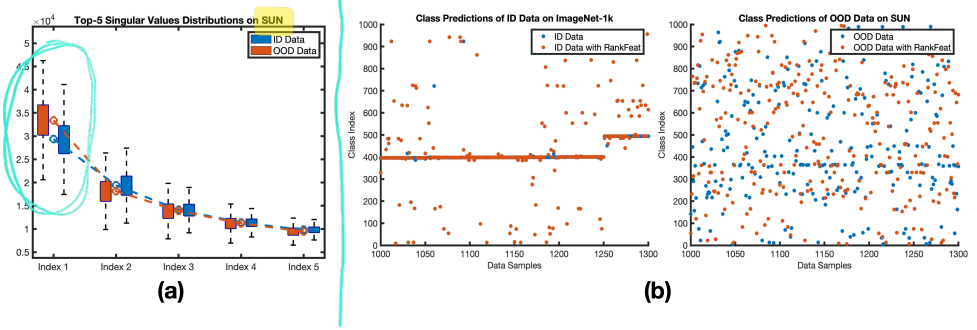

Figure 1: **(a)** The distribution of top-5 singular values for the ID and OOD features on ImageNet-1k and SUN. The OOD feature matrix tends to have a significantly larger dominant singular value. **(b)** After removing the rank-1 matrix composed by the dominant singular value and singular vectors, the class predictions of OOD data are severely perturbed, while those of ID data are moderately influenced. This observation indicates that the decisions of OOD data heavily depend on the dominant singular value and the corresponding singular vectors of the feature matrix. In light of this finding, we get motivated to propose `RankFeat` for OOD detection by removing the rank-1 matrix from the high-level feature. Both observations also hold for other OOD datasets.

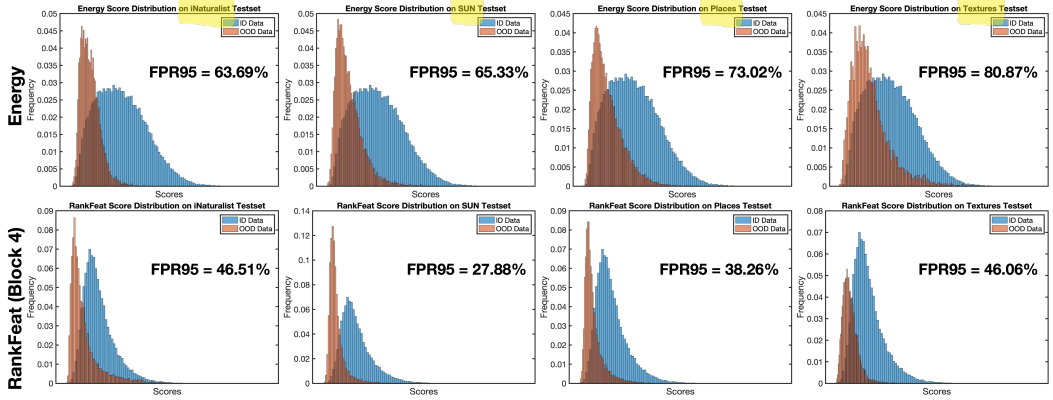

Figure 2: The score distributions of `Energy` [44] (top row) and our proposed `RankFeat` (bottom row) on four OOD datasets. Our method can better separate the ID and OOD data.

end, we propose `RankFeat`, a simple but effective *post hoc* approach for OOD detection. `RankFeat` perturbs the high-level feature by removing its rank-1 matrix composed of the dominant singular value and vectors. Then the logits derived from the perturbed features are used to compute the OOD score function. By removing the rank-1 feature, the over-confidence of OOD samples is mitigated, and consequently the ID and OOD data can be better distinguished (see Fig. 2). Our `RankFeat` establishes the *state-of-the-art* performance on the large-scale ImageNet benchmark and a suite of widely used OOD datasets across different network depths and architectures. In particular, `RankFeat` outperforms the previous best method by **17.90%** in the average false positive rate (FPR95) and by **5.44%** in the area under curve (AUROC). Extensive ablation studies are performed to reveal important insights of `RankFeat`, and comprehensive theoretical analyses are conducted to explain the working mechanism. Code is publicly available via `https://github.com/KingJamesSong/RankFeat`.

The **key results and main contributions** are threefold:

- We propose `RankFeat`, a simple yet effective *post hoc* approach for OOD detection by removing the rank-1 matrix from the high-level feature. `RankFeat` achieves the *state-of-the-art* performance across benchmarks and models, reducing the average FPR95 by **17.90%** and improving the average AUROC by **5.44%** compared to the previous best method.
- We perform extensive ablation studies to illustrate the impact of (1) removing or keeping the rank-1 matrix, (2) removing the rank-n matrix (n>1), (3) applying our `RankFeat` at various network depths, (4) the number of iterations to iteratively derive the approximate

rank-1 matrix for acceleration but without performance degradation, and (5) different fusion strategies to combine multi-scale features for further performance improvements.

- Comprehensive theoretical analyses are conducted to explain the working mechanism and to underpin the superior empirical results. We show that (1) removing the rank-1 matrix reduces the upper bound of OOD score more, (2) removing the rank-1 matrix makes the statistics of OOD feature closer to random matrices, and (3) both `RankFeat` and `ReAct` [56] work by optimizing the upper bound containing the largest singular value. `ReAct` [56] indirectly and manually clips the underlying term, while `RankFeat` directly subtracts it.

2 RankFeat: Rank-1 Feature Removal for OOD Detection

In this section, we introduce the background of OOD detection task and our proposed `RankFeat` that performs the OOD detection by removing the rank-1 matrix from the high-level feature.

Preliminary: OOD detection. The OOD detection is often formulated as a binary classification problem with the goal to distinguish between ID and OOD data. Let f denote a model trained on samples from the ID data $\mathcal{D}_{in}$. For the unseen OOD data $\mathcal{D}_{out}$ at test time, OOD detection aims to define a decision function $\mathcal{G}(\cdot)$:

$$\mathcal{G}(\mathbf{x}) = \begin{cases} \text{in} & \mathcal{S}(\mathbf{x}) > \gamma, \\ \text{out} & \mathcal{S}(\mathbf{x}) < \gamma. \end{cases} \tag{1}$$

where $\mathbf{x}$ denotes the data encountered at the inference stage, $\mathcal{S}(\cdot)$ is the seeking scoring function, and γ is a chosen threshold to make a large portion of ID data correctly classified (*e.g.,* 95%). The difficulty of OOD detection lies in designing an appropriate scoring function $\mathcal{S}(\cdot)$ such that the score distributions of ID and OOD data overlap as little as possible.

RankFeat: rank-1 feature removal. Consider the reshaped high-level feature map $\mathbf{X} \in \mathbb{R}^{C \times HW}$ of a deep network (the batch size is omitted for simplicity). Here 'high-level feature' denotes the feature map that carries rich semantics in the later layers of a network (*e.g.,* the output of Block 3 or Block 4 of a typical deep model like ResNet). Our `RankFeat` first performs the Singular Value Decomposition (SVD) on each individual feature matrix in the mini-batch to decompose the feature:

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T \tag{2}$$

where $\mathbf{S} \in \mathbb{R}^{C \times HW}$ is the rectangular diagonal singular value matrix, and $\mathbf{U} \in \mathbb{R}^{C \times C}$ and $\mathbf{V} \in \mathbb{R}^{HW \times HW}$ are left and right orthogonal singular vector matrices, respectively. Then `RankFeat` removes the rank-1 matrix from the feature as:

$$\mathbf{X}' = \mathbf{X} - \mathbf{s}_1 \mathbf{u}_1 \mathbf{v}_1^T \tag{3}$$

where $\mathbf{s}_1$ is the largest singular value, and $\mathbf{u}_1$ and $\mathbf{v}_1$ are the corresponding left and right singular vectors, respectively. The perturbed feature is fed into the rest of the network to generate the logit predictions $\mathbf{y}'$. Finally, `RankFeat` computes the energy score of the logits for the input $\mathbf{x}$ as:

$$\texttt{RankFeat}(\mathbf{x}) = \log \sum \exp(\mathbf{y}') \tag{4}$$

By removing the rank-1 matrix composed by the dominant singular value $\mathbf{s}_1$, the over-confident predictions of OOD data are largely perturbed. In contrast, the decisions of ID data are mildly influenced. This could help to separate the ID and OOD data better in the logit space.

Acceleration by Power Iteration. Since `RankFeat` only involves the dominant singular value and vectors, there is no need to compute the full SVD of the feature matrix. Hence our method can be potentially accelerated by Power Iteration (PI). The PI algorithm is originally used to approximate the dominant eigenvector of a Hermitian matrix. With a slight modification, it can also be applied to general rectangular matrices. Given the feature $\mathbf{X}$, the modified PI takes the coupled iterative update:

$$\mathbf{v}_k = \frac{\mathbf{X}\mathbf{u}_k}{||\mathbf{X}\mathbf{u}_k||}, \ \mathbf{u}_{k+1} = \left(\frac{\mathbf{v}_k^T \mathbf{X}}{||\mathbf{v}_k^T \mathbf{X}||}\right)^T \tag{5}$$

where $\mathbf{u}_0$ and $\mathbf{v}_0$ are initialized with random orthogonal vectors and converge to the left and right singular vectors, respectively. After certain iterations, the dominant singular value is computed as

$\mathbf{s}_1 = \mathbf{v}_k^T \mathbf{X} \mathbf{u}_k$. As will be illustrated in Sec. 4.3, the approximate solution yielded by PI achieves very competitive performance against the SVD but with much less time overhead.

Combination of multi-scale features. Our `RankFeat` works at various later depths of a model, *i.e.*, Block 3 and Block 4. Since intermediate features might focus on different semantic information, their decision cues are very likely to be different. It is thus natural to consider fusing the scores to leverage the distinguishable information of both features for further performance improvements. Let $\mathbf{y}'$ and $\mathbf{y}''$ denote the logit predictions of Block 3 and Block 4 features, respectively. `RankFeat` performs the fusion at the logit space and computes the score function as $\log \sum \exp((\mathbf{y}'+\mathbf{y}'')/2)$. Different fusion strategies are explored and discussed in Sec. 4.3.

3 Theoretical Analysis

In this section, we perform some theoretical analyses on `RankFeat` to support the empirical results. We start by proving that removing the rank-1 feature with a larger $\mathbf{s}_1$ would reduce the upper bound of `RankFeat` score more. Then based on Random Matrix Theory (RMT), we show that removing the rank-1 matrix makes the statistics of OOD features closer to random matrices. Finally, the theoretical connection of `ReAct` and our `RankFeat` is analyzed and discussed: both approaches work by optimizing the score upper bound determined by $\mathbf{s}_1$. `ReAct` manually uses a pre-defined threshold to clip the term with $\mathbf{s}_1$, whereas our `RankFeat` directly optimizes the bound by subtracting this term.

Removing the rank-1 matrix with a larger $\mathbf{s}_1$ would reduce the upper bound of RankFeat more. For our `RankFeat` score function, we can express its upper bound in an analytical form. Moreover, the upper bound analysis explicitly indicates that removing the rank-1 matrix with a larger first singular value would reduce the upper bound more. Specifically, we have the following proposition.

Proposition 1. *The upper bound of* `RankFeat` *score is defined as* $\texttt{RankFeat}(\mathbf{x}) < \frac{1}{HW}\left(\sum_{i=1}^{N} \mathbf{s}_i - \mathbf{s}_1\right)||\mathbf{W}||_\infty + ||\mathbf{b}||_\infty + \log(Q)$ *where Q denotes the number of categories, and $\mathbf{W}$ and $\mathbf{b}$ are the weight and bias of the last layer, respectively. A larger $\mathbf{s}_1$ would reduce the upper bound more.*

Proof. For the feature $\mathbf{X} \in \mathbb{R}^{C \times HW}$, its SVD $\mathbf{U}\mathbf{S}\mathbf{V}^T = \mathbf{X}$ can be expressed as the summation of rank-1 matrices $\mathbf{X} = \sum \mathbf{s}_i \mathbf{u}_i \mathbf{v}_i^T$. The feature perturbed by `RankFeat` can be computed as:

$$\mathbf{X}' = \mathbf{X} - \mathbf{s}_1 \mathbf{u}_1 \mathbf{v}_1^T = \sum_{i=2}^{N} \mathbf{s}_i \mathbf{u}_i \mathbf{v}_i^T \tag{6}$$

where N denotes the shorter side of the matrix (usually $N = HW$). In most deep models [23, 24], usually the last feature map needs to pass a Global Average Pooling (GAP) layer to collapse the width and height dimensions. The GAP layer can be represented by a vector

$$\mathbf{m} = \frac{1}{HW}[1, 1, \ldots, 1]^T \tag{7}$$

The pooled feature map is calculated as $\mathbf{X}'\mathbf{m}$. Then the output logits are computed by the matrix-vector product with the classification head as:

$$\mathbf{y}' = \mathbf{W}\mathbf{X}'\mathbf{m} + \mathbf{b} = \sum_{i=2}^{N} (s_i \mathbf{W}\mathbf{u}_i \mathbf{v}_i^T \mathbf{m}) + \mathbf{b} \tag{8}$$

where $\mathbf{W} \in \mathbb{R}^{W \times C}$ denotes the weight matrix, $\mathbf{b} \in \mathbb{R}^{Q \times 1}$ represents the bias vector, and $\mathbf{y}' \in \mathbb{R}^{Q \times 1}$ is the output logits that correspond to the perturbed feature $\mathbf{X}'$. Our `RankFeat` score is computed as:

$$\texttt{RankFeat}(\mathbf{x}) = \log \sum_{i=1}^{Q} \exp(\mathbf{y}'_i) \tag{9}$$

where $\mathbf{x}$ is the input image, and Q denotes the number of categories. Here we choose Energy [44] as the base function due to its theoretical alignment with the input probability density and its strong empirical performance. Eq. (9) can be re-formulated by the `Log-Sum-Exp` trick

$$\log \sum_{i=1}^{Q} \exp(\mathbf{y}'_i) = \log \sum_{i=1}^{Q} \exp(\mathbf{y}'_i - \max(\mathbf{y}')) + \max(\mathbf{y}') \tag{10}$$

The above equation directly yields the tight bound as:

$$\max(\mathbf{y}') < \log \sum \exp(\mathbf{y}') < \max(\mathbf{y}') + \log(Q) \tag{11}$$

Since $\max(\mathbf{y}') \leq \max(|\mathbf{y}'|) = ||\mathbf{y}'||_\infty$, we have

$$\texttt{RankFeat}(\mathbf{x}) = \log \sum \exp(\mathbf{y}') < \max(\mathbf{y}') + \log(Q) \leq ||\mathbf{y}'||_\infty + \log(Q) \tag{12}$$

The vector norm has the property of triangular inequality, *i.e.,* $||\mathbf{a}+\mathbf{c}|| \leq ||\mathbf{a}|| + ||\mathbf{c}||$ holds for any vectors $\mathbf{a}$ and $\mathbf{c}$. Moreover, since both $\mathbf{u}$ and $\mathbf{v}$ are orthogonal vectors, we have the relation $||\mathbf{u}_i||_\infty \leq 1$ and $||\mathbf{v}_i||_\infty \leq 1$. Relying on these two properties, injecting eq. (8) into eq. (12) leads to

$$\texttt{RankFeat}(\mathbf{x}) < \sum_{i=2}^{N} \mathbf{s}_i ||\mathbf{W}\mathbf{u}_i\mathbf{v}_i^T\mathbf{m}||_\infty + ||\mathbf{b}||_\infty + \log(Q) \leq \sum_{i=2}^{N} \mathbf{s}_i ||\mathbf{W}\mathbf{m}||_\infty + ||\mathbf{b}||_\infty + \log(Q) \tag{13}$$

Since $\mathbf{m}$ is a scaled all-ones vector, we have $||\mathbf{W}\mathbf{m}||_\infty = ||\mathbf{W}||_\infty / HW$. The bound is simplified as:

$$\texttt{RankFeat}(\mathbf{x}) < \frac{1}{HW}\Big(\sum_{i=1}^{N} \mathbf{s}_i - \mathbf{s}_1\Big)||\mathbf{W}||_\infty + ||\mathbf{b}||_\infty + \log(Q) \tag{14}$$

As indicated above, removing a larger $\mathbf{s}_1$ would reduce the upper bound of RankFeat score more. □

Remark: Considering that OOD feature usually has a much larger $\mathbf{s}_1$ (see Fig. 1(a)), RankFeat would reduce the upper bound of OOD samples more.

Notice that our bound analysis strives to improve the understanding of OOD methods from new perspectives instead of giving a strict guarantee of the score. For example, the upper bound can be used to explain the shrinkage and skew of score distributions in Fig. 2. Subtracting $\mathbf{s}_1$ would largely reduce the numerical range of both ID and OOD scores, which could squeeze score distributions. Since the dominant singular value $\mathbf{s}_1$ contributes most to the score, removing $\mathbf{s}_1$ is likely to make many samples have similar scores. This would concentrate samples in a smaller region and further skew the distribution. Given that the OOD feature tends to have a much larger $\mathbf{s}_1$, this would have a greater impact on OOD data and skew the OOD score distribution more.

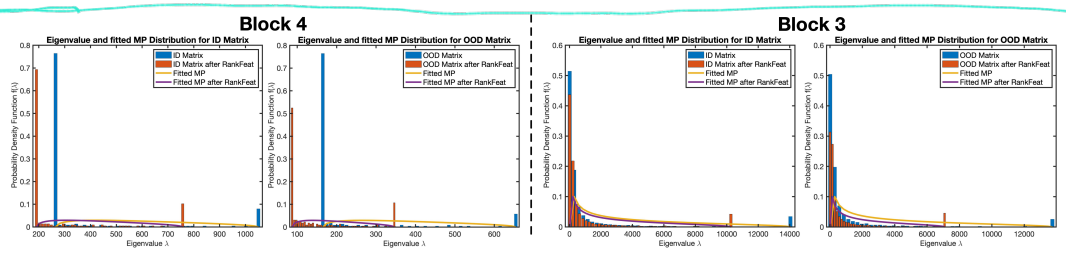

Figure 3: The exemplary eigenvalue distribution of ID/OOD feature and the fitted MP distribution. After the rank-1 matrix is removed, the lowest bin of OOD feature has a larger reduction and the middle bins gain some growth, making the ODD feature statistics closer to the MP distribution.

Removing the rank-1 matrix is likely to make the statistics of OOD features closer to random matrices. Now we turn to use RMT to analyze the statistics of OOD and ID feature matrices. For a random matrix of a given shape, the density of its eigenvalue asymptotically converges to the Manchenko-Pastur (MP) distribution [45, 54]. Formally, we have:

Theorem 1 (Manchenko-Pastur Law [45, 54])**.** *Let* $\mathbf{X}$ *be a random matrix of shape* $t \times n$ *whose entries are random variables with* $E(\mathbf{X}_{ij} = 0)$ *and* $E(\mathbf{X}_{ij}^2 = 1)$*. Then the eigenvalues of the sample covariance* $\mathbf{Y} = \frac{1}{n}\mathbf{X}\mathbf{X}^T$ *converges to the probability density function:* $\rho(\lambda) = \frac{t}{n}\frac{\sqrt{(\lambda_+ - \lambda)(\lambda - \lambda_-)}}{2\pi\lambda\sigma^2}$ *for* $\lambda \in [\lambda_-, \lambda_+]$ *where* $\lambda_- = \sigma^2(1 - \sqrt{\frac{n}{t}})^2$ *and* $\lambda_+ = \sigma^2(1 + \sqrt{\frac{n}{t}})^2$.

This theorem implies the possibility to measure the statistical distance between ID/OOD features and random matrices. To this end, we randomly sample $1,000$ ID and OOD feature matrices and compute the KL divergence between the actual eigenvalue distribution and the fitted MP distribution.

Table 1: The KL divergence between ID/OOD feature and the fitted MP distribution. When the rank-1 feature is removed, the statistics of OOD matrix are closer to random matrices.

Matrix Type	Block 4		Block 3	
	ID	OOD	ID	OOD
Original feature matrix	18.36	18.24	11.27	11.18
Removing rank-1 matrix	17.07 (↓ 1.29)	**15.79 (↓ 2.45)**	9.84 (↓ 1.45)	**8.71 (↓ 2.47)**

Fig. 3 and Table 1 present the exemplary eigenvalue distribution and the average evaluation results of Block 4 and Block 3 features, respectively. For the original feature, the OOD and ID feature matrices exhibit similar behaviors: the distances to the fitted MP distribution are roughly the same ($diff. \approx 0.1$). However, when the rank-1 matrix is removed, the OOD feature matrix has a much larger drop in the KL divergence. This indicates that removing the rank-1 matrix makes the statistics of OOD feature closer to random matrices, *i.e.,* the OOD feature is very likely to become less informative than the ID feature. The result partly explains the working mechanism of `RankFeat`: *by removing the feature matrix where OOD data might convey more information than ID data, the two types of distributions have a larger discrepancy and therefore can be better separated.*

Connection with ReAct [56]. `ReAct` clips the activations at the penultimate layer of a model to distinguish ID and OOD samples. Given the feature $\mathbf{X}$ and the pooling layer $\mathbf{m}$, the perturbation can be defined as:

$$\min(\mathbf{Xm}, \tau) = \mathbf{Xm} - \max(\mathbf{Xm} - \tau, 0) \tag{15}$$

where τ is a pre-defined threshold. Their method shares some similarity with `RankFeat` formulation $\mathbf{Xm} - \mathbf{s}_1\mathbf{u}_1\mathbf{v}_1^T\mathbf{m}$. *Both approaches subtract from the feature a portion of information that is most likely to cause the over-confidence of OOD prediction.* `ReAct` selects the manually-defined threshold τ based on statistics of the whole ID set, while `RankFeat` generates the structured rank-1 matrix from the feature itself. Taking a step further, `ReAct` has the score inequality following eq. (12)

$$\texttt{ReAct}(\mathbf{x}) < ||\mathbf{WXm} - \mathbf{W}\max(\mathbf{Xm} - \tau, 0)||_\infty + ||\mathbf{b}||_\infty + \log(Q) \tag{16}$$

Since $\mathbf{X}$ is non-negative (output of `ReLU`), we have $\max(\mathbf{Xm}) \geq {}^{\max(\mathbf{X})}/_{HW}$. Exploiting the vector norm inequality $||\mathbf{X}||_\mathrm{F} \geq ||\mathbf{X}||_2$ leads to the relation $\max(\mathbf{X}) \geq {}^{\mathbf{s}_1}/_{\sqrt{CHW}}$. Relying on this property, the above inequality can be re-formulated as:

$$\texttt{ReAct}(\mathbf{x}) < \frac{1}{HW}\sum_{i=1}^{N}\mathbf{s}_i||\mathbf{W}||_\infty - \boxed{\frac{1}{HW}\max(\frac{\mathbf{s}_1}{\sqrt{CHW}} - \tau, 0)||\mathbf{W}||_\infty} + ||\mathbf{b}||_\infty + \log(Q) \tag{17}$$

As indicated above, the upper bound of `ReAct` is also determined by the largest singular value $\mathbf{s}_1$. In contrast, the upper bound of our `RankFeat` can be expressed as:

$$\texttt{RankFeat}(\mathbf{x}) < \frac{1}{HW}\sum_{i=1}^{N}\mathbf{s}_i||\mathbf{W}||_\infty - \boxed{\frac{1}{HW}\mathbf{s}_1||\mathbf{W}||_\infty} + ||\mathbf{b}||_\infty + \log(Q) \tag{18}$$

The upper bounds of both methods resemble each other with the only different term boxed. *From this point of view, both methods distinguish the ID and OOD data by eliminating the impact of the term containing $\mathbf{s}_1$ in the upper bound.* `ReAct` optimizes it by clipping the term with a manually-defined threshold, which is indirect and might be sub-optimal. Moreover, the threshold selection requires statistics of the whole ID set. In contrast, our `RankFeat` does not require any extra data and directly subtracts this underlying term which is likely to cause the over-confidence of OOD samples.

4 Experimental Results

In this section, we first discuss the setup in Sec. 4.1, and then present the main experimental results on ImageNet-1k in Sec. 4.2, followed by the extensive ablation studies in Sec. 4.3.

4.1 Setup

Datasets. In line with [30, 56, 31], we mainly evaluate our method on the large-scale ImageNet-1k benchmark [6]. The large-scale dataset is more challenging than the traditional CIFAR benchmark [40] because the images are more realistic and diverse (*i.e.,* 1.28M images of 1,000 classes). For

the OOD datasets, we select four testsets from subsets of `iNaturalist` [62], `SUN` [67], `Places` [74], and `Textures` [5]. These datasets are crafted by [30] with non-overlapping categories from ImageNet-1k. Besides the experiment on the large-scale benchmark, we also validate the effectiveness of our approach on Species [28] and CIFAR [40] benchmark. (see Supplementary Material).

Baselines. We compare our method with 6 recent *post hoc* OOD detection methods, namely `MSP` [26], `ODIN` [43], `Energy` [44], `Mahalanobis` [42], `GradNorm` [31], and `ReAct` [56]. The detailed illustration and settings of these methods are kindly referred to Supplementary Material.

Architectures. In line with [31], the main evaluation is done using Google BiT-S model [39] pretrained on ImageNet-1k with ResNetv2-101 [24]. We also evaluate the performance on SqueezeNet [33], an alternative tiny architecture suitable for mobile devices and on T2T-ViT-24 [71], a tokens-to-tokens vision transformer that has impressive performance when trained from scratch.

For the implementation details and evaluation metrics, please refer to Supplementary Material.

Table 2: Main evaluation results on ResNetv2-101 [24]. All values are reported in percentages, and these *post hoc* methods are directly applied to the model pre-trained on ImageNet-1k [6]. The best three results are highlighted with **red**, **blue**, and **cyan**.

Methods	iNaturalist		SUN		Places		Textures		Average	
	FPR95 (↓)	AUROC (↑)	FPR95 (↓)	AUROC (↑)	FPR95 (↓)	AUROC (↑)	FPR95 (↓)	AUROC (↑)	FPR95 (↓)	AUROC (↑)
MSP [26]	63.69	87.59	79.89	78.34	81.44	76.76	82.73	74.45	76.96	79.29
ODIN [43]	62.69	89.36	71.67	83.92	76.27	80.67	81.31	76.30	72.99	82.56
Energy [44]	64.91	88.48	65.33	85.32	73.02	81.37	80.87	75.79	71.03	82.74
Mahalanobis [42]	96.34	46.33	88.43	65.20	89.75	64.46	52.23	72.10	81.69	62.02
GradNorm [31]	50.03	90.33	46.48	89.03	60.86	84.82	61.42	81.07	54.70	86.71
ReAct [56]	**44.52**	**91.81**	52.71	90.16	62.66	87.83	70.73	76.85	57.66	86.67
RankFeat (Block 4)	**46.54**	81.49	**27.88**	**92.18**	**38.26**	**88.34**	**46.06**	**89.33**	**39.69**	**87.84**
RankFeat (Block 3)	49.61	**91.42**	**39.91**	**92.01**	**51.82**	**88.32**	**41.84**	**91.44**	**45.80**	**90.80**
RankFeat (Block 3 + 4)	**41.31**	**91.91**	**29.27**	**94.07**	**39.34**	**90.93**	**37.29**	**91.70**	**36.80**	**92.15**

4.2 Results

Main results. Following [31], the main evaluation is conducted using Google BiT-S model [39] pretrained on ImageNet-1k with ResNetv2-101 architecture [24]. Table 2 compares the performance of all the *post hoc* methods. For both Block 3 and Block 4 features, our `RankFeat` achieves the best evaluation results across datasets and metrics. More specifically, `RankFeat` based on the Block 4 feature outperforms the second-best baseline by **15.01%** in the average FPR95, while the Block 3 feature-based `RankFeat` beats the second-best method by **4.09%** in the average AUROC. Their combination further surpasses other methods by **17.90%** in the average FPR95 and by **5.44%** in the average AUROC. The superior performances at various depths demonstrate the effectiveness and general applicability of `RankFeat`. The Block 3 feature has a higher AUROC but slightly falls behind the Block 4 feature in the FPR95, which can be considered a compromise between the two metrics.

Our RankFeat is also effective on alternative CNN architectures. Besides the experiment on ResNetv2 [24], we also evaluate our method on SqueezeNet [33], an alternative tiny network suitable for mobile devices and on-chip applications. This network is more challenging because the tiny network size makes the model prone to overfit the training data, which could increase the difficulty to distinguish between ID and OOD samples. Table 3 top presents the performance of all the methods. Collectively, the performances of `RankFeat` are very competitive at both depths, as well as the score fusion. Our `RankFeat` achieves the *state-of-the-art* performances, outperforming the second-best baseline by **14.22%** in FPR95 and by **7.48%** in AUROC.

Our RankFeat also suits transformer-based architectures. To further demonstrate the applicability of our method, we evaluate `RankFeat` on Tokens-to-Tokens Vision Transformer (T2T-ViT) [71], a popular transformer architecture that can achieve competitive performance against CNNs when trained from scratch. Similar to the CNN, `RankFeat` removes the rank-1 matrix from the final token of T2T-ViT before the last normalization layer and the classification head. Table 3 bottom compares the performance on T2T-ViT-24. Our `RankFeat` outperforms the second-best method by **6.11%** in FPR95 and by **2.05%** in AUROC. Since the transformer models [10, 71] do not have increasing receptive fields like CNNs, we do not evaluate the performance at alternative network depths.

Table 3: The results on SqueezeNet [33] and T2T-ViT-24 [71]. All values are reported in percentages, and these *post hoc* methods are directly applied to the model pre-trained on ImageNet-1k [6]. For results on SqueezeNet [33], the best three results are highlighted with **red**, **blue**, and **cyan**.

Model	Methods	iNaturalist		SUN		Places		Textures		Average	
		FPR95 (↓)	AUROC (↑)	FPR95 (↓)	AUROC (↑)	FPR95 (↓)	AUROC (↑)	FPR95 (↓)	AUROC (↑)	FPR95 (↓)	AUROC (↑)
SqueezeNet [33]	MSP [26]	89.83	65.41	83.03	72.25	87.27	67.00	94.61	41.84	88.84	61.63
	ODIN [43]	90.79	65.75	78.32	78.37	83.23	73.31	92.25	43.43	86.15	65.17
	Energy [44]	79.27	73.30	56.41	87.88	67.74	82.73	67.16	64.51	67.65	77.11
	Mahalanobis [42]	91.50	51.79	90.33	62.18	92.26	56.63	58.60	67.16	83.17	59.44
	GradNorm [31]	76.31	73.92	53.63	87.55	65.99	**83.28**	68.72	68.07	66.16	78.21
	ReAct [56]	76.78	68.56	87.57	66.37	88.80	66.20	51.05	76.57	76.05	69.43
	RankFeat (Block 4)	**61.67**	**83.09**	**46.72**	**88.31**	**61.31**	80.52	**38.04**	**88.82**	**51.94**	**85.19**
	RankFeat (Block 3)	**71.04**	**81.50**	**49.18**	**90.43**	**62.94**	**85.82**	**50.14**	**79.32**	**58.33**	**84.28**
	RankFeat (Block 3 + 4)	**65.81**	**83.06**	**46.64**	**90.17**	**61.56**	**84.51**	**42.54**	**85.00**	**54.14**	**85.69**
T2T-ViT-24 [71]	MSP [26]	48.92	88.95	61.77	81.37	69.54	80.03	62.91	82.31	60.79	83.17
	ODIN [43]	**44.07**	88.17	63.83	78.46	68.19	75.33	54.27	83.63	57.59	81.40
	Energy [44]	52.95	82.93	68.55	73.06	74.24	68.17	51.05	83.25	61.70	76.85
	Mahalanobis [42]	90.50	58.13	91.71	50.52	93.32	49.60	80.67	64.06	89.05	55.58
	GradNorm [31]	99.30	25.86	98.37	28.06	99.01	25.71	92.68	38.80	97.34	29.61
	ReAct [56]	52.17	**89.51**	65.23	81.03	68.93	78.20	52.54	85.46	59.72	83.55
	RankFeat	50.27	87.81	**57.18**	**84.33**	**66.22**	**80.89**	**32.64**	**89.36**	**51.58**	**85.60**

Comparison against training-needed approaches. Since our method is *post hoc*, we only compare it with other *post hoc* baselines. MOS [30] and KL Matching [28] are not taken into account because MOS needs extra training processes and KL Matching requires the labeled validation set to compute distributions for each class. Nonetheless, we note that our method can still hold an advantage against those approaches. Table 4 presents the average FPR95 and AUROC on the ImageNet-1k benchmark. Our RankFeat achieves the best performance without any extra training or validation set.

Table 4: Comparison against training-needed methods on ImageNet-1k based on ResNetv2-101 [24].

Method	Post hoc?	Free of Validation Set?	FPR95 (↓)	AUROC (↑)
KL Matching [28]	✓	✗	54.30	80.82
MOS [30]	✗	✓	39.97	90.11
RankFeat	✓	✓	**36.80**	**92.15**

4.3 Ablation Studies

In this subsection, we conduct several ablation studies based on Google-BiT-S ResNetv2-101 model. Unless explicitly specified, we apply `RankFeat` on the Block 4 feature by default.

Table 5: Ablation studies on keeping only the rank-1 matrix and removing the rank-n matrix.

Baselines	iNaturalist		SUN		Places		Textures		Average	
	FPR95 (↓)	AUROC (↑)	FPR95 (↓)	AUROC (↑)	FPR95 (↓)	AUROC (↑)	FPR95 (↓)	AUROC (↑)	FPR95 (↓)	AUROC (↑)
GradNorm [31]	50.03	90.33	46.48	89.03	60.86	84.82	61.42	81.07	54.70	86.71
ReAct [56]	**44.52**	91.81	52.71	90.16	62.66	87.83	70.73	76.85	57.66	86.67
Keeping Only Rank-1	48.97	**91.93**	62.63	84.62	72.42	79.79	49.42	88.86	58.49	86.30
Removing Rank-3	55.19	90.03	48.97	91.26	56.63	**88.81**	86.95	74.57	61.94	86.17
Removing Rank-2	50.04	89.30	48.55	90.99	56.23	88.38	76.86	81.37	57.92	87.51
Removing Rank-1	**46.54**	81.49	**27.88**	**92.18**	**38.26**	88.34	**46.06**	**89.33**	**39.69**	**87.84**

Removing the rank-1 matrix outperforms keeping only it. Instead of removing the rank-1 matrix, another seemingly promising approach is keeping only the rank-1 matrix and abandoning the rest of the matrix. Table 5 presents the evaluation results of keeping only the rank-1 matrix. The performance falls behind that of removing the rank-1 feature by 18.8% in FPR95, which indicates that keeping only the rank-1 feature is inferior to removing it in distinguishing the two distributions. Nonetheless, it is worth noting that even keeping only the rank-1 matrix achieves very competitive performance against previous best methods, such as `GradNorm` [31] and `ReAct` [56].

Removing the rank-1 matrix outperforms removing the rank-n matrix (n>1). We evaluate the impact of removing the matrix of a higher rank, *i.e.*, performing $\mathbf{X} - \sum_{i=1}^{n} s_i \mathbf{u}_i \mathbf{v}_i^T$ where $n>1$ for

the high-level feature $\mathbf{X}$. Table 5 compares the performance of removing the rank-2 matrix and rank-3 matrix. When the rank of the removed matrix is increased, the average performance degrades accordingly. This demonstrates that removing the rank-1 matrix is the most effective approach to separate ID and OOD data. This result is coherent with the finding in Fig. 1(a): only the largest singular value of OOD data is significantly different from that of ID data. Therefore, removing the rank-1 matrix achieves the best performance.

Table 6: The ablation study on applying `RankFeat` to features at different network depths.

Layer	iNaturalist		SUN		Places		Textures		Average	
	FPR95 (↓)	AUROC (↑)	FPR95 (↓)	AUROC (↑)	FPR95 (↓)	AUROC (↑)	FPR95 (↓)	AUROC (↑)	FPR95 (↓)	AUROC (↑)
Block 1	87.81	77.00	59.15	87.29	65.50	84.35	94.15	60.41	76.65	77.26
Block 2	71.84	85.80	61.44	86.46	71.68	81.65	87.89	72.04	73.23	81.49
Block 3	**49.61**	**91.42**	**39.91**	**92.01**	**51.82**	**88.32**	**41.84**	**91.44**	**45.80**	**90.80**
Block 4	**46.54**	81.49	**27.88**	**92.18**	**38.26**	**88.34**	**46.06**	**89.33**	**39.69**	**87.84**

Block 3 and Block 4 features are the most informative. In addition to exploring the high-level features at Block 3 and Block 4, we also investigate the possibility of applying `RankFeat` to features at shallow network layers. As shown in Table 6, the performances of `RankFeat` at the Block 1 and Block 2 features are not comparable to those at deeper layers. This is mainly because the shallow low-level features do not embed as rich semantic information as the deep features. Consequently, removing the rank-1 matrix of shallow features would not help to separate the ID and OOD data.

Table 7: The approximate solution by PI yields competitive performance and costs much less time consumption. The test batch size is set as 16.

Computation Technique	Processing Time Per Image (ms)	iNaturalist		SUN		Places		Textures		Average	
		FPR95 (↓)	AUROC (↑)	FPR95 (↓)	AUROC (↑)	FPR95 (↓)	AUROC (↑)	FPR95 (↓)	AUROC (↑)	FPR95 (↓)	AUROC (↑)
GradNorm [31]	80.01	50.03	90.33	46.48	89.03	60.86	84.82	61.42	81.07	54.70	86.71
ReAct [56]	8.79	**44.52**	**91.81**	52.71	90.16	62.66	87.83	70.73	76.85	57.66	86.67
SVD	18.01	46.54	81.49	**27.88**	**92.18**	38.26	**88.34**	**46.06**	**89.33**	**39.69**	**87.84**
PI (#100 iter)	9.97	46.59	81.49	27.93	92.18	38.28	88.34	46.09	89.33	39.72	**87.84**
PI (#50 iter)	9.47	46.58	81.49	27.93	92.17	**38.24**	88.34	46.12	89.32	39.72	87.83
PI (#20 iter)	9.22	46.58	81.48	27.93	92.15	38.28	88.31	46.10	89.33	39.75	87.82
PI (#10 iter)	9.03	46.77	81.29	28.21	91.84	38.44	87.94	46.08	89.37	39.88	87.61
PI (#5 iter)	9.00	48.34	79.81	30.44	89.71	41.33	84.97	45.34	89.41	41.36	85.98

The approximate solution by PI yields competitive performances. Table 7 compares the time consumption and performance of SVD and PI, as well as two recent *state-of-the-art* OOD methods `ReAct` and `GradNorm`. The performance of PI starts to become competitive against that of SVD ($<0.1\%$) from **20** iterations on with **48.41%** time reduction. Compared with `ReAct`, the PI-based `RankFeat` only requires marginally 4.89% more time consumption. `GradNorm` is not comparable against other baselines in terms of time cost because it does not support the batch mode.

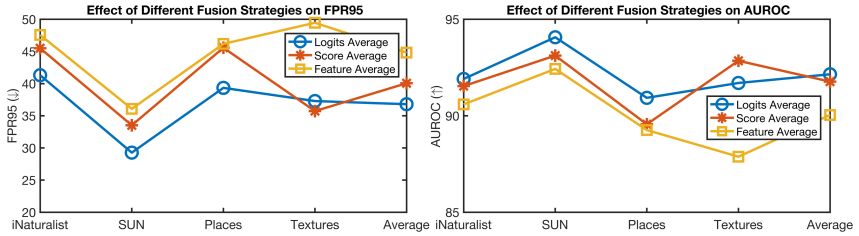

Figure 4: The impact of different fusion strategies on FPR95 and AUROC.

Fusion at the logit space achieves the best performance. Fig. 4 displays the performance of different fusion strategies in combining `RankFeat` at the Block 3 and Block 4 features. As can be observed, averaging logits outperforms other fusion strategies in most datasets and metrics. This indicates that the fusing the logits can best coordinate the benefit of both features.

5 Related Work

Distribution shifts. Distribution shifts have been a long-standing problem in the machine learning research community [21, 49, 38, 66]. The problem of distributions shifts can be generally categorized as shifts in the input space and shifts in the label space. Shifts only in the input space are often deemed as *covariate shifts* [25, 48]. In this setting, the inputs are corrupted by perturbations or shifted by domains, but the label space stays the same [29, 57]. The aim is mainly to improve the robustness and generalization of a model [27]. For OOD detection, the labels are disjoint and the main concern is to determine whether a test sample should be predicted by the pre-trained model [43, 29].

Some related sub-fields also tackle the problem of distribution shifts in the label space, such as novel class discovery [20, 73], open-set recognition [52, 63], and novelty detection [1, 60]. These sub-fields target specific distribution shifts (*e.g.,* semantic novelty), while OOD encompasses all forms of shifts.

OOD detection with discriminative models. The early work on discriminative OOD detection dates back to the classification model with rejection option [4, 14]. The OOD detection methods can be generally divided into training-need methods and *post hoc* approaches. Compared with training-needed approaches, *post hoc* methods do not require any extra training processes and could be directly applied to any pre-trained models. For the wide body of research on OOD detection, please refer to [69] for the comprehensive survey. Here we only highlight the representative *post hoc* methods. Nguyen *et al.* [47] first observed the phenomenon that neural networks easily give over-confident predictions for OOD samples. The following researches attempted to improve the OOD uncertainty estimation by proposing ODIN score [43], OpenMax score [2], Mahalanobis distance [42], and Energy score [44]. Huang *et al.* [30] pointed out that the traditional CIFAR benchmark does not extrapolate to real-world settings and proposed a large-scale ImageNet benchmark. More recently, Sun *et al.* [56] and Huang *et al.* [31] proposed to tackle the challenge of OOD detection from the lens of activation abnormality and gradient norm, respectively. In contrast, based on the empirical observation of singular value distributions, we propose a simple yet effective *post hoc* solution by removing the rank-1 subspace from the high-level features.

OOD detection with generative models. Different from discriminative models, generative models detect the OOD samples by estimating the probability density function [36, 59, 51, 61, 9, 32, 3, 34]. A sample with a low likelihood is deemed as OOD data. Recently, a multitude of methods have utilized generative models for OOD detection [50, 55, 65, 68, 37, 53, 35]. However, as pointed out in [46], generative models could assign a high likelihood to OOD data. Furthermore, generative models can be prohibitively harder to train and optimize than their discriminative counterparts, and the performance is often inferior. This might limit their practical usage.

6 Conclusion

In this paper, we present `RankFeat`, a simple yet effective approach for *post hoc* OOD detection by removing the rank-1 matrix composed by the largest singular value from the high-level feature. We demonstrate its superior empirical results and the general applicability across architectures, network depths, and benchmarks. Extensive ablation studies and comprehensive theoretical analyses are conducted to reveal the important insights and to explain the working mechanism of our method.

Acknowledgments and Disclosure of Funding

This research was supported by the EU H2020 projects AI4Media (No. 951911) and SPRING (No. 871245). We thank our colleague Zhun Zhong for the fruitful discussion and valuable suggestions.

References

[1] Davide Abati, Angelo Porrello, Simone Calderara, and Rita Cucchiara. Latent space autoregression for novelty detection. In *CVPR*, pages 481–490, 2019.

[2] Abhijit Bendale and Terrance E Boult. Towards open set deep networks. In *CVPR*, pages 1563–1572, 2016.

[3] Koby Bibas, Meir Feder, and Tal Hassner. Single layer predictive normalized maximum likelihood for out-of-distribution detection. *NeurIPS*, 34, 2021.

[4] C Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on information theory*, 16 (1):41–46, 1970.

[5] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *CVPR*, 2014.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee, 2009.

[7] James Diffenderfer, Brian Bartoldson, Shreya Chaganti, Jize Zhang, and Bhavya Kailkhura. A winning hand: Compressing deep networks can improve out-of-distribution robustness. *NeurIPS*, 34, 2021.

[8] Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. Repvgg: Making vgg-style convnets great again. In *CVPR*, pages 13733–13742, 2021.

[9] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *ICLR*, 2017.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[11] Xuefeng Du, Zhaoning Wang, Mu Cai, and Yixuan Li. Vos: Learning what you don't know by virtual outlier synthesis. *ICLR*, 2022.

[12] Andrew F Emmott, Shubhomoy Das, Thomas Dietterich, Alan Fern, and Weng-Keen Wong. Systematic construction of anomaly detection benchmarks from real data. In *Proceedings of the ACM SIGKDD workshop on outlier detection and description*, 2013.

[13] Stanislav Fort, Jie Ren, and Balaji Lakshminarayanan. Exploring the limits of out-of-distribution detection. *NeurIPS*, 34, 2021.

[14] Giorgio Fumera and Fabio Roli. Support vector machines with embedded reject option. In *International Workshop on Support Vector Machines*, pages 68–82. Springer, 2002.

[15] Saurabh Garg, Sivaraman Balakrishnan, Zachary C Lipton, Behnam Neyshabur, and Hanie Sedghi. Leveraging unlabeled data to predict out-of-distribution performance. *ICLR*, 2022.

[16] Isaac Gibbs and Emmanuel Candes. Adaptive conformal inference under distribution shift. *NeurIPS*, 34, 2021.

[17] Izhak Golan and Ran El-Yaniv. Deep anomaly detection using geometric transformations. *NeurIPS*, 2018.

[18] Eduardo Dadalto Camara Gomes, Florence Alberge, Pierre Duhamel, and Pablo Piantanida. Igeood: An information geometry approach to out-of-distribution detection. *ICLR*, 2022.

[19] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *ICLR*, 2015.

[20] Kai Han, Andrea Vedaldi, and Andrew Zisserman. Learning to discover novel visual categories via deep transfer clustering. In *ICCV*, pages 8401–8409, 2019.

[21] David J Hand. Classifier technology and the illusion of progress. *Statistical science*, 21(1):1–14, 2006.

[22] Matan Haroush, Tzviel Frostig, Ruth Heller, and Daniel Soudry. A statistical framework for efficient out of distribution detection in deep neural networks. In *ICLR*, 2022.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV*, pages 630–645. Springer, 2016.

[25] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2019.

[26] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *ICLR*, 2017.

[27] Dan Hendrycks, Norman Mu, Ekin Dogus Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. In *ICLR*, 2019.

[28] Dan Hendrycks, Steven Basart, Mantas Mazeika, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. Scaling out-of-distribution detection for real-world settings. *ICML*, 2022.

[29] 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 *CVPR*, pages 10951–10960, 2020.

[30] Rui Huang and Yixuan Li. Mos: Towards scaling out-of-distribution detection for large semantic space. In *CVPR*, pages 8710–8719, 2021.

[31] Rui Huang, Andrew Geng, and Yixuan Li. On the importance of gradients for detecting distributional shifts in the wild. *NeurIPS*, 34, 2021.

[32] Xun Huang, Yixuan Li, Omid Poursaeed, John Hopcroft, and Serge Belongie. Stacked generative adversarial networks. In *CVPR*, pages 5077–5086, 2017.

[33] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size. *ICLR*, 2017.

[34] Dihong Jiang, Sun Sun, and Yaoliang Yu. Revisiting flow generative models for out-of-distribution detection. In *ICLR*, 2022.

[35] Keunseo Kim, JunCheol Shin, and Heeyoung Kim. Locally most powerful bayesian test for out-of-distribution detection using deep generative models. *NeurIPS*, 34, 2021.

[36] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *ICLR*, 2014.

[37] Polina Kirichenko, Pavel Izmailov, and Andrew G Wilson. Why normalizing flows fail to detect out-of-distribution data. *NeurIPS*, 33:20578–20589, 2020.

[38] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *ICML*, pages 5637–5664. PMLR, 2021.

[39] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *ECCV*, 2020.

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

[41] Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. *ICLR*, 2022.

[42] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *NeurIPS*, 31, 2018.

[43] Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *ICLR*, 2018.

[44] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *NeurIPS*, 33:21464–21475, 2020.

[45] Vladimir A Marčenko and Leonid Andreevich Pastur. Distribution of eigenvalues for some sets of random matrices. *Mathematics of the USSR-Sbornik*, 1(4):457, 1967.

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

[47] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *CVPR*, pages 427–436, 2015.

[48] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *NeurIPS*, 32, 2019.

[49] Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. Mit Press, 2008.

[50] Jie Ren, Peter J Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark Depristo, Joshua Dillon, and Balaji Lakshminarayanan. Likelihood ratios for out-of-distribution detection. *NeurIPS*, 32, 2019.

[51] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, pages 1278–1286. PMLR, 2014.

[52] Walter J Scheirer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E Boult. Toward open set recognition. *IEEE TPAMI*, 2012.

[53] Robin Schirrmeister, Yuxuan Zhou, Tonio Ball, and Dan Zhang. Understanding anomaly detection with deep invertible networks through hierarchies of distributions and features. *NeurIPS*, 33:21038–21049, 2020.

[54] Anirvan M Sengupta and Partha P Mitra. Distributions of singular values for some random matrices. *Physical Review E*, 60(3):3389, 1999.

[55] Joan Serrà, David Álvarez, Vicenç Gómez, Olga Slizovskaia, José F Núñez, and Jordi Luque. Input complexity and out-of-distribution detection with likelihood-based generative models. In *ICLR*, 2019.

[56] Yiyou Sun, Chuan Guo, and Yixuan Li. React: Out-of-distribution detection with rectified activations. *NeurIPS*, 34, 2021.

[57] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *ICML*, pages 9229–9248. PMLR, 2020.

[58] Tobias Sutter, Andreas Krause, and Daniel Kuhn. Robust generalization despite distribution shift via minimum discriminating information. *NeurIPS*, 34, 2021.

[59] Esteban G Tabak and Cristina V Turner. A family of nonparametric density estimation algorithms. *Communications on Pure and Applied Mathematics*, 66(2):145–164, 2013.

[60] Jihoon Tack, Sangwoo Mo, Jongheon Jeong, and Jinwoo Shin. Csi: Novelty detection via contrastive learning on distributionally shifted instances. *NeurIPS*, 33:11839–11852, 2020.

[61] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. *NeurIPS*, 29, 2016.

[62] 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 *CVPR*, 2018.

[63] Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Open-set recognition: A good closed-set classifier is all you need. In *ICLR*, 2022.

[64] Haoran Wang, Weitang Liu, Alex Bocchieri, and Yixuan Li. Can multi-label classification networks know what they don't know? *NeurIPS*, 34, 2021.

[65] Ziyu Wang, Bin Dai, David Wipf, and Jun Zhu. Further analysis of outlier detection with deep generative models. *NeurIPS*, 33:8982–8992, 2020.

[66] Olivia Wiles, Sven Gowal, Florian Stimberg, Sylvestre Alvise-Rebuffi, Ira Ktena, Taylan Cemgil, et al. A fine-grained analysis on distribution shift. *ICLR*, 2022.

[67] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010.

[68] Zhisheng Xiao, Qing Yan, and Yali Amit. Likelihood regret: An out-of-distribution detection score for variational auto-encoder. *NeurIPS*, 33:20685–20696, 2020.

[69] Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *arXiv preprint arXiv:2110.11334*, 2021.

[70] Haotian Ye, Chuanlong Xie, Tianle Cai, Ruichen Li, Zhenguo Li, and Liwei Wang. Towards a theoretical framework of out-of-distribution generalization. *NeurIPS*, 34, 2021.

[71] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *ICCV*, 2021.

[72] Alireza Zaeemzadeh, Niccolò Bisagno, Zeno Sambugaro, Nicola Conci, Nazanin Rahnavard, and Mubarak Shah. Out-of-distribution detection using union of 1-dimensional subspaces. In *CVPR*, pages 9452–9461, 2021.

[73] Zhun Zhong, Enrico Fini, Subhankar Roy, Zhiming Luo, Elisa Ricci, and Nicu Sebe. Neighborhood contrastive learning for novel class discovery. In *CVPR*, pages 10867–10875, 2021.

[74] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE TPAMI*, 2017.

A Experimental Setup

Implementation Details. At the inference stage, all the images are resized to 480×480 for ResNetv2-101 [24] and SqueezeNet [33]. The source codes are implemented with `Pytorch 1.10.1`, and all experiments are run on a single NVIDIA Quadro RTX 6000 GPU.

Evaluation Metrics. Following [30, 56, 31], we measure the performance using two main metrics: (1) the false positive rate (FPR95) of OOD examples when the true positive rate of ID samples is at 95%; and (2) the area under the receiver operating characteristic curve (AUROC).

```
#Our RankFeat (SVD) is applied on each individual \\
#feature matrix within the mini-batch.
feat = model.features(inputs)
B, C, H, W = feat.size()
feat = feat.view(B, C, H * W)
u,s,vt = torch.linalg.svd(feat)
feat = feat - s[:,0:1].unsqueeze(2)*u[:,:,0:1].bmm(vt[:,0:1,:])
feat = feat.view(B,C,H,W)
logits = model.classifier(feat)
score = torch.logsumexp(logits, dim=1)
```

Figure 5: Pytorch-like codes of our `RankFeat` implementation.

Pseudo Code of RankFeat. Fig. 5 presents the Pytorch-like implementation of our `RankFeat`. We use `torch.linalg.svd` to conduct SVD on each individual feature matrix in the mini-batch.

B More Evaluation Results

B.1 Large-scale Species Dataset

Table 8: The evaluation results on four sub-sets of Species [28] based on ResNetv2-101 [24]. All values are reported in percentages, and these *post hoc* methods are directly applied to the model pre-trained on ImageNet-1k [6]. The best three results are highlighted with **red**, **blue**, and **cyan**.

Methods	Protozoa		Microorganisms		Plants		Mollusks		Average	
	FPR95 (↓)	AUROC (↑)	FPR95 (↓)	AUROC (↑)	FPR95 (↓)	AUROC (↑)	FPR95 (↓)	AUROC (↑)	FPR95 (↓)	AUROC (↑)
MSP [26]	75.81	83.20	72.23	84.25	61.48	87.78	85.62	70.51	73.79	81.44
ODIN [43]	75.97	**85.11**	65.94	89.35	55.69	90.79	86.22	71.31	70.96	84.14
Energy [44]	79.49	84.34	60.87	**90.30**	54.67	90.95	88.47	70.53	70.88	84.03
ReAct [56]	81.74	84.26	58.82	85.88	**36.90**	**93.78**	90.58	**76.33**	67.02	**85.06**
RankFeat (Block 4)	**66.98**	70.19	**39.06**	86.67	**46.31**	79.98	**80.14**	59.92	**58.12**	74.19
RankFeat (Block 3)	**58.99**	**88.81**	**49.72**	**90.04**	47.01	**91.85**	**80.37**	**79.61**	**59.02**	**87.58**
RankFeat (Block 3 + 4)	**52.78**	**88.65**	**37.21**	**92.82**	**38.07**	**92.88**	**76.38**	**78.13**	**51.11**	**88.37**

The Species [28] dataset is a large-scale OOD validation benchmark consisting of $71,3449$ images, which is designed for ImageNet-1k [6] and ImageNet 21-k [39] as the ID sets. We select four sub-sets as the OOD benchmark, namely `Protozoa`, `Microorganisms`, `Plants`, and `Mollusks`. Table 8 present the evaluation results. Our `RankFeat` achieves the best performance, surpassing other methods by **15.91%** in the average FPR95 and by **3.31%** in the average AUROC.

B.2 CIFAR100 with Different Architectures

We also evaluate our method on the CIFAR benchmark with various model architectures. The evaluation OOD datasets are the same with those of the ImageNet-1k benchmark. We take ResNet-56 [23] and RepVGG-A0 [8] pre-trained on ImageNet-1k as the backbones, and then fine-tune them on CIAR100 [40] for 100 epochs. The learning rate is initialized with 0.1 and is decayed by 10 every 30 epoch. Notice that this training process is to obtain a well-trained classifier but the ODO methods (including ours) are still *post hoc* and do not need any extra training.

Table 9: The evaluation results with different model architectures on CIFAR100 [40]. All values are reported in percentages, and these *post hoc* methods are directly applied to the model. The best two results are highlighted with **red** and **blue**.

Model	Methods	iNaturalist		SUN		Places		Textures		Average	
		FPR95 (↓)	AUROC (↑)	FPR95 (↓)	AUROC (↑)	FPR95 (↓)	AUROC (↑)	FPR95 (↓)	AUROC (↑)	FPR95 (↓)	AUROC (↑)
RepVGG-A0 [8]	MSP [26]	61.55	85.03	91.05	69.19	65.45	**82.10**	86.68	65.56	76.18	75.47
	ODIN [43]	50.20	87.88	88.00	66.56	61.85	79.34	84.87	63.89	71.23	74.42
	Energy [44]	53.71	84.59	86.71	66.58	59.71	78.64	84.57	63.88	71.18	73.42
	Mahalanobis [42]	81.43	74.81	89.77	67.12	79.49	73.06	**64.95**	**82.19**	78.91	74.30
	GradNorm [31]	78.87	68.21	95.10	44.73	66.25	75.41	92.98	43.83	83.30	58.05
	ReAct [56]	**48.09**	**93.00**	**73.87**	**78.12**	**61.63**	78.43	75.23	81.36	**64.71**	**82.73**
	RankFeat	**40.19**	**88.06**	**70.47**	**76.35**	**57.75**	**83.58**	**52.89**	**83.28**	**55.33**	**82.82**
ResNet-56 [23]	MSP [26]	77.69	78.25	93.54	66.93	81.57	76.71	88.47	65.79	85.32	71.92
	ODIN [43]	66.92	79.25	95.05	50.45	77.45	72.88	90.51	53.47	82.48	64.01
	Energy [44]	65.24	79.13	95.05	49.33	77.10	72.32	90.39	52.68	81.95	63.37
	Mahalanobis [42]	89.47	69.32	91.38	54.76	82.32	77.53	**68.83**	**79.64**	83.00	70.31
	GradNorm [31]	96.72	42.09	94.19	47.97	94.61	48.09	89.14	50.18	93.67	47.08
	ReAct [56]	**50.59**	**90.56**	**69.23**	**85.79**	**55.38**	**87.98**	82.60	75.51	**64.50**	**84.96**
	RankFeat	**34.62**	**88.21**	**61.82**	**80.50**	**53.79**	**89.71**	**30.89**	**91.31**	**45.28**	**87.43**

Table 9 compares the performance against all the *post hoc* baselines. Our `RankFeat` establishes the *state-of-the-art* performances across architectures on most datasets and metrics, outperforming the second best method by **9.38 %** in the average FPR95 on RepVGG-A0 and by **19.22 %** in the average FPR95 on ResNet-56. Since the CIFAR images are small in resolution (*i.e.,* 32×32), the downsampling times and the number of feature blocks of the original models are reduced. Hence we only apply `RankFeat` to the final feature before the last GAP layer.

B.3 One-class CIFAR10

To further demonstrate the applicability of our method, we follow [12, 17, 60] and conduct experiments on one-class CIFAR10. The setup is as follows: we choose one of the classes as the ID set while keeping other classes as OOD sets. Table 10 reports the average AUROC on CIFAR10. Our `RankFeat` outperforms other baselines on most sub-set as well as on the average result.

Table 10: The average AUROC (%) on one-class CIFAR10 based on ResNet-56.

Methods	Plane	Car	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck	Mean
MSP	59.75	52.48	62.96	48.73	59.15	52.39	67.33	59.34	54.55	51.97	56.87
Energy	**83.12**	91.56	68.99	56.02	75.03	77.33	69.50	88.41	82.88	84.74	77.76
ReAct	82.24	96.69	78.32	76.84	76.11	86.80	86.15	90.95	89.91	**94.17**	85.82
RankFeat	79.26	**98.54**	**82.04**	**80.28**	**82.89**	**90.28**	**89.06**	**95.30**	**94.11**	94.02	**88.58**

C Baseline Methods

For the convenience of audiences, we briefly recap the previous *post hoc* methods for OOD detection. Some implementation details of the methods are also discussed.

MSP [26]. One of the earliest work considered directly using the Maximum Softmax Probability (MSP) as the scoring function for OOD detection. Let $f(\cdot)$ and $\mathbf{x}$ denote the model and input, respectively. The MSP score can be computed as:

$$\texttt{MSP}(\mathbf{x}) = \max\Big(\text{Softmax}(f(\mathbf{x}))\Big) \tag{19}$$

Despite the simplicity of this approach, the MSP score often fails as neural networks could assign arbitrarily high confidences to the OOD data [47].

ODIN [43]. Based on MSP [26], ODIN [43] further integrated temperature scaling and input perturbation to better separate the ID and OOD data. The ODIN score is calculated as:

$$\texttt{ODIN}(\mathbf{x}) = \max\Big(\text{Softmax}(\frac{f(\bar{\mathbf{x}})}{T})\Big) \tag{20}$$

where T is the hyper-parameter temperature, and $\bar{\mathbf{x}}$ denote the perturbed input. Following the setting in [31], we set T=1000. According to [31], the input perturbation does not bring any performance improvement on the ImageNet-1k benchmark. Hence, we do not perturb the input either.

Energy score [44]. Liu *et al.* [44] argued that an energy score is superior than the MSP because it is theoretically aligned with the input probability density, *i.e.,* the sample with a higher energy correspond to data with a lower likelihood of occurrence. Formally, the energy score maps the logit output to a scalar function as:

$$\texttt{Energy}(\mathbf{x}) = \log \sum_{i=1}^{C} \exp(f_i(\mathbf{x})) \tag{21}$$

where C denotes the number of classes.

Mahalanobis distance [42]. Lee *et al.* [42] proposed to model the Softmax outputs as the mixture of multivariate Gaussian distributions and use the Mahalanobis distance as the scoring function for OOD uncertainty estimation. The score is computed as:

$$\texttt{Mahalanobis}(\mathbf{x}) = \max_i \Big(-(f(\mathbf{x}) - \mu_i)^T \Sigma (f(\mathbf{x}) - \mu_i) \Big) \tag{22}$$

where μ_i denotes the feature vector mean, and Σ represents the covariance matrix across classes. Following [31], we use 500 samples randomly selected from ID datasets and an auxiliary tuning dataset to train the logistic regression and tune the perturbation strength ϵ. For the tuning dataset, we use FGSM [19] with a perturbation size of 0.05 to generate adversarial examples. The selected ϵ is set as 0.001 for ImageNet-1k.

GradNorm [31]. Huang *et al.* [31] proposed to estimate the OOD uncertainty by utilizing information extracted from the gradient space. They compute the KL divergence between the Softmax output and a uniform distribution, and back-propagate the gradient to the last layer. Then the vector norm of the gradient is used as the scoring function. Let $\mathbf{w}$ and $\mathbf{u}$ denote the weights of last layer and the uniform distribution. The score is calculated as:

$$\texttt{GradNorm}(\mathbf{x}) = ||\frac{\partial D_{KL}(\mathbf{u}||\text{Softmax}(f(\mathbf{x})))}{\partial \mathbf{w}}||_1 \tag{23}$$

where $||\cdot||_1$ denotes the L_1 norm, and $D_{KL}(\cdot)$ represents the KL divergence measure.

ReAct [56]. In [56], the authors observed that the activations of the penultimate layer are quite different for ID and OOD data. The OOD data is biased towards triggering very high activations, while the ID data has the well-behaved mean and deviation. In light of this finding, they propose to clip the activations as:

$$\max(f_{l-1}(\mathbf{x}), \tau) \tag{24}$$

where $f_{l-1}(\cdot)$ denotes the activations for the penultimate layer, and τ is the upper limit computed as the 90-th percentile of activations of the ID data. Finally, the Energy score [44] is computed for estimating the OOD uncertainty.

D Visualization about RankFeat

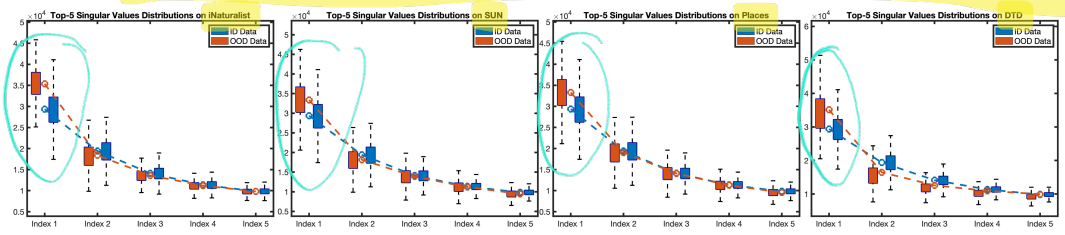

Figure 6: The top-5 singular value distribution of the ID dataset and OOD datasets. The first singular values $\mathbf{s}_1$ of OOD data are consistently much larger than those of ID data on each OOD dataset.

D.1 Singular Value Distribution

Fig. 6 compares the top-5 singular value distribution of ID and OOD feature matrices on all the datasets. Our novel observation consistently holds for every OOD dataset: the dominant singular value s_1 of OOD feature always tends to be significantly larger than that of ID feature.

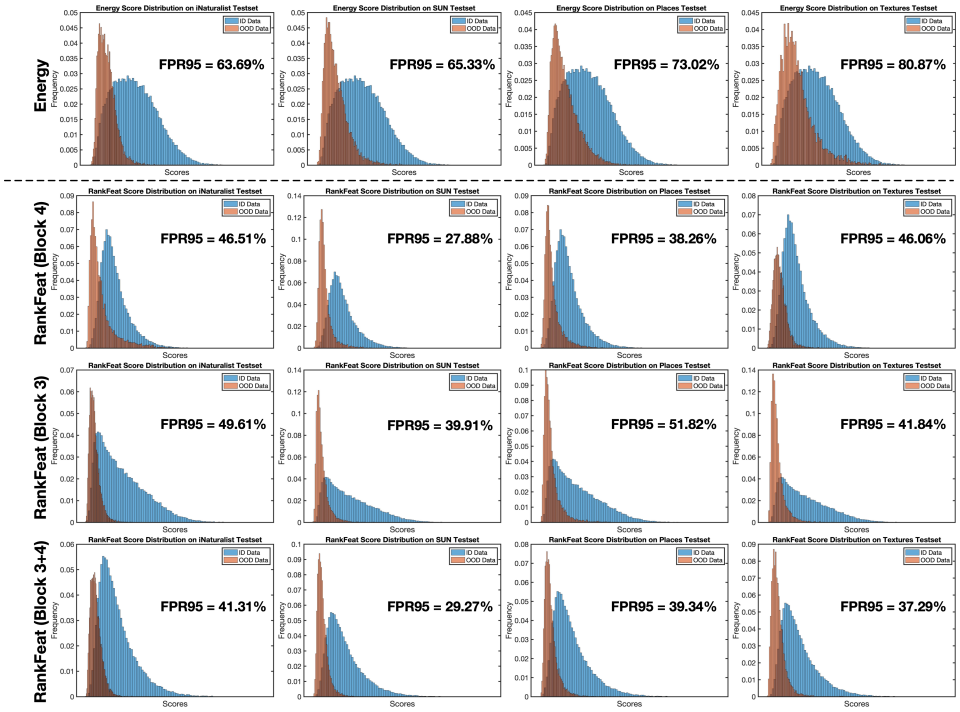

Figure 7: The score distributions of `Energy` [44] (top row) and our proposed `RankFeat` (rest rows) on four OOD datasets. Our `RankFeat` applies to different high-level features at the later depths of the network, and their score functions can be further fused.

D.2 Score Distribution

Fig. 7 displays the score distributions of `RankFeat` at Block 3 and Block 4, as well as the fused results. Our `RankFeat` works for both high-level features. For the score fusion, when Block 3 and Block 4 features are of similar scores ($diff.<5\%$), the feature combination could have further improvements.

D.3 Output Distribution

Fig. 8(a) presents the output distribution (*i.e.,* the logits after `Softmax` layer) on `ImageNet` and `iNaturalist`. After our `RankFeat`, the OOD data have a larger reduction in the probability output; most of OOD predictions are of very small probabilities (<0.1).

D.4 Logit Distribution

Fig. 8(b) displays the logits distribution of our `RankFeat`. The OOD logits after `RankFeat` have much less variations and therefore are closer to the uniform distribution.

E Why are the singular value distributions of ID and OOD features different?

In the paper, we give some theoretical analysis to explain the working mechanism of our `RankFeat`. It would be also interesting to investigate why the singular value distributions of the ID and OOD

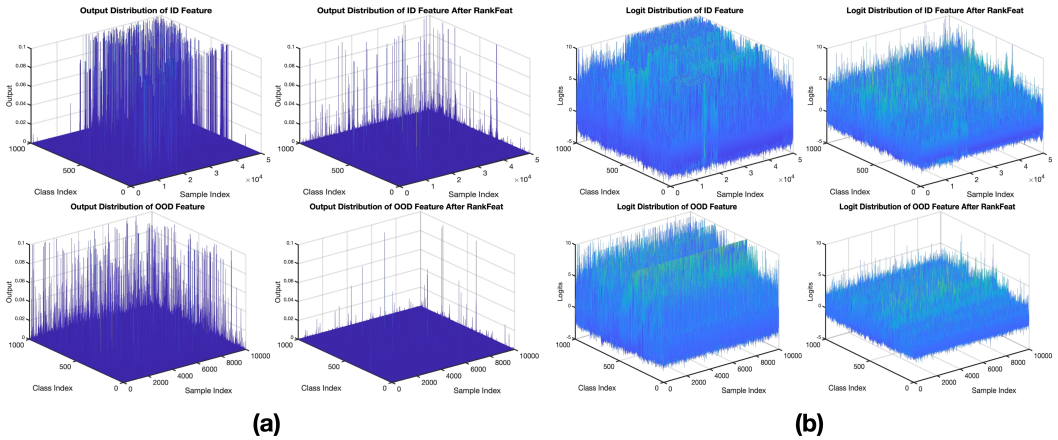

Figure 8: **(a)** Output distributions of `RankFeat`. **(b)** Logit distributions of `RankFeat`.

features are different. Here we give an intuitive conjecture. Since the network is well trained on the ID training set, when encountered with ID data, the feature matrix is likely to be more informative. Accordingly, more singular vectors would be active and the matrix energies spread over the corresponding singular values, leading to a more flat spectrum. On the contrary, for the unseen OOD data, the feature is prone to have a more compact representation, and less singular vectors might be active. In this case, the dominant singular value of OOD feature would be larger and would take more energies of the matrix. The informativeness can also be understood by considering applying PCA on the feature matrix. Suppose that we are using PCA to reduce the dimension of ID and OOD feature to 1. The amount of retained information can be measured by explained variance (%). The metric is defined as $\sum_{i=0}^{k} \mathrm{s}_i^2 / \sum_{j=0}^{n} \mathrm{s}_j^2$ where k denotes the projected dimension and n denotes the total dimension. It measures the portion of variance that the projected data could account for. We compute the average explained variance of all datasets and present the result in Table 11.

Table 11: The average explained variance ratio (%) of the ID and OOD datasets.

Dataset	ImageNet-1k	iNaturalist	SUN	Places	Textures
Explained Variance (%)	**28.57**	38.74	35.79	35.17	42.21

As can be observed, the OOD datasets have a larger explained variance ratio than the ID dataset. *That being said, to retain the same amount of information, we need fewer dimensions for the projection of OOD features. This indicates that the information of OOD feature is easier to be captured and the OOD feature matrix is thus less informative.*

As for how the training leads to the difference, we doubt that the well-trained network weights might cause and amplify the gap in the dominant singular value of the ID and OOD feature. To verify this guess, we compute the singular values distributions of the Google BiT-S ResNetv2-100 model [24, 39] with different training steps, as well as a randomly initialized network as the baseline.

Fig. 9 depicts the top-5 largest singular value distributions of the network with different training steps. Unlike the trained networks, the untrained network with random weights has quite a similar singular value distribution for the ID and OOD data. The singular values of both ID and OOD features are of similar magnitudes with the untrained network. However, when the number of training steps is increased, the gap of dominant singular value between ID and OOD feature is magnified accordingly. This phenomenon supports our conjecture that the well-trained network weights cause and amplify the difference of the largest singular value. Interestingly, our finding is coherent with [63]. In [63], the authors demonstrate that the classification accuracy of a model is highly correlated with its ability of OOD detection and open-set recognition. Training a stronger model could naturally improve the OOD detection performance. We empirically show that the gap of the dominant singular value is gradually amplifying as the training goes on, which serves as supporting evidence for [63].

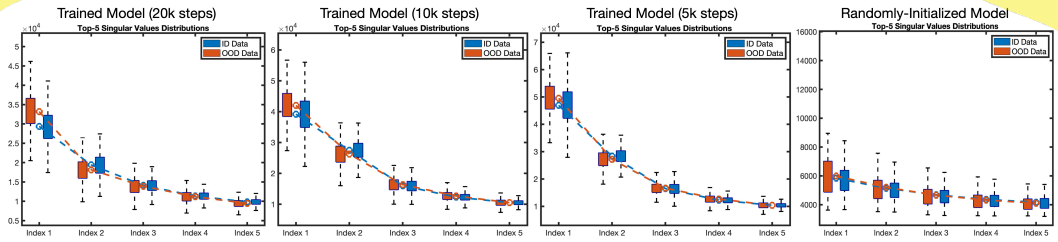

Figure 9: The top-5 largest singular value distributions of the pre-trained network with different training steps. For the untrained network initialized with random weights, the singular values distributions of ID and OOD feature exhibit very similar behaviors. As the training step increases, the difference between the largest singular value is gradually amplified.

F Theorem and Proof of Manchenko-Pastur Law

In the paper, we use the MP distribution of random matrices to show that removing the rank-1 matrix makes the statistics of OOD features closer to random matrices. For self-containment and readers' convenience, here we give a brief proof of Manchenko-Pastur Law.

Theorem 2. *Let* $\mathbf{X}$ *be a random matrix of shape* $t \times n$ *whose entries are random variables with* $E(\mathbf{X}_{ij} = 0)$ *and* $E(\mathbf{X}_{ij}^2 = 1)$. *Then the eigenvalues of the sample covariance* $\mathbf{Y} = \frac{1}{n}\mathbf{X}\mathbf{X}^T$ *converges to the probability density function:* $\rho(\lambda) = \frac{t}{n}\frac{\sqrt{(\lambda_+ - \lambda)(\lambda - \lambda_-)}}{2\pi\lambda\sigma^2}$ *for* $\lambda \in [\lambda_-, \lambda_+]$ *where* $\lambda_- = \sigma^2(1 - \sqrt{\frac{n}{t}})^2$ *and* $\lambda_+ = \sigma^2(1 + \sqrt{\frac{n}{t}})^2$.

Proof. Similar with the deduction of our bound analysis, the sample covariance $\mathbf{Y}$ can be written as the sum of rank-1 matrices:

$$\mathbf{Y} = \sum_{s=0}^{t} = \mathbf{Y}_n^s, \ \mathbf{Y}_n^s = \mathbf{U}_n^s \mathbf{D}_n^s (\mathbf{U}_n^s)^* \tag{25}$$

where $\mathbf{U}_n^s$ is a unitary matrix, and $\mathbf{D}_n^s$ is a diagonal matrix with the only eigenvalue $\beta = {}^n\!/_t$ for large n (rank-1 matrix). Then we can compute the Stieltjes transform of each $\mathbf{Y}_n^s$ as:

$$s_n(z) = \frac{1}{n}\mathrm{tr}(\mathbf{Y}_n^s - z\mathbf{I})^{-1} \tag{26}$$

Relying on Neumann series, the above equation can be re-written as:

$$s_n(z) = -\frac{1}{n}\sum_{k=0}^{\infty}\frac{\mathrm{tr}(\mathbf{Y}_n^s)^t}{z^{k+1}} = -\frac{1}{n}\Big(\frac{n}{z} + \sum_{k=1}^{\infty}\frac{\beta^k}{z^{k+1}}\Big) = -\frac{1}{n}\Big(\frac{n-1}{z} + \frac{1}{z-\beta}\Big) \tag{27}$$

Let $z := z_n(s)$ and we can find the function inverse of the transform:

$$nsz_n(s)^2 - n(s\beta - 1)z_n(s) - (n-1)\beta = 0 \tag{28}$$

The close-formed solution is calculated as:

$$\begin{aligned} z_n(s) &= \frac{n(s\beta - 1) \pm \sqrt{n^2(s\beta-1)^2 + 4n(n-1)s\beta}}{2ns} \\ &\approx \frac{1}{2ns}\Big(n(s\beta - 1) \pm \Big|n(s\beta + 1) - \cancel{\frac{2s\beta}{\beta + 1}}\Big|\Big) \end{aligned} \tag{29}$$

For large n, the term $\frac{2s\beta}{\beta+1}$ is sufficiently small and we can omit it. The solution is defined as:

$$z_n(s) = -\frac{1}{s} + \frac{\beta}{n(1 + s\beta)} \tag{30}$$

The R transform of each $\mathbf{Y}_n^s$ is given by:

$$R_{\mathbf{Y}_n^s}(s) = z_n(-s) - \frac{1}{s} = \frac{\beta}{n(1 - s\beta)} \tag{31}$$

Accordingly, the R transform for $\mathbf{Y}_n$ is given by:

$$R_{\mathbf{Y}}(s) = tR_{\mathbf{Y}_n^s}(s) = \frac{\beta t}{n(1-s\beta)} = \frac{1}{1-s\beta} \tag{32}$$

Thus, the inverse Stieltjes transform of $\mathbf{Y}$ is

$$z(s) = -\frac{1}{s} + \frac{1}{1+s\beta} \tag{33}$$

Then the Stieltjes transform of $\mathbf{Y}$ is computed by inverting the above equation as:

$$s(z) = \frac{-(z+\beta+1) + \sqrt{(z+\beta+1)^2 - 4\beta z}}{2z\beta} \tag{34}$$

Since $\beta = {}^{b}/_{t}$, finding the limiting distribution of the above equation directly gives the Manchenko-Pastur distribution:

$$\begin{gathered} \rho(\lambda) = \frac{t}{n}\frac{\sqrt{(\lambda_+ - \lambda)(\lambda - \lambda_-)}}{2\pi\lambda\sigma^2} \; for \; \lambda \in [\lambda_-, \lambda_+], \\ \lambda_- = \sigma^2(1 - \sqrt{\frac{n}{t}})^2, \lambda_+ = \sigma^2(1 + \sqrt{\frac{n}{t}})^2 \end{gathered} \tag{35}$$

The theorem is thus proved. □