# Activation Subspaces for Out-of-Distribution Detection

Barış Zöngür [1]     Robin Hesse [1]     Stefan Roth [1,2]

[1]TU Darmstadt     [2]hessian.AI

## Abstract

*To ensure the reliability of deep models in real-world applications, out-of-distribution (OOD) detection methods aim to distinguish samples close to the training distribution (in-distribution, ID) from those farther away (OOD). In this work, we propose a novel OOD detection method that utilizes singular value decomposition of the weight matrix of the classification head to decompose the model's activations into* decisive *and* insignificant *components, which contribute maximally, respectively minimally, to the final classifier output. We find that the subspace of* insignificant *components more effectively distinguishes ID from OOD data than raw activations in regimes of large distribution shifts (Far-OOD). This occurs because the classification objective leaves the insignificant subspace largely unaffected, yielding features that are "untainted" by the target classification task. Conversely, in regimes of smaller distribution shifts (Near-OOD), we find that activation shaping methods profit from only considering the* decisive *subspace, as the* insignificant *component can cause interference in the activation space. By combining two findings into a single approach, termed* ActSub*, we achieve state-of-the-art results in various standard OOD benchmarks.*[1]

## 1. Introduction

Improvements in the accuracy of deep neural networks (DNNs) continue to drive their deployment in a broad range of real-world settings. Consequently, ensuring the reliability of these models across a variety of conditions is becoming increasingly crucial. Since realistic applications can violate the closed-world assumption of typical DNNs, an active area of research is devoted to the development of out-of-distribution (OOD) detection methods [60]. OOD detection is concerned with distinguishing OOD inputs, deviating from any of the predefined class distributions, from in-distribution (ID) inputs that belong to the training distribution [17]. Methods for OOD detection usually utilize score functions [33, 36, 37] to output a single scalar for each input, indicating if that input is OOD or ID. These score
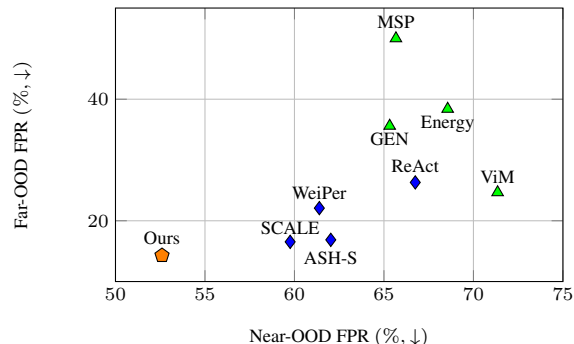
---

[1]The code is available at https://github.com/visinf/actsub/.



Figure 1. *False positive rate (FPR) for OOD detection when OOD inputs are close to (Near-OOD) and far from (Far-OOD) the ID domain.* Our approach outperforms established baselines, especially for more challenging Near-OOD examples.

functions can be based on model inherent information, such as logits [20, 36, 64], distances [15, 31, 35, 52], or (shaped) activations [11, 51, 57, 58].

In this work, we show that DNN activations can be decomposed into *decisive directions/components*, which contribute significantly to the classifier output, and *insignificant directions/components*, which have minimal effect on the model's output. Interestingly, activations still vary strongly across the insignificant subspace, which affects typical score functions for OOD detection. Based on this observation, we derive two important implications for OOD detection: First, while decisive directions are affected by the model's classification objective – yielding activations that specifically support the separation of the target classes (rather than ID from OOD data) – this effect is less pronounced for insignificant directions that are mostly ignored during training. As a result, the insignificant activation subspace is akin to powerful features from a random neural network [4, 10, 53], which allow to solve other classification tasks, such as OOD detection. While a similar proposition was already made in [8], we extend these results by utilizing a larger set of insignificant directions. Second, directions from the insignificant subspace can interfere with the decisive subspace, causing different activations to produce the same output. This leads to them being handled inconsistently by activation shaping methods, which is undesirable.

To make use of these implications, we decompose the weight matrix of the linear classification head in a DNN with singular value decomposition, allowing us to decompose the weight matrix into decisive and insignificant subspaces. The first observation is realized by computing score functions after projecting activations into the *insignificant* subspace, *i.e.*, when decisive directions are pruned from the activations. The second implication is addressed by applying activation shaping methods on the projection of activations into the *decisive* subspace, *i.e.*, when interfering insignificant directions are pruned from the activations. In a wide range of experiments, we demonstrate that the proposed activation subspaces drastically improve OOD detection, making our approach state of the art (SotA).

Specifically, our contributions are as follows: *(i)* We decompose activations into orthogonal *decisive* and *insignificant* subspaces using singular value decomposition of the weight matrix of the classification head. *(ii)* We show that the insignificant components provide a powerful feature space for OOD detection. *(iii)* We reduce the influence of interference between insignificant and decisive directions in the activation space by selectively applying activation shaping to the decisive subspace. *(iv)* We combine the energy score from the shaped decisive subspace and the cosine similarity score from the insignificant components. *(v)* Our empirical findings suggest that when applied in combination, scores from the two activation subspaces significantly outperform previous baselines, achieving SotA results.

## 2. Related Work

**Preliminaries.**   OOD detection can be reduced to estimating a score function that separates ID and OOD inputs. A binary decision function $G(\cdot)$ determines if an input $\mathbf{x}$ is OOD ($G = 1$) for model $F$ using a threshold $\tau$ and score function $S(\cdot)$ via $G(\mathbf{x}, F, \tau) = [S(\mathbf{x}, F) \geq \tau]$, with $[\cdot]$ denoting the Iverson bracket [28]. To improve OOD detection, **training time** regularizations [18, 24, 38, 39, 46] aim to constrain the activation space to increase ID-OOD separability. On the other hand, **post-hoc** methods utilize an already trained model and its available training data [11, 36, 52, 55]. We focus on post-hoc methods, which can be further categorized by the information used to construct the score function.

**Distance-based methods** typically rely on the activation's position in the embedding space. The distance of that position to the estimated ID distribution can be measured with a Mahalanobis distance [31]. NuSA utilizes the norm of the projection of the activation to the null space [8]. KNN measures the average Euclidean distance of the normalized activation [52]. Recently, it has been shown that distances of activations to decision boundaries are discriminative for OOD detection [35]. Alternatively, PCA-based methods use the reconstruction error as score function [13, 15, 55]. For

ID data, one expects low coverage over residual directions of PCA, as those are the directions with low expected variance. Consequently, if the norm of the residual, hence the reconstruction error, is high, the input is detected as OOD. We emphasize that the directions deemed insignificant in our work are rather distinct from PCA residuals, as our directions are determined by their contribution to the decision, not the projection's variance.

**Logit-based methods** utilize the model's prediction instead of the activation [17, 20, 36, 64]. For example, one can use the prediction's confidence [17] or the information entropy of the prediction [5, 37] as the score function [17]. Similarly, the free Helmholtz energy [36], simply referred to as *energy*, is highly used in the literature [11, 14, 21, 34, 51, 66]. PCA-based methods commonly combine the reconstruction error with the energy [15, 55]. NNGuide uses the energy as a guidance term for cosine similarity [41]. SHE uses the energy function of a Hopfield network to store class patterns [62]. Activation shaping methods utilize the energy of the shaped activations [1, 11, 51, 57, 58, 61].

**Activation shaping methods** aim to actively identify and mitigate activation channels that decrease ID-OOD separability. ReAct uses truncation to mitigate abnormal OOD activations [51]. BATS uses inherent information in batch normalization layers [26] to determine the range for truncation [66]. RankFeat removes the direction of the highest variance, estimated from the data matrix [48]. VRA targets both abnormally high and low activations using a variational method [58]. ASH selects top channels while pruning the rest [11], applying exponential scaling to top activations. SCALE further analyses and extends ASH, demonstrating that the key factor in ASH's effectiveness is scaling rather than pruning [57]. In parallel, methods, such as DICE, target the weight matrix to eliminate abnormal channels [50]. LINe uses Shapley [47] values to identify important neurons [1]. DDCS leverages the discriminability of neurons when selecting the channels to prune [61]. SeTAR uses low-rank approximation to refine the weight matrix, primarily targeting vision-language models [32].

**Gradient-based methods** investigate the gradients of the model weights. GradNorm uses the norm of the gradients [25]. GradOrth uses the projection of gradient to the singular directions of the data matrix, following a similar motivation as PCA methods [2]. GAIA utilizes abnormalities in gradient-based attributions [6].

## 3. OOD Detection in Activation Subspaces

In this work, we propose a novel algorithm, termed *ActSub*, for OOD detection that leverages decisive and insignificant activation subspaces of a DNN. We will first convey the intuition behind our approach and then outline the algorithm in detail in Sec. 3.1.
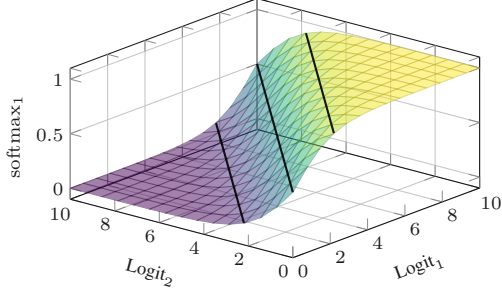
Figure 2. *Softmax output (softmax$_1$) for the class associated with Logit$_1$, assuming a setting with two classes.* The black lines represent contour lines for the output values 0.1, 0.5, and 0.9. Note how they are parallel, illustrating that there is a direction in the logit space that does not affect the softmax output.

**Insignificant directions for OOD detection.** We let $F\colon \mathbb{R}^m \mapsto \mathbb{R}^n$ denote a DNN that maps an input $\mathbf{x} \in \mathbb{R}^m$ to a feature activation $\mathbf{a} \in \mathbb{R}^n$. To classify $\mathbf{x}$, such an activation is projected onto class logits $\mathbf{l} \in \mathbb{R}^c$ via a linear transformation $\mathbf{l} = \mathbf{W}\mathbf{a}$ with $\mathbf{W} \in \mathbb{R}^{c \times n}$ (we omit the bias $\mathbf{b}$ for simplicity of exposition and empirical insignificance). Our intuition behind using activation subspaces is based on a simple observation that was also discussed in [8, 55]: there are directions in the activation space of a DNN that have only a minimal impact on the classification output. With a grain of salt, we call these directions *insignificant* while they turn out indeed important for OOD detection.

The existence of these directions is easy to show with the nullspace of $\mathbf{W}$ [8] – the nullspace of a matrix $\mathbf{W}$ consists of all vectors $\mathbf{q}$ such that $\mathbf{W}\mathbf{q} = 0$ [49]. For any matrix $\mathbf{W} \in \mathbb{R}^{c \times n}$ with $n > c$, the nullspace is guaranteed to be non-trivial, and we let $\mathbf{Q}$ denote the orthogonal basis spanning that nullspace. For example, in an ImageNet-1k [9, 44] trained ResNet-50 [16], the activation $\mathbf{a}$ is in $\mathbb{R}^{n=2048}$ while the class logits $\mathbf{l}$ are in $\mathbb{R}^{c=1000}$. Since 2048>1000, the weight matrix $\mathbf{W}$ of the model has a non-trivial nullspace and, therefore, there are directions $\mathbf{q}$ in the column space of $\mathbf{Q}$ that do not affect the model's classification output.

The existence of such insignificant directions has two interesting implications for OOD detection. First, activations along these directions still encode input statistics, making them particularly useful for OOD detection [8]. We posit that these insignificant directions are helpful because DNNs tend to enforce activations to align with known classification directions due to the cross-entropy loss, which maximizes the class posterior [27]. This presents a challenge for OOD detection because OOD activations are also biased toward high alignment with the known directions, making them difficult to distinguish from in-domain (ID) activations. Second, the insignificant directions remain "invisible" to the loss during training – since the output and the gradients along these directions are constant, the model retains its behavior from the random initialization of weights.

In this insignificant activation subspace, the DNN thus behaves akin to a random neural network, which are known to support challenging classification tasks [10]. We exploit this for OOD detection.

**Proposition 3.1.** *Insignificant directions in the activation space that only marginally affect the output of a classifier can be useful for distinguishing OOD from ID activations.*

Until now, our presented intuition mostly aligns with that of [8]; however, we go beyond their work by considering a larger set of insignificant directions. Further, we identify a novel second implication of the existence of insignificant directions: Different activations along the insignificant directions yield the same model output, while individual activation elements $\mathbf{a}_i$ may change. Consequently, the highest activating elements can vary, potentially impacting the response of activation shaping methods [11, 51, 57, 58]. This inconsistency can lead to different prunings for activations that result in the same model output, which is undesirable.

**Proposition 3.2.** *Insignificant directions in the activation space may interfere with decisive directions, potentially harming activation shaping methods.*

### 3.1. Going beyond the nullspace of weights

Now that we have established two propositions claiming that insignificant directions in the activation space of a DNN are helpful for OOD detection, we extend our analysis by exploring additional insignificant directions.

**The softmax-invariant direction.** We assume that a classification model was trained using a cross-entropy loss, where the softmax function of the logits $\mathbf{l}$ is computed as

$$\mathrm{softmax}(\mathbf{l})_i = \frac{e^{\mathbf{l}_i}}{\sum_{j=1}^c e^{\mathbf{l}_j}}. \tag{1}$$

When plotting the softmax output for the two-dimensional case (*cf*. Fig. 2), it becomes directly evident that there is a direction where the softmax output remains constant (denoted by the parallel black lines). It is easy to show that this direction is given by $\mathbf{1} = (1, \dots, 1)^T \in \mathbb{R}^n$ for the general $n$-dimensional case. Due to the linearity of $\mathbf{W} \in \mathbb{R}^{c \times n}$, there is also a direction $\mathbf{p}$ in the activation space of the model where the softmax output remains constant when adding that direction to any activation vector $\mathbf{a}$:

$$\mathrm{softmax}(\mathbf{W}\mathbf{a})_i = \mathrm{softmax}(\mathbf{W}(\mathbf{a} + \alpha\mathbf{p}))_i \quad \forall \alpha \in \mathbb{R}. \tag{2}$$

Specifically, that direction is given by $\mathbf{p} = \mathbf{W}^\dagger \mathbf{1}$, with $\mathbf{W}^\dagger$ denoting the pseudoinverse of $\mathbf{W}$. Note that by definition, $\mathbf{p}$ is not in the null space of $\mathbf{W}$, since $\mathbf{W}\mathbf{p} = \mathbf{1} \neq 0$.

**Relaxing insignificant directions.** Directions in the nullspace of $\mathbf{W}$ and the above softmax-invariant direction do not change the output of the model *at all* when added
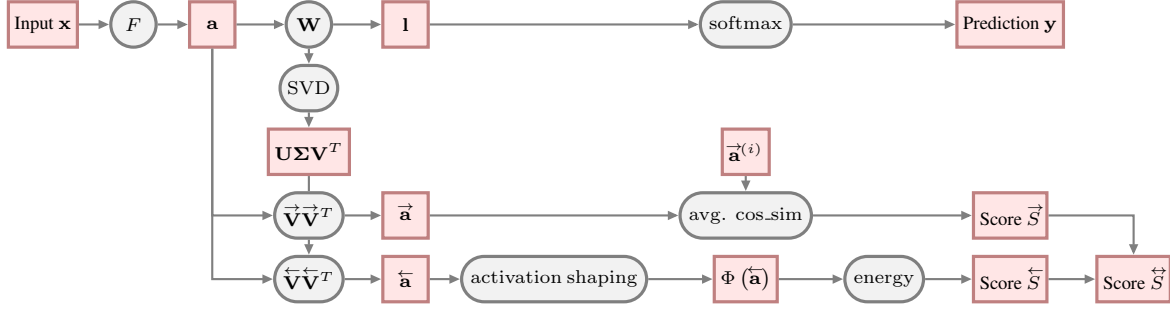
Figure 3. *Flow diagram of our proposed ActSub algorithm.* Briefly, an input $\mathbf{x}$ is fed through a model $F$ to obtain the activation $\mathbf{a}$, which is decomposed into the insignificant component $\vec{\mathbf{a}}$ and the decisive component $\overleftarrow{\mathbf{a}}$ using SVD on the weight matrix $\mathbf{W}$. Our final score $\overleftrightarrow{S}$ is a combination of $\vec{S}$ (obtained by comparing $\vec{\mathbf{a}}$ to insignificant activations from the training set $\vec{\mathbf{a}}^{(i)}$) and $\overleftarrow{S}$ (obtained by applying activation shaping $\Phi(\cdot)$ on $\overleftarrow{\mathbf{a}}$). Please refer to Sec. 3.2 for a detailed explanation.

to any activation. However, Propositions 3.1 and 3.2 also apply for insignificant directions that only *marginally* affect the model output. Thus, in a more general sense, we are interested in finding all directions in the activation space that only slightly affect the classification output. To find such directions, we make use of the singular value decomposition (SVD) by factorizing the classifier weight matrix as

$$\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T. \qquad (3)$$

Without loss of generality, we assume that the singular values $\mathbf{\Sigma}_{ii}$ are in descending order. Intuitively, the rows in $\mathbf{V}^T$ build an orthonormal basis where the first rows correspond to the most important, *i.e.* decisive, subspace of $\mathbf{W}$, and the last rows to the least important, *i.e.* insignificant, subspace. The corresponding singular values indicate *how* important the respective directions are. Although the SVD in Eq. (3) does not explicitly include the softmax function, we empirically find that the right singular vector corresponding to the smallest non-negative singular value almost perfectly aligns with the softmax-invariant direction, $\mathbf{p} = \mathbf{W}^\dagger \mathbf{1}$, described earlier, for many backbone models. We believe this property emerges during training with the softmax, as this direction is irrelevant to the loss, leading to $\mathbf{W}$ naturally ignoring it. Further, the right singular vectors corresponding to the zero singular values lie in the nullspace of $\mathbf{W}$. These two findings nicely connect the previously outlined theory with our actual method and further validate our approach.

With $\mathbf{W}$ being decomposed into orthonormal directions of decreasing importance, we define two auxiliary bases

$$\overleftarrow{\mathbf{V}}^T = \begin{cases} \mathbf{v}_i^T & \text{if } i \leq k \\ \mathbf{0}^T & \text{otherwise} \end{cases}, \quad \vec{\mathbf{V}}^T = \begin{cases} \mathbf{0}^T & \text{if } i \leq k \\ \mathbf{v}_i^T & \text{otherwise} \end{cases}, \quad (4)$$

for subspaces that only contain the decisive, respectively insignificant, directions of $\mathbf{V}^T$. $\mathbf{v}_i$ denotes the $i^{\text{th}}$ column in $\mathbf{V}$. We can now express an activation $\mathbf{a}$ in terms of these bases, effectively pruning the insignificant, respectively decisive, directions from $\mathbf{a}$:

$$\overleftarrow{\mathbf{a}} = \overleftarrow{\mathbf{V}}\overleftarrow{\mathbf{V}}^T \mathbf{a}, \qquad\qquad \vec{\mathbf{a}} = \vec{\mathbf{V}}\vec{\mathbf{V}}^T \mathbf{a}. \qquad (5)$$

Since $\overleftarrow{\mathbf{a}}$ is mostly relevant for the decision of the model, we call it the *decisive component*. Correspondingly, we call $\vec{\mathbf{a}}$ the *insignificant component*.

The hyperparameter $k$ in Eq. (4) specifies the number of directions in $\mathbf{V}^T$ that are considered decisive/insignificant. In other words, $k$ determines the degrees of freedom of the respective subspace. Each additional direction increases the expected norm of the projected representation; thus the norm can serve as a proxy for the amount of information captured in each subspace. We select $k$ such that the norms of the decisive ($\overleftarrow{\mathbf{a}}^{(i)}$) and insignificant ($\vec{\mathbf{a}}^{(i)}$) components of activations $\mathbf{a}^{(i)}$ of the training samples $\mathbf{x}^{(i)}$ are as close as possible:

$$\arg\min_k \left( \frac{1}{N} \sum_{i=1}^N \left( \|\vec{\mathbf{a}}^{(i)}\| - \|\overleftarrow{\mathbf{a}}^{(i)}\| \right) \right). \qquad (6)$$

Our approach eliminates the hyperparameter search on $k$, which can otherwise vastly differ for different models and numbers of classes. We provide further analysis in the supplement.

### 3.2. Utilizing the activation subspaces

Building on the above approach to extract the decisive, respectively insignificant, directions of the activation $\mathbf{a}$, we now outline our OOD detection algorithm, combining two score functions that are based on Propositions 3.1 and 3.2.

**Comparing directions.** In Proposition 3.1, we argued that OOD and ID activations should be easier to distinguish in the insignificant subspace that only has a minor impact on the model output. Using the above formulation, we can directly formalize this idea with a score function $\vec{S}$. We extend the implementation of NNGuide [41] to insignificant components and calculate the average cosine similarity of pruned test-time input activations to pruned activations from training samples; contrary to NNGuide, we

omit the guidance term. More specifically, we iterate over a small random subset (10%) of the training dataset and compute the average cosine similarity $\cos\_\mathrm{sim}(\cdot)$ between the insignificant component $\vec{\mathbf{a}}$ and insignificant components of the top $N$ samples $\vec{\mathbf{a}}^{(i)}$ from the randomly sampled training subset with the highest similarity to $\vec{\mathbf{a}}$:

$$\vec{S} = -\log\left(1 - \frac{1}{N}\sum_{i=1}^{N}\cos\_\mathrm{sim}(\vec{\mathbf{a}}^{(i)}, \vec{\mathbf{a}})\right). \quad (7)$$

While the average cosine similarity is a discriminative measure, it lies within a tight range, causing issues when combining it with our second score function $\overset{\leftarrow}{S}$ (see below). To address this, we use $\log(1 - \dots)$ on the average cosine similarity to expand the range of the values and bring both score functions into similar ranges.

**Activation shaping.** Using the above formulations, the logit output of our model can be approximated as

$$\mathbf{U\Sigma V}^T\mathbf{a} \approx \mathbf{U\Sigma}\overset{\leftarrow}{\mathbf{V}}^T\overset{\leftarrow}{\mathbf{a}}. \quad (8)$$

Recalling Proposition 3.2, $\overset{\leftarrow}{\mathbf{a}}$ only contains decisive directions, which should improve activation shaping methods $\Phi(\cdot)$ by reducing the interference from insignificant directions. We thus compute the second score function $\overset{\leftarrow}{S}$ of our approach as the energy $E$ [36] over $K$ logits:

$$
\begin{aligned}
\overset{\leftarrow}{S} &= -E\left(\mathbf{U\Sigma}\overset{\leftarrow}{\mathbf{V}}^T\Phi(\overset{\leftarrow}{\mathbf{a}})\right)\\
&= -\log\sum_{j}^{K}\exp\left(\left(\mathbf{U\Sigma}\overset{\leftarrow}{\mathbf{V}}^T\Phi(\overset{\leftarrow}{\mathbf{a}})\right)_j\right).
\end{aligned} \quad (9)
$$

The final score function $\overset{\leftrightarrow}{S}$ of our proposed **ActSub** algorithm for OOD detection is defined as

$$\overset{\leftrightarrow}{S} = \vec{S}^{\lambda}\cdot\overset{\leftarrow}{S}, \quad (10)$$

with $\lambda$ being a scalar that modulates the two score functions and is estimated via a hyperparameter search on a small range of values on the validation split (see supplement). Fig. 3 provides a flow diagram for our proposed approach.

**Distinction to PCA-based approaches.** At first glance, our proposed method may appear similar to PCA-based OOD detection methods [13, 15, 55]. They project activations onto the directions with the lowest variance, as determined by applying PCA to the activation covariance matrix of the in-distribution training samples. However, these methods are fundamentally different from our proposed approach. Theoretically, there is no direct connection between the directions of the smallest variance in the activations and those that only marginally affect the linear classifier. However, the model could still learn to align these directions. To demonstrate that this is not the case, in Fig. 4, we show the
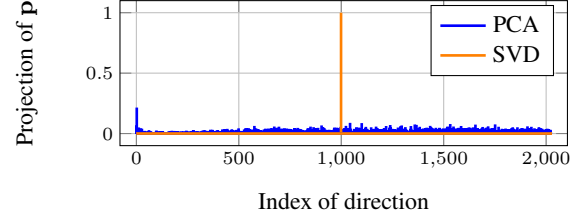


Figure 4. *Projection of the softmax-invariant direction* $\mathbf{p}$ *from Eq.* (2) *onto the right singular vectors of the weight matrix (SVD) or the eigenvectors of the activation covariance (PCA).* The $x$-axis represents the respective directions in descending order of their singular values, respectively, eigenvalues. The $y$-axis shows the magnitude of the projection onto each of the corresponding directions. The softmax-invariant direction almost perfectly aligns with the right singular vector corresponding to the smallest non-negative singular value. For PCA, it aligns with many directions of high variance, especially the highest variance one.

projection of the softmax-invariant direction $\mathbf{p}$ from Eq. (2) to the right singular vectors of the weight matrix (SVD) and the eigenvectors of the activation covariance (PCA). We use an ImageNet-1k [9, 44] pre-trained ResNet-50 [16] with an activation dimension of $n = 2048$ and samples from the ImageNet-1k training split with 1000 classes. For SVD on the weight matrix $\mathbf{W}$, the softmax-invariant direction perfectly aligns with the direction of the smallest non-negative singular value. However, for PCA on the training activations, the softmax-invariant direction shows high alignment with many directions of high variance. Interestingly, the alignment is particularly high with the direction of the largest variance.

## 4. Experiments

**Metrics.** We report results on two established standard metrics for OOD detection [17]: the area under the receiver operating-characteristic curve (AUC), where higher values indicate better accuracy, and the false positive rate (FPR) of the OOD inputs, where the true positive rate of ID inputs is 95 %. For FPR, lower indicates better results.

### 4.1. Main results

To demonstrate the effectiveness of our method, we follow the standard setting in the literature [24, 51] and report the OOD detection accuracies for an ImageNet-1k [9, 44] trained ResNet-50 [16] for OOD data from iNaturalist [22], SUN [56], Places [65], and Textures [7] in Tab. 1. We use the ResNet-50 checkpoint of torchvision [42] to ensure reproducibility and fair comparison. We compare our method against established and state-of-the-art methods, focusing on different method families, such as distance-based, logit/energy-based, and activation shaping-based approaches. To ensure fair comparisons to the respective method families, we not only report results for our fi-

| | Method | iNaturalist | | SUN | | Places | | Textures | | Average | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | AUC (↑) | FPR (↓) | AUC (↑) | FPR (↓) | AUC (↑) | FPR (↓) | AUC (↑) | FPR (↓) | AUC (↑) | FPR (↓) |
| Distance | Mahalanobis [31] | 52.65 | 97.00 | 42.41 | 98.50 | 41.79 | 98.40 | 85.01 | 55.80 | 55.47 | 87.43 |
| | KNN [55] | 86.47 | 59.00 | 80.72 | 68.82 | 75.76 | 76.28 | 97.07 | 11.77 | 85.01 | 53.97 |
| | ViM [52] | 87.42 | 71.85 | 81.07 | 81.79 | 78.40 | 83.12 | 96.83 | 14.88 | 85.93 | 62.91 |
| | CoRP [13] | 89.32 | 50.07 | 83.74 | 62.56 | 78.91 | 72.76 | 98.14 | 9.02 | 87.53 | 48.60 |
| | fDBD [35] | 93.67 | 40.24 | 86.97 | 60.60 | 84.27 | 66.40 | 92.12 | 37.50 | 89.26 | 51.19 |
| | $\vec{S}$ (ours) | 91.77 | 42.80 | 87.43 | 56.71 | 82.08 | 69.23 | **98.33** | **8.53** | 89.90 | 44.32 |
| Logit | MSP [17] | 88.17 | 51.44 | 79.95 | 72.04 | 78.84 | 74.34 | 54.90 | 78.69 | 81.41 | 63.18 |
| | ODIN [33] | 91.32 | 41.07 | 84.71 | 64.63 | 81.95 | 68.36 | 85.77 | 50.55 | 85.94 | 56.15 |
| | Energy [36] | 89.95 | 55.72 | 85.89 | 59.26 | 82.86 | 64.92 | 85.99 | 53.72 | 86.17 | 58.41 |
| | NNGuide [41] | 95.12 | 25.73 | 91.21 | 37.18 | 88.67 | 46.97 | 92.30 | 27.70 | 91.82 | 34.39 |
| | NRE [15] | 91.09 | 50.36 | 87.55 | 54.19 | 84.00 | 64.13 | 92.59 | 29.33 | 88.81 | 49.50 |
| | DML [64] | 91.61 | 47.32 | 86.14 | 57.40 | 84.68 | 61.43 | 86.72 | 52.80 | 87.28 | 54.74 |
| | SHE [62] | 90.18 | 34.22 | 84.69 | 54.19 | 90.15 | 45.35 | 87.93 | 45.09 | 88.24 | 44.71 |
| | CoRP [13] | 95.15 | 26.85 | 90.76 | 40.38 | 87.35 | 51.26 | 97.17 | 12.11 | 92.61 | 32.65 |
| | $-E \cdot \vec{S}$ (ours) | 95.21 | 26.30 | 91.20 | 39.82 | 87.63 | 51.84 | 97.43 | 10.35 | 92.87 | 32.08 |
| Activation shaping | ReAct [51] | 96.22 | 20.38 | 94.20 | 24.20 | 91.58 | 33.85 | 89.80 | 47.30 | 92.95 | 31.43 |
| | BATS [66] | 97.67 | 12.57 | 95.33 | 22.62 | 91.83 | 34.34 | 92.27 | 38.90 | 94.28 | 27.11 |
| | DICE [50] | 96.24 | 18.64 | 93.94 | 25.45 | 90.67 | 36.86 | 92.74 | 28.07 | 93.40 | 27.25 |
| | NRE [15] | 97.97 | 10.17 | 95.80 | 18.50 | 93.39 | 27.31 | 95.95 | 18.67 | 95.76 | 18.66 |
| | VRA [58] | 97.12 | 15.70 | 94.25 | 26.94 | 91.27 | 37.85 | 95.62 | 21.47 | 94.57 | 25.49 |
| | LINe [1] | 97.56 | 12.26 | 95.26 | 19.48 | 92.85 | 28.52 | 94.44 | 22.54 | 95.03 | 20.70 |
| | ASH-S [11] | 97.87 | 11.49 | 94.02 | 27.98 | 90.98 | 39.78 | 97.60 | 11.93 | 95.12 | 22.80 |
| | CoRP [13] | 97.85 | 10.77 | 95.75 | 18.70 | 93.13 | 28.69 | 97.21 | 12.57 | 95.98 | 17.68 |
| | fDBD [35] | 98.07 | 10.19 | 94.87 | 24.58 | 92.00 | 36.12 | 97.48 | 12.52 | 95.61 | 20.85 |
| | DDCS [61] | 97.85 | 11.63 | 95.68 | 18.63 | 92.89 | 28.78 | 95.77 | 18.40 | 95.55 | 19.36 |
| | SCALE [57] | 98.17 | 9.50 | 95.02 | 23.27 | 92.26 | 34.51 | 97.37 | 12.93 | 95.71 | 20.05 |
| | NN-SCALE | 98.04 | 10.08 | 94.72 | 24.12 | 91.74 | 35.58 | 97.53 | 12.50 | 95.51 | 20.57 |
| | $\overleftarrow{S}$ (ours) | **98.67** | **6.71** | **96.48** | **15.16** | **95.05** | **21.95** | 89.09 | 43.48 | 94.80 | 21.82 |
| | $\overleftrightarrow{S}$ ActSub (ours) | 98.51 | 7.19 | 96.40 | 16.73 | 94.40 | 25.64 | 97.66 | 11.01 | **96.74** | **15.14** |

Table 1. *OOD detection accuracies for ResNet-50 trained on ImageNet-1k* (all in %). We consider three model families. For baseline methods that are mentioned multiple times, we report the version that corresponds to each model family. For each setting, we select the top results from the respective paper (*e.g.*, for fDBD [35], we consider the variant with SCALE [57]).

nal method ($\overleftrightarrow{S}$), but also separate it into its components: for distance-based approaches, we report OOD detection results when only considering the cosine similarities of the insignificant component ($\vec{S}$); for logit/energy-based approaches, we replace the decisive component ($\overleftarrow{S}$) of $\overleftrightarrow{S}$ with energy over the logits ($-E(\mathbf{Wa}) \cdot \vec{S}$); and for activation shaping-based approaches, we also consider only using activation shaping on the decisive component ($\overleftarrow{S}$). Note that method families are not strictly defined and may overlap; rather, they indicate the primary focus of each respective method. If not stated otherwise, we use the current SotA activation shaping method SCALE [57] to define $\Phi$ in $\overleftarrow{S}$ (*cf*. Eq. 9); however, as we show in Tab. 4, other activation shaping methods can be used.

Remarkably, the different components of our method achieve superior accuracy in each respective method family. We achieve the highest average values among distance-based approaches, particularly excelling in detecting OOD samples from the Textures dataset. We increase the AUC by 0.64 % points and reduce the FPR by 6.87 % points on average compared to the closest baseline (in terms of AUC) fDBD [35]. Similarly, we outperform other methods when focusing on logit/energy-based approaches, outperforming NNGuide [41] by 1.05 %/2.31 % points for AUC/FPR, re-

spectively. Our approach achieves SotA results among activation shaping methods on iNaturalist, SUN, and Places. However, the accuracy on Textures lags significantly, which impacts the overall average. When considering our final method ActSub ($\overleftrightarrow{S}$) and not only its individual components, we achieve SotA results on all datasets. Specifically, compared to the recent SCALE [57], our method increases the AUC by 1.03 % and reduces the FPR by 4.91 % points.

We hypothesize that combining both components works particularly well because the activation shaping part suffers on Textures while the distance part recovers the accuracy when combined. Both target different aspects of the distribution shift, and thus, they complement each other when combined. Additionally, we evaluate NNGuide [41] combined with the current SotA activation shaping-method SCALE (denoted as NN-SCALE) to demonstrate that a straightforward approach of combining SCALE with guided cosine similarity does not suffice.

## 4.2. Further backbones and datasets

To ensure that our findings generalize to different backbones and datasets, we next conduct a number of further experiments analogous to Sec. 4.1.

**MobileNetV2.** In Tab. 2, we report results on iNatural-

| Method | iNaturalist AUC (↑) | FPR (↓) | SUN AUC (↑) | FPR (↓) | Places AUC (↑) | FPR (↓) | Textures AUC (↑) | FPR (↓) | Average AUC (↑) | FPR (↓) |
|---|---|---|---|---|---|---|---|---|---|---|
| MSP [17] | 85.71 | 63.09 | 76.01 | 79.67 | 75.51 | 81.47 | 76.49 | 75.12 | 78.43 | 74.84 |
| Energy [36] | 88.91 | 59.50 | 84.50 | 62.65 | 81.19 | 69.37 | 85.03 | 58.05 | 84.91 | 62.39 |
| ODIN [33] | 91.33 | 45.61 | 83.44 | 63.03 | 80.85 | 70.01 | 85.61 | 52.45 | 85.31 | 57.78 |
| ReAct [51] | 92.72 | 43.07 | 87.26 | 52.47 | 84.07 | 59.91 | 90.96 | 40.20 | 88.75 | 48.91 |
| BATS [66] | 94.33 | 31.56 | 90.21 | 41.68 | 86.26 | 52.43 | 90.76 | 38.69 | 90.39 | 41.09 |
| DICE [50] | 93.57 | 32.30 | 92.86 | 31.22 | 88.02 | 46.78 | 96.25 | 16.28 | 92.68 | 31.64 |
| NRE [15] | 93.66 | 35.84 | 90.77 | 40.35 | 86.76 | 52.38 | 95.39 | 18.44 | 91.65 | 36.75 |
| LINe [1] | 95.53 | 24.95 | 92.94 | 33.19 | 88.98 | 47.95 | 97.05 | 12.30 | 93.62 | 29.60 |
| ASH-S [11] | 91.94 | 39.10 | 90.02 | 43.62 | 84.73 | 58.84 | **97.10** | **13.12** | 90.95 | 38.67 |
| NNGuide [41] | 91.19 | 45.73 | 87.87 | 51.03 | 84.44 | 60.60 | 92.47 | 29.50 | 88.99 | 46.72 |
| CoRP [13] | 94.27 | 31.72 | 90.98 | 40.77 | 86.42 | 55.69 | 97.49 | 10.48 | 92.29 | 34.66 |
| DDCS [61] | 96.87 | 17.44 | 95.83 | 17.42 | 91.80 | 30.49 | 94.86 | 25.11 | 94.84 | 22.61 |
| SCALE [57] | 94.43 | 30.26 | 91.67 | 38.56 | 87.35 | 53.28 | 96.66 | 14.63 | 92.52 | 34.18 |
| $\overrightarrow{S}$ (ours) | 70.43 | 82.72 | 73.12 | 83.53 | 64.83 | 91.89 | 96.42 | 17.27 | 76.20 | 68.85 |
| $\overleftarrow{S}$ (ours) | **97.67** | **12.54** | **96.24** | **17.30** | **93.51** | **28.66** | 92.58 | 36.21 | 95.00 | 23.68 |
| $\overleftrightarrow{S}$ **ActSub** (ours) | 97.43 | 13.90 | 96.02 | 18.31 | 93.08 | 30.26 | 94.95 | 24.34 | **95.37** | **21.70** |

Table 2. *OOD detection accuracies for MobileNetV2 trained on ImageNet-1k* (all in %). We report each of our components separately and in combination. For baseline methods, we consider best-reported results with this setting, *e.g.*, in combination with a previous method.

| Method | CIFAR10 AUC (↑) | FPR (↓) | CIFAR100 AUC (↑) | FPR (↓) |
|---|---|---|---|---|
| Energy [36] | 92.53 | 35.55 | 77.39 | 81.34 |
| ReAct [51] | 93.77 | 35.31 | 80.24 | 77.38 |
| DICE [50] | 93.09 | 30.21 | 83.15 | 62.00 |
| LINe [1] | 95.75 | 20.88 | 84.63 | 45.74 |
| ASH-S [11] | 95.25 | 21.12 | 87.76 | 47.89 |
| DDCS [61] | 96.05 | 19.35 | 86.73 | 43.05 |
| SCALE [57] | 96.16 | 17.84 | 88.44 | 45.86 |
| $\overleftrightarrow{S}$ **ActSub** (ours) | **96.87** | **15.37** | **89.35** | **41.26** |

Table 3. *CIFAR OOD detection accuracies for DenseNet-101* (all in %). We report average accuracy over four OOD datasets when CIFAR10 and CIFAR100 are ID.

| Method | Near-OOD (Avg.) AUC (↑) | FPR (↓) | Far-OOD (Avg.) AUC (↑) | FPR (↓) |
|---|---|---|---|---|
| MSP [17] | 76.02 | 65.67 | 85.23 | 51.47 |
| Energy [36] | 75.89 | 68.56 | 89.47 | 38.40 |
| ReAct [51] | 77.38 | 66.75 | 93.67 | 26.31 |
| RankFeat [48] | 50.99 | 91.83 | 53.93 | 87.17 |
| ViM [52] | 72.08 | 71.35 | 92.68 | 24.67 |
| SHE [62] | 73.78 | 73.01 | 90.92 | 41.45 |
| GEN [37] | 76.85 | 65.32 | 89.76 | 35.61 |
| ASH-S [11] | 79.63 | 62.03 | 96.47 | 16.86 |
| WeiPer+KLD [14] | 80.05 | 61.39 | 95.54 | 22.08 |
| SCALE [57] | 81.36 | 59.76 | 96.53 | 16.53 |
| $\overleftrightarrow{S}$ **ActSub** w/ ReAct | 77.70 | 65.36 | 96.56 | 15.57 |
| $\overleftrightarrow{S}$ **ActSub** w/ ASH-S | 84.06 | **52.46** | 96.91 | **14.22** |
| $\overleftrightarrow{S}$ **ActSub** w/ SCALE | **84.24** | 52.60 | **96.96** | 14.29 |

Table 4. *OpenOOD OOD detection accuracies for ResNet-50 trained on ImageNet-1k* (all in %). We report average Near- and Far-OOD accuracy.

ist, SUN, Places, and Textures for an ImageNet-1k trained MobileNetV2 [45] (from torchvision). Since evaluations on this backbone are less established in related work than for ResNet-50, with fewer reported results, we include a smaller set of baselines. However, to ensure fair and complete comparisons, we reproduce key baselines, such as SCALE. Confirming the results of the ResNet-50 setup, our combined method achieves SotA results on MobileNet, with the activation shaping variant $\overleftarrow{S}$ lagging behind in Textures while the distance variant $\overrightarrow{S}$ excels on Textures.

**CIFAR with DenseNet-101.** Table 3 shows the accuracy of ActSub for CIFAR10 [29] and CIFAR100 [29] as ID setup. To vary the backbone even more, we use the popular DenseNet-101 [23] model, trained as in [1, 50]. For OOD detection, we report the average accuracy over four datasets: SVHN [40], iSUN [59], Places365 [65], and Textures [7]. Similarly to previous experiments, our method ActSub outperforms the considered baselines. We increase the AUC over SCALE by 0.71 %/0.91 % points and reduce the FPR by 2.47 %/4.60 % points for CIFAR10 and CIFAR100, respectively. These observations emphasize the generalization capabilities of our approach to different datasets.

**OpenOOD.** We extend our experiments to the more challenging OpenOOD [63] benchmark. OpenOOD proposes a categorization of OOD datasets based on how close they are to the domain as Near- and Far-OOD. For ImageNet-1k as ID, OpenOOD considers NINCO [3] and SSB-Hard [54] as Near-OOD; iNaturalist [22], Textures [7], and Open-Image-O [55] as Far-OOD. We again use the torchvision ResNet-50 checkpoint trained on ImageNet-1k. We further evaluate ActSub in combination with ReAct, ASH-S, and SCALE, and present the results in Table 4. For Far-OOD, we observe a consistent improvement with our method ActSub. For ReAct, ASH-S, and SCALE, we increase the AUC by 2.89 %/0.44 %/0.43 % points and reduce the FPR by 10.74 %/2.64 %/2.24 % points, respectively. Furthermore, ActSub significantly outperforms SCALE and ASH-S in the more challenging Near-OOD case, increasing the AUC by 2.88 %/4.43 % points and reducing the FPR by 7.16 %/9.57 % points, respectively. When combined with SCALE, ActSub reaches SotA results for OpenOOD.

| | Basis | Near-OOD (Avg.) | | Far-OOD (Avg.) | |
|---|---|---|---|---|---|
| | | AUC (↑) | FPR (↓) | AUC (↑) | FPR (↓) |
| $\vec{S}$ | PCA | 64.10 | **75.84** | 71.55 | 63.00 |
| | SI-PCA | 65.68 | 78.42 | 90.62 | 34.94 |
| | Null Space | 66.43 | 77.64 | 92.20 | 31.11 |
| | SVD | **67.51** | 76.27 | **92.74** | **29.51** |
| $\overleftarrow{S}$ | PCA | 80.97 | 60.30 | 96.10 | 18.75 |
| | SI-PCA | 81.37 | 60.06 | **96.36** | **17.54** |
| | Null Space | 81.68 | 59.71 | 96.35 | 17.80 |
| | SVD | **84.90** | **56.39** | 93.58 | 29.31 |
| $\overleftrightarrow{S}$ | PCA | 80.40 | 58.44 | 96.17 | 17.30 |
| | SI-PCA | 80.69 | 59.99 | 96.73 | 15.43 |
| | Null Space | 80.04 | 60.94 | 96.78 | 15.10 |
| | SVD | **84.24** | **52.60** | **96.96** | **14.29** |

Table 5. *Different bases for $\vec{S}$, $\overleftarrow{S}$, and $\overleftrightarrow{S}$.* Our proposed SVD basis outperforms other bases for $\vec{S}$ on the Far-OOD setting and for $\overleftarrow{S}$ in the Near-OOD setting *(gray)*. However, $\overleftarrow{S}$ with SVD suffers in the Far-OOD setting *(red)*. When combining both score functions, SVD leads in both settings.

## 4.3. Ablation studies

Motivated by the behavior of our method on Textures – a Far-OOD dataset compared to iNaturalist, SUN, and Places – we conduct ablations with the experimental setting of OpenOOD to gain a better understanding of different methods under Near- and Far-OOD settings separately.

**Different basis-selection strategies.** In Tab. 5, we investigate how different basis-selection strategies for decisive and insignificant subspaces affect the individual and combined components of our method. We compare our method *(SVD)* against three different basis-selection strategies: *(PCA)* We take principal directions from eigenvectors of the covariance matrix of activations as the decisive component; residual directions define the insignificant component. *(SI-PCA)* We enforce the softmax-invariant direction to be captured in the residual subspace of PCA by setting the activation variance along the softmax-invariant direction to zero. *(Null Space)* We define the insignificant component as the null space of the weight matrix, leaving the row space for the decisive component. For PCA and SI-PCA, we take $D = 512$ directions, which corresponds to 95 % of the variance.

For the cosine similarity on the insignificant component, *i.e.*, $\vec{S}$, we observe that PCA particularly suffers in Far-OOD. Capturing the proposed softmax-invariant direction in the PCA residuals strongly improves the accuracy, highlighting the importance of distinguishing between activation variance and activation directions that only marginally affect the classifier output. Null space improves over SI-PCA, as it targets the directions orthogonal to the model's decision. Our relaxation of the null space further improves the accuracy. For activation shaping on the decisive component, *i.e.*, $\overleftarrow{S}$, we observe similar trends as before, but now on the Near-OOD setup, with our SVD decomposition achieving the highest accuracy. However, the Far-OOD accuracy underperforms with SVD. Combining information from both

| | Component for $\vec{S}$ | Near-OOD (Avg.) | | Far-OOD (Avg.) | |
|---|---|---|---|---|---|
| | | AUC (↑) | FPR (↓) | AUC (↑) | FPR (↓) |
| $\vec{S}$ | $\overleftarrow{\mathbf{a}}$ | 70.18 | 73.74 | 76.34 | 60.50 |
| | $\mathbf{a}$ | **71.75** | **70.11** | 87.77 | 40.80 |
| | $\overrightarrow{\mathbf{a}}$ | 67.51 | 76.27 | **92.74** | **29.30** |
| $\overleftrightarrow{S}$ | $\overleftarrow{\mathbf{a}}$ | 81.89 | 56.77 | 92.47 | 28.41 |
| | $\mathbf{a}$ | 84.11 | 52.86 | 96.00 | 17.98 |
| | $\overrightarrow{\mathbf{a}}$ | **84.24** | **52.60** | **96.96** | **14.29** |

Table 6. *Different components for $\vec{S}$.* We show how different activation components ($\overleftarrow{\mathbf{a}}$, $\mathbf{a}$, $\overrightarrow{\mathbf{a}}$) affect our cosine similarity-based score function $\vec{S}$, when considered in isolation and in combination with the activation shaping-based $\overleftarrow{S}$ (*i.e.*, $\overleftrightarrow{S}$).

subspaces of our method ($\overleftrightarrow{S}$) again complements each other, achieving the highest accuracy in both settings.

**Different components for $\vec{S}$.** Tab. 6 shows the accuracy when different activation components – *i.e.*, the entire activation $\mathbf{a}$, the decisive component $\overleftarrow{\mathbf{a}}$, and the insignificant component $\overrightarrow{\mathbf{a}}$ – are utilized for our cosine similarity-based score $\vec{S}$. For Far-OOD, the accuracy of $\vec{S}$ from the decisive component $\overleftarrow{\mathbf{a}}$ is significantly behind that of the activation $\mathbf{a}$. This observation follows our motivation on the model enforcing alignment on the directions with high significance. Supporting our approach, using the insignificant component $\overrightarrow{\mathbf{a}}$ yields the best results. The same conclusions are drawn when combining both of our score functions ($\overleftrightarrow{S}$).

**Limitations.** *(i)* Similar to other distance-based approaches, our cosine similarity-based score $\vec{S}$ relies on the training data, which limits the applicability when not available. *(ii)* While our method is not overly sensitive to the hyperparameter $\lambda$ (see supplement), the optimal $\lambda$ can still vary for different backbones. *(iii)* Our method relies on a linear classification head. *(iv)* Our evaluation is limited to the image classification domain, and the extension to other tasks, such as semantic segmentation, could provide valuable insights about our method's generalization.

## 5. Conclusion

OOD detection methods rely on the information contained in the activation space. In this study, we define two orthogonal subspaces within the activation space: *decisive* and *insignificant*. The decisive component of activations determines the model's decision. On the other hand, the insignificant component is decoupled from the classification output and yields powerful features, akin to a random neural network, that are discriminative for OOD detection. Further, we observe that selectively applying activation shaping to the decisive component increases the separability of ID and OOD distributions, particularly for OOD inputs close to the ID domain. By combining discriminative information from both subspaces, our method ActSub outperforms relevant baselines across a variety of different benchmarks.

# References

[1] Yong Hyun Ahn, Gyeong-Moon Park, and Seong Tae Kim. LINe: Out-of-distribution detection by leveraging important neurons. In *CVPR*, pages 19852–19862, 2023. 2, 6, 7, iv

[2] Sima Behpour, Thang Long Doan, Xin Li, Wenbin He, Liang Gou, and Liu Ren. GradOrth: A simple yet efficient out-of-distribution detection with orthogonal projection of gradients. In *NeurIPS*, 2023. 2

[3] Julian Bitterwolf, Maximilian Müller, and Matthias Hein. In or out? Fixing ImageNet out-of-distribution detection evaluation. In *ICML*, pages 2471–2506, 2023. 7, i, ii

[4] Yun-Hao Cao and Jianxin Wu. A random CNN sees objects: One inductive bias of CNN and its applications. In *AAAI*, pages 194–202, 2022. 1

[5] Robin Chan, Matthias Rottmann, and Hanno Gottschalk. Entropy maximization and meta classification for out-of-distribution detection in semantic segmentation. In *ICCV*, pages 5108–5117, 2021. 2

[6] Jinggang Chen, Junjie Li, Xiaoyang Qu, Jianzong Wang, Jiguang Wan, and Jing Xiao. GAIA: Delving into gradient-based attribution abnormality for out-of-distribution detection. In *NeurIPS*, 2023. 2

[7] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *CVPR*, pages 3606–3613, 2014. 5, 7, i, iv

[8] Matthew Cook, Alina Zare, and Paul D. Gader. Outlier detection through null space analysis of neural networks. *arXiv:2007.01263 [cs.LG]*, 2020. 1, 2, 3

[9] 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, 2009. 3, 5, i, ii, iii

[10] Sjoerd Dirksen, Martin Genzel, Laurent Jacques, and Alexander Stollenwerk. The separation capacity of random neural networks. *Int. J. Comput. Vision*, 23(309):1–47, 2022. 1, 3

[11] Andrija Djurisic, Nebojsa Bozanic, Arjun Ashok, and Rosanne Liu. Extremely simple activation shaping for out-of-distribution detection. In *ICLR*, 2023. 1, 2, 3, 6, 7, i, iv

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

[13] Kun Fang, Qinghua Tao, Kexin Lv, Mingzhen He, Xiaolin Huang, and Jie Yang. Kernel PCA for out-of-distribution detection. In *NeurIPS*, 2024. 2, 5, 6, 7

[14] Maximilian Granz, Manuel Heurich, and Tim Landgraf. WeiPer: OOD detection using weight perturbations of class projections. In *NeurIPS*, 2024. 2, 7, iii, iv

[15] Xiaoyuan Guan, Zhouwu Liu, Wei-Shi Zheng, Yuren Zhou, and Ruixuan Wang. Revisit PCA-based technique for out-of-distribution detection. In *ICCV*, pages 19374–19382, 2023. 1, 2, 5, 6, 7

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 3, 5, i, ii

[17] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *ICLR*, 2017. 1, 2, 5, 6, 7, iv

[18] Dan Hendrycks, Mantas Mazeika, and Thomas G. Dietterich. Deep anomaly detection with outlier exposure. In *ICLR*, 2019. 2

[19] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *ICCV*, pages 8320–8329, 2021. ii

[20] Dan Hendrycks, Steven Basart, Mantas Mazeika, Andy Zou, Joseph Kwon, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. Scaling out-of-distribution detection for real-world settings. In *ICML*, pages 8759–8773, 2022. 1, 2

[21] Claus Hofmann, Simon Schmid, Bernhard Lehner, Daniel Klotz, and Sepp Hochreiter. Energy-based Hopfield boosting for out-of-distribution detection. In *NeurIPS*, 2024. 2

[22] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alexander Shepard, Hartwig Adam, Pietro Perona, and Serge J. Belongie. The iNaturalist species classification and detection dataset. In *CVPR*, pages 8769–8778, 2018. 5, 7

[23] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *CVPR*, pages 770–778, 2017. 7

[24] Rui Huang and Yixuan Li. MOS: Towards scaling out-of-distribution detection for large semantic space. In *CVPR*, pages 8710–8719, 2021. 2, 5

[25] Rui Huang, Andrew Geng, and Yixuan Li. On the importance of gradients for detecting distributional shifts in the wild. In *NeurIPS*, pages 677–689, 2021. 2

[26] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015. 2

[27] Ziwei Ji and Matus Telgarsky. Directional convergence and alignment in deep learning. In *NeurIPS*, 2020. 3

[28] Donald E. Knuth. Two notes on notation. *The American Mathematical Monthly*, 99(5):403–422, 1992. 2

[29] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. 7, ii, iii, iv

[30] Yann LeCun, Corinna Cortes, and Chris Burges. MNIST handwritten digit database, 2010. ii

[31] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution

samples and adversarial attacks. In *NeurIPS*, pages 7167–7177, 2018. 1, 2, 6

[32] Yixia Li, Boya Xiong, Guanhua Chen, and Yun Chen. Se-TAR: Out-of-distribution detection with selective low-rank approximation. In *NeurIPS*, 2024. 2

[33] Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *ICLR*, 2018. 1, 6, 7

[34] Ziqian Lin, Sreya Dutta Roy, and Yixuan Li. MOOD: Multi-level out-of-distribution detection. In *CVPR*, pages 15313–15323, 2021. 2

[35] Litian Liu and Yao Qin. Fast decision boundary based out-of-distribution detector. In *ICML*, 2024. 1, 2, 6

[36] Weitang Liu, Xiaoyun Wang, John D. Owens, and Yixuan Li. Energy-based out-of-distribution detection. In *NeurIPS*, 2020. 1, 2, 5, 6, 7, iv

[37] Xixi Liu, Yaroslava Lochman, and Christopher Zach. GEN: Pushing the limits of softmax-based out-of-distribution detection. In *CVPR*, pages 23946–23955, 2023. 1, 2, 7, i, ii, iii, iv

[38] Haodong Lu, Dong Gong, Shuo Wang, Jason Xue, Lina Yao, and Kristen Moore. Learning with mixture of prototypes for out-of-distribution detection. In *ICLR*, 2024. 2

[39] Yifei Ming, Yiyou Sun, Ousmane Dia, and Yixuan Li. How to exploit hyperspherical embeddings for out-of-distribution detection? In *ICLR*, 2023. 2

[40] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS*2011*. 7, iv

[41] Jaewoo Park, Yoon Gyo Jung, and Andrew Beng Jin Teoh. Nearest neighbor guidance for out-of-distribution detection. In *ICCV*, pages 1686–1695, 2023. 2, 4, 6, 7

[42] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 5

[43] Jie Ren, Stanislav Fort, Jeremiah Z. Liu, Abhijit Guha Roy, Shreyas Padhy, and Balaji Lakshminarayanan. A simple fix to Mahalanobis distance for improving near-OOD detection. *arXiv:2106.09022 [cs.LG]*, 2021. iii

[44] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *Int. J. Comput. Vision*, 115(3):211–252, 2015. 3, 5, i, ii, iii

[45] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted residuals and linear bottlenecks. In *CVPR*, pages 4510–4520, 2018. 7, ii

[46] Vikash Sehwag, Mung Chiang, and Prateek Mittal. SSD: A unified framework for self-supervised outlier detection. In *ICLR*, 2021. 2

[47] L. S. Shapley. *17. A Value for n-Person Games*, pages 307–318. Princeton University Press, 1953. 2

[48] Yue Song, Nicu Sebe, and Wei Wang. RankFeat: Rank-1 feature removal for out-of-distribution detection. In *NeurIPS*, 2022. 2, 7, iv

[49] G. Strang. *Linear Algebra and Its Applications*. Thomson, Brooks/Cole, 2006. 3

[50] Yiyou Sun and Yixuan Li. DICE: Leveraging sparsification for out-of-distribution detection. In *ECCV*, pages 691–708, 2022. 2, 6, 7, iv

[51] Yiyou Sun, Chuan Guo, and Yixuan Li. ReAct: Out-of-distribution detection with rectified activations. In *NeurIPS*, 2021. 1, 2, 3, 5, 6, 7, i, iv

[52] Yiyou Sun, Yifei Ming, Xiaojin Zhu, and Yixuan Li. Out-of-distribution detection with deep nearest neighbors. In *ICML*, pages 20827–20840, 2022. 1, 2, 6, 7, iii, iv

[53] Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. Deep image prior. In *CVPR*, pages 9446–9454, 2018. 1

[54] Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Open-set recognition: A good closed-set classifier is all you need. In *ICLR*, 2022. 7

[55] Haoqi Wang, Zhizhong Li, Litong Feng, and Wayne Zhang. ViM: Out-of-distribution with virtual-logit matching. In *CVPR*, pages 4911–4920, 2022. 2, 3, 5, 6, 7, iii

[56] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In *CVPR*, pages 3485–3492, 2010. 5

[57] Kai Xu, Rongyu Chen, Gianni Franchi, and Angela Yao. Scaling for training time and post-hoc out-of-distribution detection enhancement. In *ICLR*, 2024. 1, 2, 3, 6, 7, i, ii, iii, iv

[58] Mingyu Xu, Zheng Lian, Bin Liu, and Jianhua Tao. VRA: Variational rectified activation for out-of-distribution detection. In *NeurIPS*, 2023. 1, 2, 3, 6

[59] Pingmei Xu, Krista A. Ehinger, Yinda Zhang, Adam Finkelstein, Sanjeev R. Kulkarni, and Jianxiong Xiao. TurkerGaze: Crowdsourcing saliency with webcam based eye tracking. *arXiv:1504.06755 [cs.CV]*, 2015. 7, iv

[60] Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *Int. J. Comput. Vision*, 132(12):5635–5662, 2024. 1

[61] Yue Yuan, Rundong He, Yicong Dong, Zhongyi Han, and Yilong Yin. Discriminability-driven channel selection for out-of-distribution detection. In *CVPR*, pages 26171–26180, 2024. 2, 6, 7, iv

[62] Jinsong Zhang, Qiang Fu, Xu Chen, Lun Du, Zelin Li, Gang Wang, Xiaoguang Liu, Shi Han, and Dongmei Zhang. Out-of-distribution detection based on in-distribution data patterns memorization with modern Hopfield energy. In *ICLR*, 2023. 2, 6, 7, iii, iv

[63] Jingyang Zhang, Jingkang Yang, Pengyun Wang, Haoqi Wang, Yueqian Lin, Haoran Zhang, Yiyou Sun, Xuefeng Du, Kaiyang Zhou, Wayne Zhang, Yixuan Li, Ziwei Liu, Yiran Chen, and Hai Li. OpenOOD v1.5: Enhanced benchmark for out-of-distribution detection. *arXiv:2306.09301 [cs.LG]*, 2023. 7, i, ii, iii

[64] Zihan Zhang and Xiang Xiang. Decoupling MaxLogit for out-of-distribution detection. In *CVPR*, pages 3388–3397, 2023. 1, 2, 6

[65] Bolei Zhou, Àgata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE T. Pattern Anal. Mach. Intell.*, 40(6):1452–1464, 2018. 5, 7, iv

[66] Yao Zhu, Yuefeng Chen, Chuanlong Xie, Xiaodan Li, Rong Zhang, Hui Xue, Xiang Tian, Bolun Zheng, and Yaowu Chen. Boosting out-of-distribution detection with typical features. In *NeurIPS*, 2022. 2, 6, 7