# LINe: Out-of-Distribution Detection by Leveraging Important Neurons

Yong Hyun Ahn[1], Gyeong-Moon Park[2,*], Seong Tae Kim[2,*]
[1]Department of Artificial Intelligence, Kyung Hee University
[2]Department of Computer Science and Engineering, Kyung Hee University

## Abstract

*It is important to quantify the uncertainty of input samples, especially in mission-critical domains such as autonomous driving and healthcare, where failure predictions on out-of-distribution (OOD) data are likely to cause big problems. OOD detection problem fundamentally begins in that the model cannot express what it is not aware of. Post-hoc OOD detection approaches are widely explored because they do not require an additional re-training process which might degrade the model's performance and increase the training cost. In this study, from the perspective of neurons in the deep layer of the model representing high-level features, we introduce a new aspect for analyzing the difference in model outputs between in-distribution data and OOD data. We propose a novel method, Leveraging Important Neurons (LINe), for post-hoc Out of distribution detection. Shapley value-based pruning reduces the effects of noisy outputs by selecting only high-contribution neurons for predicting specific classes of input data and masking the rest. Activation clipping fixes all values above a certain threshold into the same value, allowing LINe to treat all the class-specific features equally and just consider the difference between the number of activated feature differences between in-distribution and OOD data. Comprehensive experiments verify the effectiveness of the proposed method by outperforming state-of-the-art post-hoc OOD detection methods on CIFAR-10, CIFAR-100, and ImageNet datasets. Code is available on https://github.com/LINe-OOD*

## 1. Introduction

Recently, deep learning has made tremendous advances in various fields. This advancement has captivated numerous researchers, leading to many attempts to apply deep learning techniques to real-world applications. However, applying these state-of-the-art techniques to real-world applications is often limited for several reasons. One primary obstacle is the presence of unseen classes of samples during training. These samples, referred to as out-of-distribution (OOD) data, can compromise a model's stability and, in some cases, severely impair its performance. The inherent characteristics of OOD samples can lead to potentially severe consequences in mission-critical domains, such as autonomous driving and medical applications. So, effectively handling these OOD samples is vital to avoid problems, such as car crashes and misdiagnosis.

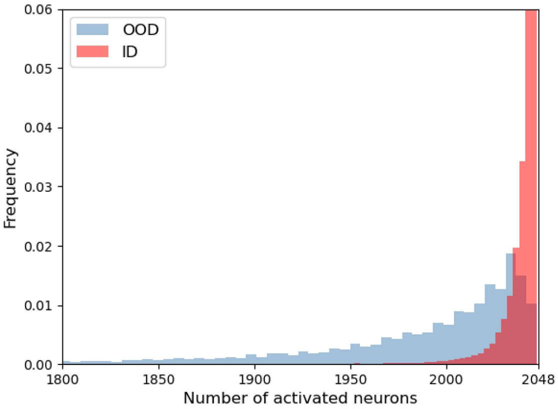

Figure 1. **Illustration of the number of activated neuron in the penultimate layer.** The distributions of the number of activated neurons from ID (ImageNet [19]) and two OOD datasets (textures [10] and iNaturalist [51]) are presented. We define an activated neuron as one fired with an activation value greater than zero. Most of the neurons are activated when the model receives ID samples but fewer neurons are activated in OOD samples.

For OOD detection, numerous techniques have been explored to analyze the distinction between in-distribution (ID) data and OOD data. There are several approaches to OOD detection, including confidence-based methods [7, 11, 16, 19, 29], density-based methods [1, 20, 22–24, 27, 38, 40, 42, 50, 62, 63], and distance-based methods [9, 18, 28, 35, 41, 46, 48, 49, 58]. The post-hoc method is one approach in OOD detection that offers significant advantages in real-world applications as it eliminates the need for a re-training process, which could potentially degrade the model's per-

[*]Corresponding authors: Gyeong-Moon Park (gmpark@khu.ac.kr) and Seong Tae Kim (st.kim@khu.ac.kr).

formance and increase training costs [56].

Post-hoc OOD detection methods [32, 44, 45] employ outputs such as model logits or layer activations, which are typically used for prediction, to calculate OOD scores. These scores allow for the differentiation of ID and OOD data based on the disparities in their respective scores [56]. Enhancing the overall OOD score distribution difference between ID and OOD data is a critical aspect in improving the performance of post-hoc OOD detection methods. Recent studies [44,45] have found that augmenting the overall difference is contingent upon the capacity to mitigate noisy signals. For example, ReAct [44] demonstrates that OOD data exhibit considerable values in the penultimate layer activation. By truncating these noisy activations, ReAct effectively improves OOD detection performance. Similarly, DICE [45] uncovers the presence of noisy signals that increase the variance of the OOD score distribution. By selectively employing the most salient weights, DICE enhances OOD detection performance while reducing the impact of noisy signals. Minimizing the influence of noisy model outputs is crucial for advancing post-hoc OOD detection performance. In this study, we reveal an additional factor that is also important in the context of OOD detection.

Figure 1 displays the histogram of the number of activated features in the penultimate layer. We define neurons with activation values greater than zero as activated neurons. As the figure demonstrates, a majority of neurons are activated when the model encounters ID samples. However, for OOD samples, fewer neurons are activated. To understand the primary cause of the observation in Figure 1, we investigate the model from the perspective of neuron-concept association [4,5,21,52]. According to [4], neurons are trained to detect disentangled high-level features in the deep layers of convolutional neural networks (CNNs), and they can even learn new unlabeled abstract concepts from the data [52]. The number of neurons representing these high-level features increases in deeper layers and becomes the most significant amount in the penultimate layer [5]. Since the activation in the penultimate layer represents the presence of these high-level features [4], different input images activate high-level features in varying ways, such as magnitude and patterns. These distinct patterns of activation in the penultimate layer are ultimately used to predict the input image's class. As each class possesses unique characteristics related to high-level concepts, the associated high-level features differ for each class. Consequently, each associated high-level feature can be categorized into one class-specific feature group. The disparity in associated high-level features in neurons results in a difference in penultimate layer activation between ID and OOD samples, which are predicted as the same class. Thus, taking into account the number of activated essential neurons can serve as a useful indicator for distinguishing ID and OOD samples.

In this paper, we present a novel post-hoc OOD detection method called ***L****everaging* ***I****mportant* ***Ne****urons* (**LINe**). LINe harnesses two crucial aspects: considering the number of activated important neurons and minimizing noisy activations to enhance OOD detection performance. To achieve this, we introduce two powerful techniques within LINe: 1) Shapley-based pruning and 2) activation clipping (AC).

Shapley value-based pruning is a method that mitigates the influence of noisy outputs by selecting activations essential for inferring input data classes. While several methods exist for identifying important activations, Sun et al. [45] use activation magnitude to determine their importance. However, this approach alone is insufficient for quantifying a neuron's importance. Hence, we employ the Shapley value [43] to more accurately measure each neuron's contribution. Applying the Shapley value concept [43] to neural networks allows us to quantify each neuron's contribution to identifying a specific class. Moreover, neurons with high Shapley values are associated with critical input image features [21]. By leveraging the Shapley value [43], we can identify neurons that represent important high-level features for each class, which are essential for reducing noisy output.

Activation clipping, a concept introduced in ReAct [44], is another powerful technique. Interpreting activation clipping in terms of class-specific features provides a new understanding of its role in considering the number of activated important neurons for OOD detection. Activation clipping adjusts values exceeding a certain threshold to the threshold value. This modification enables us to account for the differences in the number of activated features between in-distribution and OOD data by treating numerous class-specific features equally. By considering the variation in the number of activated class-specific features, we can effectively augment the overall OOD score difference between ID and OOD data, leading to enhanced performance.

Our key contributions are summarized as follows:

- We propose a simple yet effective post-hoc OOD detection method, named LINe, which uses the Shapley value to rank the contribution of neurons and gives a new inspiration for leveraging selected important class-specific neurons.
- We unveil the important factor for improving OOD scoring and show a new way of understanding the role of activation clipping for OOD detection. By applying activation clipping, we can fully consider the number of activated class-specific features and achieve higher OOD detection performance.
- Comprehensive experiments have been conducted to verify the effectiveness of the proposed method on the CIFAR-10, CIFAR-100, and ImageNet-1K. Compared to the competitive post-hoc method DICE [45], LINe reduces the FPR95 by up to 14.05%.

# 2. Background and Related Work

## 2.1. Neuron-Concept Association

Neuron-concept association methods are a field of study that tries to interpret the internal computation of CNN to a human-understandable concept [3,8,14,37]. Several studies have shown that neurons of shallower layers tend to learn more simple and low-level concepts, such as curves and edges, while deeper layers learn more abstract and high-level concepts, such as arm and face [52,59,60]. The methods for quantifying the concept's contribution are also introduced in [12, 14, 33]. Network Dissection [4, 5, 59] assigns each neuron to a concept to quantify its role. Bau et al. [6] investigate the effect of concept-specific neurons by observing the change of concept-related contents in generative models. Recently, Wang et al. [52] show that models can learn abstract concepts like mammal and carnivore, which is not in the label set of training data.

## 2.2. Shapley Value

Shapely value [26, 43, 47] is a concept from Game Theory, which evaluates each property's individual and collaborative effects. Studies have been conducted using Shapley value in CNNs to measure each neuron's contribution and interpret models' behaviors [2, 13, 21, 34, 47]. Neuron Shapley [13] sorts the Shapley values to identify the most influential neurons from all hidden layers as image categories. Khazar et al. [21] show neurons that have high Shapley values have high correlations with important features of the input image. Previous studies have inspired us to select important class-specific neurons through the contribution scores calculated from the Shapley value. LINe can effectively eliminate the negative effects of noisy signals by leveraging the selected class-specific neurons.

## 2.3. Out-of-Distribution Detection

OOD detection aims to find inputs with different characteristics from the training data [56]. A lot of research efforts have been devoted to developing an effective method to distinguish OOD inputs from ID inputs. Confidence-based methods perform OOD detection by quantifying OOD scores based on different scoring functions [7,11,16,19,29]. Hendrycks et al. [16] used a maximum softmax probability (MSP) of the model as a baseline confidence-base OOD scoring function. ODIN [30] utilizes perturbation of inputs and temperature scaling on the softmax layer to increase the difference between ID and OOD. To enhance the effectiveness of confidence-based scores, recently, Liu et al. [32] introduced an energy-based score with the theoretical interpretation from a likelihood perspective, which is further adopted in [31, 36, 53] to distinguish ID and OOD samples. Distance-based approaches measure the distance between input sample and typical ID samples or centroids of them [9, 18, 28, 35, 41, 46, 48, 49, 58]. These approaches are based on simple evidence that OOD samples should have more distance than IDs. Similarly, density-based methods identify OOD samples based on the distribution of the training samples and use density (or likelihood) [1, 20, 22–24, 27, 38, 40, 42, 50, 62, 63].

But none of the aforementioned methods consider the number of activated features, which can be a good indicator to distinguish ID and OOD samples. The most similar study to our study is DICE [45]. DICE leverages sparsification to reduce the effect of noisy signals by selectively using salient weights from activation [45]. In this study, we effectively eliminate the outcomes of noisy signals by accurately selecting neurons leveraging the contribution of neurons calculated from Shapley value [21, 43]. In addition, our method performs OOD detection more effectively by considering the number of activated features. The neurons are known to be associated with the concepts, and the activation patterns of neurons in deep layers are different in ID and OOD. This gives a theoretical background for considering the number of activation in OOD detection. To the best of our knowledge, our work is the first study to leverage the neuron contribution based on Shapley value and consider the number of activated features for calculating OOD score.

# 3. Method

## 3.1. Method Overview

Our method mainly consists of two parts: *Activation Clipping* and *Shapley-based pruning*. Activation clipping is a method of clipping each neuron activation to a specific value when the activation value exceeds a particular threshold ($\delta$). Through activation clipping, our method can consider the number of high-level features during calculating OOD scores, which are analyzed in terms of neuron-concept association. Next, Shapley-based pruning eliminates the negative effect of noisy signals by measuring neural contributions of neural networks using important neurons only. We can accurately measure the contribution of each neuron using Shapley value, which is a mathematically grounded method. Also, by applying Shapley value, we can obtain supporting evidence that neurons with large contributions represent critical features for recognizing input image [21]. This allows us to assemble all contributions for each class and select important class-specific neurons representing class-specific features. More details of activation clipping and Shapley-based pruning will be described in Subsection 3.2 and Subsection 3.3. In Subsection 3.4, we will explain the overall method.

## 3.2. Activation Clipping

For a pre-trained deep neural network $f_\theta(x)$ parameterized by $\theta$, $f_\theta(x)$ encodes an input $x \in \mathbb{R}^d$, where $d$ indicates

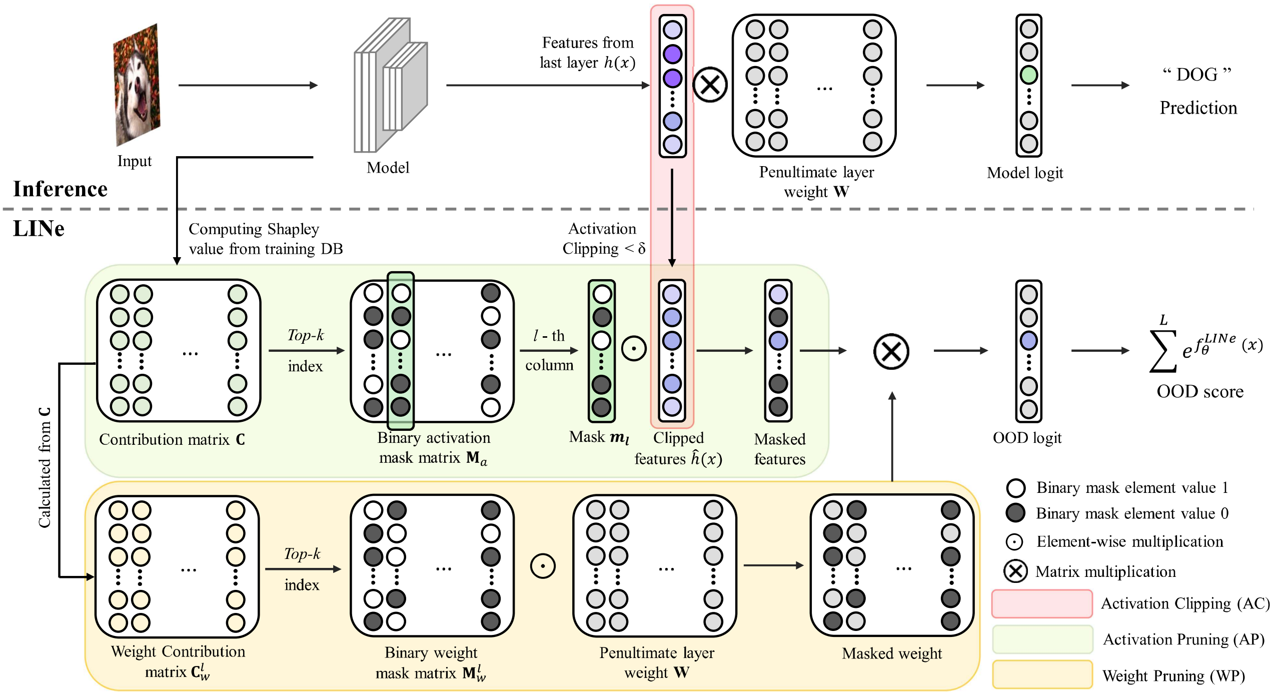
Input
Model
Features from last layer $h(x)$
Penultimate layer weight **W**
Model logit
" DOG "
Prediction
Inference
LINe
Computing Shapley value from training DB
Activation Clipping < δ
Top-k
index
l - th
column
Contribution matrix **C**
Binary activation mask matrix $\mathbf{M}_a$
Mask $m_l$
Clipped features $\hat{h}(x)$
Masked features
OOD logit
$\sum^{L} e^{f_\theta^{LINe}(x)}$
OOD score
Calculated from C
Weight Contribution matrix $\mathbf{C}_w^l$
Binary weight mask matrix $\mathbf{M}_w^l$
Penultimate layer weight **W**
Masked weight
○ Binary mask element value 1
● Binary mask element value 0
⊙ Element-wise multiplication
⊗ Matrix multiplication
🟥 Activation Clipping (AC)
🟩 Activation Pruning (AP)
🟨 Weight Pruning (WP)

Figure 2. **Overall procedure of LINe for calculating OOD score.** LINe is composed of mainly three parts. 1) Activation Clipping (AC), 2) Activation Pruning (AP), 3) Weight Pruning (WP). AC limits the magnitude of the activation, which allows LINe to consider the number of activated features. AP uses contribution matrix **C**, which is precomputed from Shapley value [43]. AP leverages **C** to select important neurons and mask others. WP uses an adjusted contribution matrix which is obtained from the contribution matrix **C**. Leveraging AP and WP, LINe can reduce noisy signals effectively.

dimension of input $x$, and predicts a class distribution for $L$ different classes, i.e., $f_\theta(x) \in \mathbb{R}^L$. Feature vector from the penultimate layer of the network denotes $h(x) \in \mathbb{R}^q$, where $q$ stands for the dimension of penultimate layer output $h(x)$. Weight matrix $\mathbf{W} \in \mathbb{R}^{q \times L}$ weights the importance of each feature in $h(x)$ and transfers to output $f_\theta(x)$ as follows:

$$f_\theta(x) = \mathbf{W}^T h(x) + \mathbf{b}. \tag{1}$$

Activation clipping is applied to the feature vector in the penultimate layer. For each neuron that is activated above a certain threshold, AC limits the magnitude of the activation. As we discussed in Figure 1, the number of activated features in ID and OOD samples are different. By limiting the magnitude of the activation to the same value $\delta$, we can treat every activated high-level feature equally, which allows us to consider the number of activated features in the OOD score. This increases the OOD score difference between ID and OOD samples, thereby improving performance.

For each activation $a_i$ in $h(x)$, penultimate layer feature $h(x)$ can be denoted as $h(x) = [a_1, a_2, \cdots, a_q]$. With a clipping threshold $\delta$, the clipped activation $\hat{a}_i$ can be described as $\hat{a}_i = min(a_i, \delta)$. The clipped feature vector $\hat{h}(x)$ is then,

$$\hat{h}(x) = [\hat{a}_1, \hat{a}_2, \cdots, \hat{a}_q]. \tag{2}$$

The model output after AC can be given as:

$$f_\theta^{AC}(x) = \mathbf{W}^T \hat{h}(x) + \mathbf{b}. \tag{3}$$

### 3.3. Shapley-based Pruning

Shapley-based pruning selectively uses a subset of important neuron activation and weights. To select these subsets, we calculate Shapley value [43] defined by an average of the effect of removing a single unit (e.g., marginal contribution) to all possible combinations of units. However, computing all the combinations of units is computationally expensive and practically infeasible for recent large neural networks. Therefore, we use the Taylor approximation to compute the Shapley value, which is introduced in [21]. For input $x^l \in D$, where $x^l$ denotes the sample of class $l$ from dataset $D$, a contribution(i.e., Shapley value) of $i$-th neuron $a_i$ in class $l$, $s_i^l$ is calculated as

$$s_i^l = \left| f_\theta(x^l) - f_\theta(x^l; a_i \leftarrow 0) \right| = \left| a_i \nabla_{a_i} f_\theta(x^l) \right|. \tag{4}$$

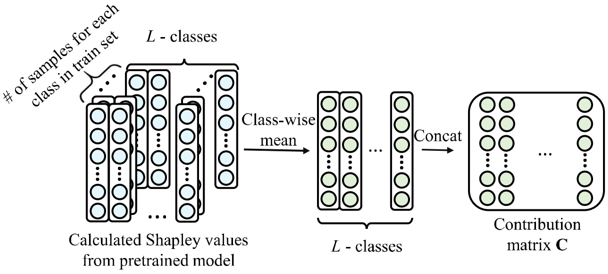
# of samples for each class in train set
L - classes
Class-wise mean
Concat
Calculated Shapley values from pretrained model
L - classes
Contribution matrix C

Figure 3. **Calculation of contribution matrix C.**

### 3.3.1 Activation Pruning

Activation pruning (AP) selectively uses the subset of important neuron activation. Through AP, We can effectively reduce the impact of noisy activation. From the contribution of each neuron $s^l_i$, which is using obtained from all training data, contribution matrix $\mathbf{C} \in \mathbb{R}^{q \times L}$ is defined as the class-specific average of all contribution $s^l_i$. An ($i$,$l$)-th entry of contribution matrix $c_{il} \in \mathbf{C}$ is defined as:

$$\mathbf{C}_{il} = \frac{1}{n}\sum^{n} s^l_i, \tag{5}$$

where $n$ denotes the number of training images in class $l$. We select *top-k* neurons for each class based on the k-largest elements from each column in $\mathbf{C}$ and define an activation mask matrix $\mathbf{M}_a \in \mathbb{R}^{q \times L}$, where we set 1 for the $k$-largest elements from every column in $\mathbf{C}$, otherwise 0. The model output after AP with the predicted class $l$ is given as:

$$f^{AP}_\theta(x) = \mathbf{W}^T(\mathbf{m}_l \odot h(x)) + \mathbf{b}, \tag{6}$$

where $\mathbf{m}_l \in \mathbb{R}^q$ indicates $l$-th column of mask matrix $\mathbf{M}_a$ and $\odot$ denotes the element-wise multiplication.

### 3.3.2 Weight Pruning

Weight pruning (WP) selectively uses the subset of important penultimate layer weights. Through WP, We can effectively reduce the impact of noisy signals due to the overparameterized model. The contribution of neurons is further used to refine the weight matrix as WP. For this purpose, we define a weight contribution matrix for the class $l$ as $\mathbf{C}^l_w = \mathbf{c}_l \odot \mathbf{W} \in \mathbb{R}^{q \times L}$ where $\mathbf{c}_l$ indicates the $l$-th column of contribution matrix $\mathbf{C}$. We select *top-k* weights for each class based on the $k$-largest elements in $\mathbf{C}^l_w$ and define a mask matrix for class $l$ as $\mathbf{M}^l_w \in \mathbb{R}^{q \times L}$ by setting 1 for the k-largest elements in $\mathbf{C}^l_w$, otherwise 0. The model output after WP with the predicted class $l$ is given as:

$$f^{WP}_\theta(x) = (\mathbf{W} \odot \mathbf{M}^l_w)^T h(x) + \mathbf{b}. \tag{7}$$

From the Shapley-based AP and WP, we can determine the neurons representing important high-level features for each class, which play a crucial role in reducing noisy output.

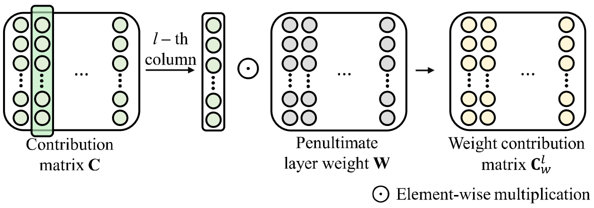
l – th column
⊙
Contribution matrix C
Penultimate layer weight W
Weight contribution matrix $\mathbf{C}^l_w$
⊙ Element-wise multiplication

Figure 4. **Calculation of contribution matrix $\mathbf{C^l_w}$.**

## 3.4. Leveraging Important Neurons (LINe)

As already described in the previous Subsection 3.2 and 3.3, there are two ideas to improve the performance of post-hoc OOD detection. One is considering the number of activated high-level features, and the other is reducing noisy signals from less important neurons. LINe achieves both ways to improve post-hoc OOD detection performance through the procedures described above. By adapting LINe to a model, we can effectively increase the overall difference in OOD scores between ID and OOD data. As a result, the model output under LINe with the predicted class $l$ is described as

$$f^{LINe}_\theta(x) = (\mathbf{W} \odot \mathbf{M}^l_w)^T(\mathbf{m}_l \odot \hat{h}(x)) + \mathbf{b}. \tag{8}$$

Please note that there is no parameter change in the model itself and ID classification accuracy can be preserved.

# 4. Experiments

Comprehensive experiments have been conducted to evaluate our method. In section 4.1, we used the CIFAR [25] benchmark, which is one of the famous benchmarks in OOD studies [32, 44, 45]. In Section 4.2, experiments were conducted based on a large-scale dataset, ImageNet, with various OOD datasets. Section 4.3 analyzes why our method is effective through various ablation experiments.

## 4.1. Evaluation on CIFAR Benchmarks

**Implementation Details.** In this experiment, we used 10,000 test images each from CIFAR-10 [25] and CIFAR-100 [25] as ID data, respectively. The model's performance was evaluated using six commonly used OOD datasets as an OOD benchmark. The list of six OOD datasets is as follows: SVHN [39], Textures [10], iSUN [55], LSUN-Crop [57], LSUN-Resize [57], and Places365 [61]. As the pre-trained models, we used DenseNet [17]. As in [45], the models are trained from CIFAR-10 [25] and CIFAR-100 [25] with 50,000 training images respectively. Following [45], the models were trained during 100 epochs with batch size 64, weight decay 0.0001, momentum 0.9, and

Table 1. **Comparison on CIFAR benchmarks.** Comparison with competitive post-hoc OOD detection methods on CIFAR benchmarks. All values in this table are percentages and averaged over six OOD test datasets. ↓ indicates smaller value means better performance and ↑ indicates vice versa. **Bold** numbers are superior results. The overall detail results for each OOD dataset are provided in supplementary material.

| **Method** | **CIFAR-10** | | **CIFAR-100** | |
|---|---|---|---|---|
| | **FPR95** ↓ | **AUROC** ↑ | **FPR95** ↓ | **AUROC** ↑ |
| MSP [16] | 48.73 | 92.46 | 80.13 | 74.36 |
| ODIN [30] | 24.57 | 93.71 | 58.14 | 84.49 |
| Mahalanobis [28] | 31.42 | 89.15 | 55.37 | 82.73 |
| Energy [32] | 26.55 | 94.57 | 68.45 | 81.19 |
| ReAct [44] | 26.45 | 94.95 | 62.27 | 84.47 |
| DICE [45] | 20.83 | 95.24 | 49.72 | 87.23 |
| DICE + ReAct | 16.48 | 96.64 | 49.57 | 85.08 |
| **LINe (Ours)** | **14.71** | **96.99** | **35.67** | **88.67** |

start learning rate 0.1. The learning rate was decayed by a factor of 10 at epochs 50, 75, and 90. We used the entire training dataset to estimate the contribution matrix $C$.

**Comparison.** For comparison, we adopted recent post-hoc OOD detection methods: MSP [16], ODIN [30], Mahalanobis distance [28], Energy [32], ReAct [44], and DICE [45]. For all methods, the performances were measured by OOD scores, derived from the same DenseNet model.

**Experimental Results.** Table 1 shows the comparisons between LINe and other post-hoc OOD detection methods on CIFAR-10 and CIFAR-100 benchmarks. As shown in the table, our method achieved state-of-the-art performances by outperforming all the other methods on both CIFAR-10 and CIFAR-100 datasets. In CIFAR-100, LINe reduced FPR 95 by 14.05% compared to the competitive method DICE [45]. In CIFAR-100, LINe achieved an FPR 95 of 14.05% which was lower than the FPR 95 of DICE [45] (49.72%) and DICE + ReAct (49.57%). DICE + ReAct is the method that is implemented by applying ReAct [44] on DICE [45]. DICE [45] removed the noisy signals by using the magnitude of activation and weight. LINe not only removed the noisy signals using class-specific neurons but also considered the number of activations of the high-level feature.

## 4.2. Evaluation on ImageNet

**Implementation Details.** In real-world applications, the model encounters high-resolution images with various scenes and features, and evaluation on a large-scale dataset can provide clues about model performance in a real-world application. Therefore, in this experiment, we evaluated LINe on a large-scale ImageNet dataset. Based on [19], a subset of the four datasets where all the overlapping categories with ImageNet-1k were eliminated was used as OOD datasets. The four OOD datasets are as follows: Textures [10], Places365 [61], iNaturalist [51], and SUN [54]. We used a pre-trained ResNet-50 model [15], which is trained with ImageNet-1k. The entire training dataset was used to estimate the contribution matrix $C$, and all images were resized to 224 × 224 at test time.

**Experimental Results.** In Table 2, we reported the performances of four OOD test datasets respectively. The average results from the four OOD test datasets were also reported. LINe outperformed all baselines including MSP [16], ODIN [30], Mahalanobis distance [28], Energy score [32], ReAct [44], DICE [45], and DICE + ReAct [45]. We compared LINe with Energy [32] first. LINe drastically reduced the FPR95 by 37.71%, which shows the benefit of leveraging important neurons under the same OOD scoring function. Next, we compared LINe with ReAct [44]. LINe reduced the FPR95 by 10.73%, which allows us to see the advantages of leveraging important neurons using Shapley value. LINe further outperformed recent DICE [45] and DICE + ReAct by 14.05% and 6.55%, respectively. Experimental results showed that the proposed method can be applied to real-world large dataset for OOD detection.

## 4.3. Ablation Study

In this section, we discuss the effectiveness of each part used in LINe and the detailed differences from other similar approaches. We also analyze the effect of hyperparameters.

### 4.3.1 Ablation Study of LINe on ImageNet

Table 3 shows an ablation study over various parts used in LINe. As shown in the table, each part of Shapley-based pruning (AP and WP) improved the performance. Comparing LINE w/o WP with Energy + AC allows us to see the advantages of reducing the noisy activation, which reduces FPR95 by 8.52%. Next, we compared LINe w/o AP with Energy + AC, which also shows the benefits of reducing noisy weights. Compared to Energy + AC, LINe w/o AP reduced FPR95 by 12.21%. Finally, we compared LINe w/o AP with DICE + ReAct [45] at Table 2, which allows us to see the benefit of leveraging class-wise contribution under similar circumstances. LINe w/o AP reduced the FPR95 by 4.06% from 27.25% to 23.19%. DICE + ReAct [45] uses activations to select important weights, while LINe w/o AP selects important weights using class-wise contribution derived from the Shapley value [43].

### 4.3.2 Effect of Changing AC Threshold on ImageNet

In Section 3.2, we discussed the meaning of AC in terms of the neuron-concept association. AC allows us to consider the number of activated high-level features in the penultimate layer. In Table 4, we show various OOD detection performances of the model by changing threshold $\delta$. Starting from clipping threshold $\delta = \infty$, the value of FPR95 is the highest in the table. As clipping threshold $\delta$ becomes

Table 2. **Comparison on ImageNet benchmark.** Comparisons with competitive post-hoc OOD detection methods on ImageNet benchmark. All values in this table are percentages and averaged over four OOD test datasets.

| **Method** | **OOD Datasets** | | | | | | | | **Average** | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **iNaturalist** | | **SUN** | | **Places** | | **Textures** | | | |
| | FPR95 ↓ | AUROC ↑ | FPR95 ↓ | AUROC ↑ | FPR95 ↓ | AUROC ↑ | FPR95 ↓ | AUROC ↑ | FPR95 ↓ | AUROC ↑ |
| MSP [16] | 54.99 | 87.74 | 70.83 | 80.86 | 73.99 | 79.76 | 68.00 | 79.61 | 66.95 | 81.99 |
| ODIN [30] | 47.66 | 89.66 | 60.15 | 84.59 | 67.89 | 81.78 | 50.23 | 85.62 | 56.48 | 85.41 |
| Mahalanobis [28] | 97.00 | 52.65 | 98.50 | 42.41 | 98.40 | 41.79 | 55.80 | 85.01 | 87.43 | 55.47 |
| Energy [32] | 55.72 | 89.95 | 59.26 | 85.89 | 64.92 | 82.86 | 53.72 | 85.99 | 58.41 | 86.17 |
| ReAct [44] | 20.38 | 96.22 | 24.20 | 94.20 | 33.85 | 91.58 | 47.30 | 89.80 | 31.43 | 92.95 |
| DICE [45] | 25.63 | 94.49 | 35.15 | 90.83 | 46.49 | 87.48 | 31.72 | 90.30 | 34.75 | 90.77 |
| DICE + ReAct [45] | 18.64 | 96.24 | 25.45 | 93.94 | 36.86 | 90.67 | 28.07 | 92.74 | 27.25 | 93.40 |
| **LINe (Ours)** | **12.26** | **97.56** | **19.48** | **95.26** | **28.52** | **92.85** | **22.54** | **94.44** | **20.70** | **95.03** |

Table 3. **Ablation Study of LINe on ImageNet.** Ablation on the effectiveness of Shapley-based pruning used in LINe and comparison with similar approaches. Values are percentages and averaged over OOD datasets. AC, AP, and WP denote activation clipping, activation pruning, and weight pruning, respectively.

| **Method** | **AC** | **AP** | **WP** | **FPR95↓** | **AUROC↑** |
|---|---|---|---|---|---|
| Energy [32] | | | | 58.41 | 86.17 |
| Energy + AC | ✓ | | | 35.40 | 91.86 |
| LINe w/o WP | ✓ | ✓ | | 26.88 | 93.77 |
| LINe w/o AP | ✓ | | ✓ | 23.19 | 94.57 |
| **LINe (Ours)** | ✓ | ✓ | ✓ | **20.70** | **95.03** |

Table 4. **Ablation on different thresholds ($\delta$) of AC.** Ablation on the different thresholds ($\delta$) of clipping. All values are percentages and averaged over multiple OOD test datasets.

| **Threshold ($\delta$)** | **FPR95 ↓** | **AUROC ↑** |
|---|---|---|
| $\delta = 0.1$ | 41.18 | 88.44 |
| $\delta = 0.4$ | 23.43 | 94.79 |
| $\delta = 0.8$ | **20.70** | **95.03** |
| $\delta = 1.0$ | 21.69 | 94.81 |
| $\delta = 1.5$ | 26.96 | 93.99 |
| $\delta = 2.0$ | 31.88 | 92.97 |
| $\delta = \infty$ (no AC) | 44.88 | 89.14 |

smaller, the OOD detection performance is improved. But at clipping threshold $\delta < 0.8$, the OOD detection performance dropped. This is because, as the clipping threshold $\delta$ approaches 0, all the penultimate output values also approach zero. It is obvious that performance will drop when all the penultimate output values approach zero.

#### 4.3.3 Effect of Changing Pruning Percentile on ImageNet

In this section, we conducted ablation studies on pruning percentiles ($p$) variation on ImageNet datasets as ID data. In Table 5 we show effect of changing pruning percentile ($p_w$ and $p_a$) on ImageNet. $p_a$ indicates pruning percentile for AP, $p_w$ indicates pruning percentile for WP. For a fixed WP percentile $p_w$ with extremely high value (e.g., $p_w$ = 90), the performance tends to increase when the AP percentile $p_a$ falls. Since LINe considers the number of activated features for detecting OOD samples, pruning most of the activations or weights has negatively impacted the performance. A lower pruning percentile is better to leverage differences in the number of activated features between ID and OOD samples. But to restrict the negative effect of noisy signals, we need some portions that can remove the noisy signals. As a result, the model performs the best when the pruning percentile is $p_w$ = 10 and $p_a$ = 10 on ImageNet.

Table 5. **Effect of changing pruning percentile ($p$) on ImageNet.** Ablation on the different pruning percentile ($p_a$ and $p_w$) of pruning. All values are percentages and averaged over multiple OOD test datasets. $p_a$ denotes pruning percentile for AP, $p_w$ indicates pruning percentile for WP.

| | $p_a$ = 90 | $p_a$ = 70 | $p_a$ = 50 | $p_a$ = 30 | $p_a$ = 10 |
|---|---|---|---|---|---|
| | **FPR95 ↓** | **FPR95 ↓** | **FPR95 ↓** | **FPR95 ↓** | **FPR95 ↓** |
| $p_w$ = 90 | 27.56 | 24.79 | 24.74 | 24.55 | 24.54 |
| $p_w$ = 70 | 27.45 | 25.93 | 25.81 | 27.45 | 33.32 |
| $p_w$ = 50 | 33.69 | 27.78 | 26.28 | 25.90 | 27.45 |
| $p_w$ = 30 | 27.46 | 26.10 | 27.17 | 24.36 | 28.43 |
| $p_w$ = 10 | 27.64 | 26.75 | 27.75 | 25.41 | **20.70** |

#### 4.3.4 Effect of Changing Pruning Percentile on CIFAR Benchmarks

In Table 6 and Table 7, we show effect of changing pruning percentile ($p_w$ and $p_a$) on CIFAR Benchmarks. Both tables show similar tendencies. For all AP percentile $p_a$, the performance tends to increase when the WP percentile $p_w$ increases. In Table 6, highest performance appeared at $p_w$ = 90 and $p_a$ = 90. On the other hand in Table 7, the highest performance appeared at $p_w$ = 90 and $p_a$ = 10. This result may seem to conflict with the result in Table 5, our observation in Table 8 can explain the cause of the difference.

Table 6. **Effect of changing pruning percentile ($p$) in CIFAR-10.** Ablation on the different pruning percentile ($p_a$ and $p_w$) of pruning. All values are percentages and averaged over multiple OOD test datasets. $p_a$ denotes pruning percentile for AP, $p_w$ indicates pruning percentile for WP.

| | $p_a$ = 90 | $p_a$ = 70 | $p_a$ = 50 | $p_a$ = 30 | $p_a$ = 10 |
|---|---|---|---|---|---|
| | **FPR95** ↓ | **FPR95** ↓ | **FPR95** ↓ | **FPR95** ↓ | **FPR95** ↓ |
| $p_w$ = 90 | **14.72** | 15.00 | 15.00 | 15.00 | 14.99 |
| $p_w$ = 70 | 14.80 | 15.12 | 15.12 | 15.12 | 15.10 |
| $p_w$ = 50 | 14.80 | 15.12 | 15.11 | 15.11 | 15.10 |
| $p_w$ = 30 | 14.80 | 15.12 | 15.11 | 15.12 | 15.10 |
| $p_w$ = 10 | 14.80 | 15.13 | 15.13 | 15.12 | 15.73 |

Table 7. **Effect of changing pruning percentile ($p$) in CIFAR-100.** Ablation on the different pruning percentile ($p_a$ and $p_w$) of pruning. All values are percentages and averaged over multiple OOD test datasets. $p_a$ denotes pruning percentile for AP, $p_w$ indicates pruning percentile for WP.

| | $p_a$ = 90 | $p_a$ = 70 | $p_a$ = 50 | $p_a$ = 30 | $p_a$ = 10 |
|---|---|---|---|---|---|
| | **FPR95** ↓ | **FPR95** ↓ | **FPR95** ↓ | **FPR95** ↓ | **FPR95** ↓ |
| $p_w$ = 90 | 38.75 | 37.81 | 37.81 | 37.75 | **35.67** |
| $p_w$ = 70 | 38.37 | 39.30 | 40.07 | 39.75 | 40.81 |
| $p_w$ = 50 | 38.37 | 39.19 | 40.54 | 40.27 | 42.14 |
| $p_w$ = 30 | 38.37 | 39.19 | 40.65 | 40.21 | 39.32 |
| $p_w$ = 10 | 38.40 | 39.31 | 40.76 | 40.91 | 38.17 |

Table 8. **Percentage of class-specific neuron overlap in multiple classes.** Difference between the percentage of class-specific neuron overlap in multiple classes on three data sets. For each dataset, we calculated the proportion of very important (top 10%) neurons in more than $o$% of the class. All values are percentages.

| **Overlap** | **CIFAR-10** | **CIFAR-100** | **ImageNet** |
|---|---|---|---|
| $o$ = 20 | 24.56 | 26.90 | 1.70 |
| $o$ = 30 | 23.39 | 0.58 | 0.15 |

### 4.3.5 Discussion

In Table 8, we compared the percentage of class-specific neuron overlap in multiple classes on three data sets. For each dataset, we calculated the proportion of very important (top 10%) neurons in more than $o$% of the class. These neurons activate in various classes which have semantically different features. Therefore, the higher proportion of these neurons can be seen as an overparameterized model with more numbers of generally activated neurons. These overparameterized models make OOD detection difficult by creating noisy signals, which can be reduced by leveraging AP and WP. To get optimal results from overparameterized models, we can lessen the effect of overparameterized weights with high WP percentile. Also, the effectiveness of considering the number of activated class-specific neurons can be maximized in the low AP percentile as a tendency shown in Table 7. But for a much more overparameterized model, which is shown in Table 6, we need to set high pruning percentile on both $p_w$ and $p_a$.

The degree of an overparameterized model can be understood from Section 4.1. We used the same pre-trained model architecture for evaluating CIFAR-10 and CIFAR-100. Of the two models with the same structure, we can intuitively understand that a model trained using a relatively small dataset is more likely to be overparameterized, which can also be observed in Table 8. The proportion of very important (top 10%) neurons in more than 30% of the class(i.e., $o$ = 30) on CIFAR-10 is the largest compared to other datasets. This observation shows that our pre-trained model used to evaluate CIFAR-10 is more overparameterized than other models we used.

## 5. Conclusion

In this paper, we propose a powerful OOD Detection method called LINe. LINe adopts a neural-concept association, which only uses important activations and weights selectively by measuring class-wise contribution from the Shapley value. Through LINe, we can effectively reduce the influence of the noise signal and make a difference in the overall OOD score between ID and OOD sample distribution. We conducted extensive experiments to demonstrate that LINe is superior to state-of-the-art OOD detection methods and effective on multiple datasets. From several theoretical studies and insights, we show how our method improves the performance of OOD detection. Our method is effective but has some limitations. It is fundamentally a trade-off relationship that pruning neurons to reduce the noisy output and considers the number of class-specific feature activations. Users have to examine the trade-off considering the degree of overparameterization of the model. We hope that as our study proposes an effective way to view OOD detection from a feature presentation perspective, attempts to understand the behavior of neural networks will be applied to multiple domains to discover other effective methods.

## Acknowledgements

This work was supported in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by the Korea Government (MSIT)(No. 2022-0-00078: Explainable Logical Reasoning for Medical Knowledge Generation, No. 2021-0-02068: Artificial Intelligence Innovation Hub, No. RS-2022-00155911: Artificial Intelligence Convergence Innovation Human Resources Development (Kyung Hee University)) and by the National Research Foundation of Korea (NRF) grant funded by the Korea government(MSIT) (No. 2021R1G1A1094990).

## References

[1] Davide Abati, Angelo Porrello, Simone Calderara, and Rita Cucchiara. Latent space autoregression for novelty detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 481–490, 2019. 1, 3

[2] Marco Ancona, Cengiz Oztireli, and Markus Gross. Explaining deep neural networks with a polynomial time algorithm for shapley value approximation. In *International Conference on Machine Learning*, pages 272–281. PMLR, 2019. 3

[3] Sarah Adel Bargal, Andrea Zunino, Vitali Petsiuk, Jianming Zhang, Kate Saenko, Vittorio Murino, and Stan Sclaroff. Guided zoom: Questioning network evidence for fine-grained classification. In *British Machine Vision Conference (BMVC)*, 2019. 3

[4] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6541–6549, 2017. 2, 3

[5] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Agata Lapedriza, Bolei Zhou, and Antonio Torralba. Understanding the role of individual units in a deep neural network. *Proceedings of the National Academy of Sciences*, 117(48):30071–30078, 2020. 2, 3

[6] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B Tenenbaum, William T Freeman, and Antonio Torralba. Gan dissection: Visualizing and understanding generative adversarial networks. *ICLR*, 2019. 3

[7] Abhijit Bendale and Terrance E Boult. Towards open set deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1563–1572, 2016. 1, 3

[8] Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K Su. This looks like that: deep learning for interpretable image recognition. *Advances in neural information processing systems*, 32, 2019. 3

[9] Xingyu Chen, Xuguang Lan, Fuchun Sun, and Nanning Zheng. A boundary based out-of-distribution classifier for generalized zero-shot learning. In *European Conference on Computer Vision*, pages 572–588. Springer, 2020. 1, 3

[10] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014. 1, 5, 6

[11] Terrance DeVries and Graham W Taylor. Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint arXiv:1802.04865*, 2018. 1, 3

[12] Amirata Ghorbani, James Wexler, James Y Zou, and Been Kim. Towards automatic concept-based explanations. *Advances in Neural Information Processing Systems*, 32, 2019. 3

[13] Amirata Ghorbani and James Y Zou. Neuron shapley: Discovering the responsible neurons. *Advances in Neural Information Processing Systems*, 33:5922–5932, 2020. 3

[14] Mara Graziani, Vincent Andrearczyk, and Henning Müller. Regression concept vectors for bidirectional explanations in histopathology. In *Understanding and Interpreting Machine Learning in Medical Image Computing Applications*, pages 124–132. Springer, 2018. 3

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Proceedings of the European conference on computer vision*, pages 630–645. Springer, 2016. 6

[16] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *Proceedings of International Conference on Learning Representations*, 2017. 1, 3, 6, 7

[17] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 4700–4708, 2017. 5

[18] Haiwen Huang, Zhihan Li, Lulu Wang, Sishuo Chen, Bin Dong, and Xinyu Zhou. Feature space singularity for out-of-distribution detection. *arXiv preprint arXiv:2011.14654*, 2020. 1, 3

[19] Rui Huang and Yixuan Li. Towards scaling out-of-distribution detection for large semantic space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 1, 3, 6

[20] Dihong Jiang, Sun Sun, and Yaoliang Yu. Revisiting flow generative models for out-of-distribution detection. In *International Conference on Learning Representations*, 2021. 1, 3

[21] Ashkan Khakzar, Soroosh Baselizadeh, Saurabh Khanduja, Christian Rupprecht, Seong Tae Kim, and Nassir Navab. Neural response interpretation through the lens of critical pathways. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13528–13538, 2021. 2, 3, 4

[22] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018. 1, 3

[23] Polina Kirichenko, Pavel Izmailov, and Andrew G Wilson. Why normalizing flows fail to detect out-of-distribution data. *Advances in neural information processing systems*, 33:20578–20589, 2020. 1, 3

[24] Ivan Kobyzev, Simon JD Prince, and Marcus A Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):3964–3979, 2020. 1, 3

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

[26] Harold William Kuhn and Albert William Tucker. *Contributions to the Theory of Games*. Number 28. Princeton University Press, 1953. 3

[27] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, pages 7167–7177, 2018. 1, 3

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

samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, pages 7167–7177, 2018. 1, 3, 6, 7

[29] Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International conference on learning representations*, 2018. 1, 3

[30] Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *Proceedings of International Conference on Learning Representations*, 2018. 3, 6, 7

[31] Ziqian Lin, Sreya Dutta Roy, and Yixuan Li. Mood: Multi-level out-of-distribution detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15313–15323, 2021. 3

[32] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems*, 33:21464–21475, 2020. 2, 3, 5, 6, 7

[33] Weizeng Lu, Xi Jia, Weicheng Xie, Linlin Shen, Yicong Zhou, and Jinming Duan. Geometry constrained weakly supervised object localization. In *European Conference on Computer Vision*, pages 481–496. Springer, 2020. 3

[34] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017. 3

[35] Yifei Ming, Yiyou Sun, Ousmane Dia, and Yixuan Li. Cider: Exploiting hyperspherical embeddings for out-of-distribution detection. *arXiv preprint arXiv:2203.04450*, 2022. 1, 3

[36] Peyman Morteza and Yixuan Li. Provable guarantees for understanding out-of-distribution detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 8, 2022. 3

[37] Jesse Mu and Jacob Andreas. Compositional explanations of neurons. *Advances in Neural Information Processing Systems*, 33:17153–17163, 2020. 3

[38] Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do deep generative models know what they don't know? *arXiv preprint arXiv:1810.09136*, 2018. 1, 3

[39] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. 5

[40] Stanislav Pidhorskyi, Ranya Almohsen, and Gianfranco Doretto. Generative probabilistic novelty detection with adversarial autoencoders. *Advances in neural information processing systems*, 31, 2018. 1, 3

[41] Jie Ren, Stanislav Fort, Jeremiah Liu, Abhijit Guha Roy, Shreyas Padhy, and Balaji Lakshminarayanan. A simple fix to mahalanobis distance for improving near-ood detection. *arXiv preprint arXiv:2106.09022*, 2021. 1, 3

[42] Mohammad Sabokrou, Mohammad Khalooei, Mahmood Fathy, and Ehsan Adeli. Adversarially learned one-class classifier for novelty detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3379–3388, 2018. 1, 3

[43] Lloyd S Shapley. A value for n-person games. *Classics in game theory*, 69, 1997. 2, 3, 4, 6

[44] Yiyou Sun, Chuan Guo, and Yixuan Li. React: Out-of-distribution detection with rectified activations. *Advances in Neural Information Processing Systems*, 34:144–157, 2021. 2, 5, 6, 7

[45] Yiyou Sun and Yixuan Li. Dice: Leveraging sparsification for out-of-distribution detection. In *European Conference on Computer Vision*, 2022. 2, 3, 5, 6, 7

[46] Yiyou Sun, Yifei Ming, Xiaojin Zhu, and Yixuan Li. Out-of-distribution detection with deep nearest neighbors. *arXiv preprint arXiv:2204.06507*, 2022. 1, 3

[47] Mukund Sundararajan and Amir Najmi. The many shapley values for model explanation. In *International conference on machine learning*, pages 9269–9278. PMLR, 2020. 3

[48] Engkarat Techapanurak, Masanori Suganuma, and Takayuki Okatani. Hyperparameter-free out-of-distribution detection using cosine similarity. In *Proceedings of the Asian Conference on Computer Vision*, 2020. 1, 3

[49] Joost Van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. Uncertainty estimation using a single deep deterministic neural network. In *International conference on machine learning*, pages 9690–9700. PMLR, 2020. 1, 3

[50] Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International conference on machine learning*, pages 1747–1756. PMLR, 2016. 1, 3

[51] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 8769–8778, 2018. 1, 6

[52] Andong Wang, Wei-Ning Lee, and Xiaojuan Qi. Hint: Hierarchical neuron concept explainer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10254–10264, 2022. 2, 3

[53] Haoran Wang, Weitang Liu, Alex Bocchieri, and Yixuan Li. Can multi-label classification networks know what they don't know? *Advances in Neural Information Processing Systems*, 34:29074–29087, 2021. 3

[54] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 3485–3492. IEEE Computer Society, 2010. 6

[55] Pingmei Xu, Krista A Ehinger, Yinda Zhang, Adam Finkelstein, Sanjeev R Kulkarni, and Jianxiong Xiao. Turkergaze: Crowdsourcing saliency with webcam based eye tracking. *arXiv preprint arXiv:1504.06755*, 2015. 5

[56] Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *arXiv preprint arXiv:2110.11334*, 2021. 2, 3

[57] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 5

[58] Alireza Zaeemzadeh, Niccolo Bisagno, Zeno Sambugaro, Nicola Conci, Nazanin Rahnavard, and Mubarak Shah. Out-of-distribution detection using union of 1-dimensional subspaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9452–9461, 2021. 1, 3

[59] Bolei Zhou, David Bau, Aude Oliva, and Antonio Torralba. Interpreting deep visual representations via network dissection. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2131–2145, 2018. 3

[60] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. *arXiv preprint arXiv:1412.6856*, 2014. 3

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

[62] Ev Zisselman and Aviv Tamar. Deep residual flow for out of distribution detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13994–14003, 2020. 1, 3

[63] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International conference on learning representations*, 2018. 1, 3