ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA240724

Perturb-and-Compare Approach for Detecting Out-of-Distribution Samples in Constrained Access Environments

Heeyoung Lee[a,1], Hoyoon Byun[b,1], Changdae Oh[c], JinYeong Bak[a,*] and Kyungwoo Song[b,*]

[a]Sungkyunkwan University, Suwon, South Korea
[b]Yonsei University, Seoul, South Korea
[c]University of Wisconsin–Madison, Madison, Wisconsin, United States

Abstract. Accessing machine learning models through remote APIs has been gaining prevalence following the recent trend of scaling up model parameters for increased performance. Even though these models exhibit remarkable ability, detecting out-of-distribution (OOD) samples remains a crucial safety concern for end users as these samples may induce unreliable outputs from the model. In this work, we propose an OOD detection framework, MixDiff, that is applicable even when the model's parameters or its activations are not accessible to the end user. To bypass the access restriction, MixDiff applies an identical input-level perturbation to a given target sample and a similar in-distribution (ID) sample, then compares the relative difference in the model outputs of these two samples. MixDiff is model-agnostic and compatible with existing output-based OOD detection methods. We provide theoretical analysis to illustrate MixDiff's effectiveness in discerning OOD samples that induce overconfident outputs from the model and empirically demonstrate that MixDiff consistently enhances the OOD detection performance on various datasets in vision and text domains.

1 Introduction

Recent developments in deep neural networks (DNNs) opened the floodgates for a wide adaptation of machine learning methods in various domains such as computer vision, natural language processing and speech recognition. As these models garner more users and widen their application area, the magnitude of impact that they may bring about when encountered with a failure mode is also amplified. One of the causes of these failure modes is when an out-of-distribution (OOD) sample is fed to the model. These samples are problematic because DNNs often produce unreliable outputs if there is a large deviation from the in-distribution (ID) samples that the model has been validated to perform well.

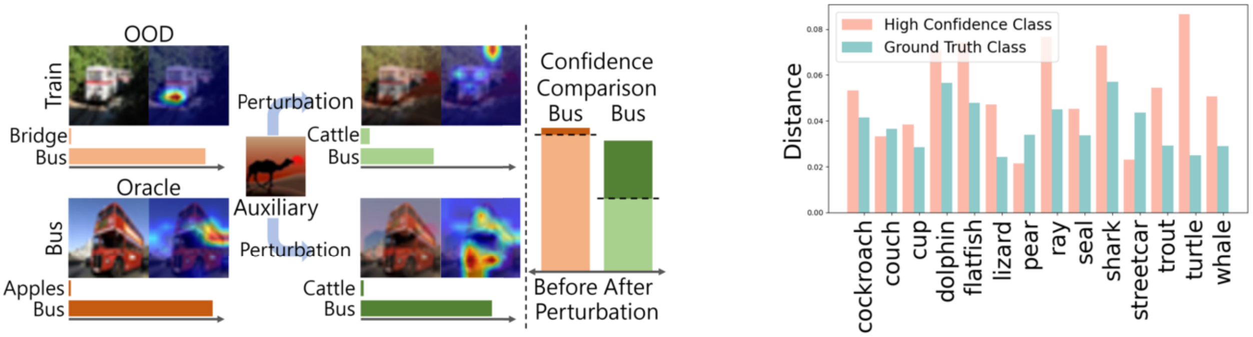
OOD detection is the task of determining whether an input sample is from ID or OOD. This work focuses on semantic shift [36] where distribution shift is manifested by samples of unseen class labels at test time. Several studies explore measuring how uncertain a model is about a target sample relying on the model's output [10, 20]. While these methods are desirable in that they do not assume access to the information inside the model, they can be further enhanced given access to the model's internal activations, [29] or its parameters [12]. However, the access to the model's internal states is not always permitted. With the advent of foundation models [26, 25], users often find themselves interacting with the model through remote APIs [27]. This limits the utilization of rich information inside the model [13], as well as the modification possibilities [28] that can be effectively used to detect OOD samples. In this work, we explore ways to bypass this access restriction through the only available modification point, namely, the models' inputs.

Data samples in the real world may contain distracting features that can negatively affect the model's performance. Sometimes these distractors may possess characteristics resembling a class that is different from the sample's true label. In this case, the model's predictions for an ID sample could become uncertain as it struggles to decide which class the sample belongs to. Similarly, the model could put too much emphasis on a feature that resembles a certain in-distribution characteristic from an OOD sample, outputting an overconfident prediction, even though the sample does not belong to any of the classes that the model was tasked to classify.

We start from the intuition that the contributing features in a misclassified sample, either misclassified as ID or OOD, will tend to be more sensitive to perturbations. In other words, these features that the model has overemphasized will be more brittle when compared to the actual characteristics of the class that these features resemble. Take as an example the image that is at the top left corner of Figure 1a. This sample is predicted to be a bus with a high confidence score, despite it belonging to an OOD class train. When we exact a perturbation to this sample by mixing it with some other auxiliary sample, the contribution of the regions that led to the model's initial prediction is significantly reduced as can be seen by the change in the class activation maps (CAM) [4]. However, when the same perturbation is applied to an actual image of a bus, the change is significantly less abrupt. The model's prediction scores show a similar behavior.

To experimentally verify the intuition, we collect OOD samples that induce high confidence scores from the model and compute CAMs for these samples before and after perturbation. Two versions of CAMs are computed with a zero-shot image classifier using CLIP model [26]. One with respect to the predicted class of the sample and the other with respect to the ground truth class of the sample. Figure 1b shows that the L_1 distance between the CAMs of the unperturbed

[1] Equal contribution.
[*] Corresponding Authors: jy.bak@skku.edu, kyungwoo.song@yonsei.ac.kr.



(a) (b)

Figure 1: **(a)** Class activation map of an OOD sample (train) for the predicted class (bus) exhibits a high degree of sensitivity when an auxiliary image (camel) is mixed to it. The same class activation map of an image of an actual bus is more robust to the same perturbation. (Top 2 classes are shown). **(b)** Average L_1 distance of the class activation maps of high confidence class and the ground truth class after perturbation (averaged over each OOD class).

and perturbed versions of an OOD sample's predicted class tends to be higher when compared to its ground truth class, even though the OOD sample had a high confidence score for that class. We provide experimental details in Appendix E [16].

Motivated by the above idea, we propose an OOD detection framework, MixDiff, that exploits this perturb-and-compare approach without any additional training. MixDiff employs a widely used data augmentation method Mixup [39] as the perturbation method so as to promote diverse interaction of features in the samples. Its overall procedure is outlined as follows: (1) perturb the target sample by applying Mixup with an auxiliary sample and get the model's prediction by feeding the perturbed target sample to the model; (2) perturb an ID sample of the predicted class of the target (oracle sample in Figure 1a) by following the same procedure; (3) compare the uncertainty scores of the perturbed samples. By comparing how the model's outputs of the target sample and a similar ID sample behave under the same perturbation, MixDiff augments the limited information contained in the model's prediction scores. This gives MixDiff the ability to better discriminate OOD and ID samples, even when the model's prediction scores for the original samples are almost identical.

We summarize our key contributions and findings as follows: (1) We propose an OOD detection framework, MixDiff, that enhances existing OOD scores in constrained access environments where only the models' inputs and outputs are accessible. (2) We provide a theoretical insight as to how MixDiff can mitigate the overconfidence issue of existing output-based OOD scoring functions. (3) MixDiff consistently improves various output-based OOD scoring functions when evaluated on OOD detection benchmark datasets in constrained access scenarios where existing methods' applicability is limited.

2 Related work

Output-based OOD scoring functions Various works propose OOD scoring functions measuring a classifier's uncertainty from its prediction scores. Some of these methods rely solely on the model's prediction probability. Maximum softmax probability (MSP) [10] utilizes the maximum value of the prediction distribution. Thulasidasan et al. [32] use Shannon entropy as a measure of uncertainty, while GEN [20] proposes a generalized version of the entropy score. KL Matching [11] finds the minimum KL divergence between the target and ID samples. D2U [37] measures the deviation of output distribution from the uniform distribution. If we take a step down to the logit space, maximum logit score (MLS) [11] utilizes the maximum value of the logits. Energy score [19] takes LogSumExp over the logits for the OOD score. MCM [22] emphasizes the importance of temperature scaling in vision-language models [26]. While these output-based methods are desirable in that they take a relaxed assumption on model accessibility, they suffer from the model's overconfidence issue [23]. This motivates us to investigate the perturb-and-compare approach as a calibration measure.

Enhancing output-based OOD scores Another line of work focuses on enhancing the aforementioned output-based OOD scores to make them more discriminative. ODIN [18] utilizes Softmax temperature scaling and gradient-based input preprocessing to enhance MSP [10]. ReAct [29] alleviates the overconfidence issue by clipping the model's activations if they are over a certain threshold. BAT [43] uses batch normalization [14] statistics for activation clipping. DICE [28] leverages weight sparsification to mitigate the overparameterization issue. Recently, methods that are based on activation [6] or weight pruning [1] approaches also have been proposed. These approaches effectively mitigate the overconfidence issue. However, all of these methods require access to either gradients, activations or parameters; hence limits their applicability in remote API environments. Our work stands out as an OOD score enhancement method in constrained access environments, where models' gradients, activations, and parameters are not accessible, leaving the model inputs as the only available modification point.

Utilization of deeper access for more discriminative OOD scores Several studies exploit the rich information that the feature space provides when designing OOD scores. Olber et al. [24], Zhang et al. [40] utilize ID samples' activations for comparison with a target sample. Models' inner representations are employed in methods that rely on class-conditional Mahalanobis distance [17]. ViM [34] proposes an OOD score that complements the energy score [19] with additional information from the feature space. Sun et al. [30] use the target sample's feature level KNN distance to ID samples. GradNorm [13] employs the gradient of the prediction probabilities' KL divergence to the uniform distribution. Zhang and Xiang [42] show that decoupling MLS [11] can lead to increased detection performance if given access to the model parameters. However, these methods are not applicable to black-box API models where one can only access the model's two endpoints, namely, the inputs and outputs.

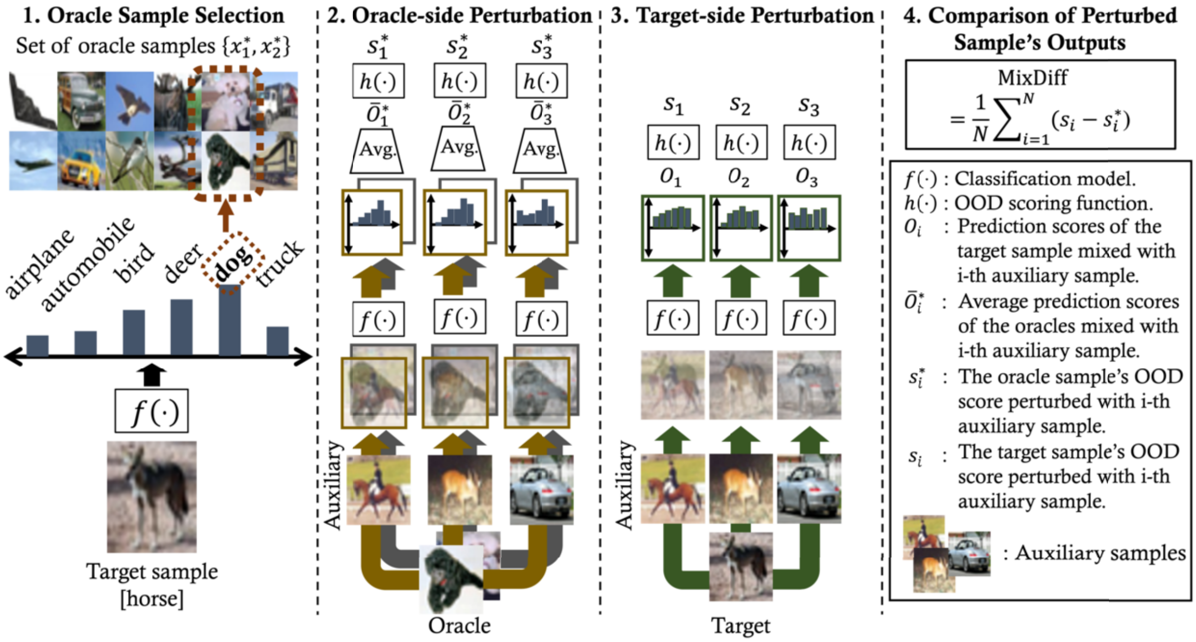

1. Oracle Sample Selection
Set of oracle samples $\{x_1^*, x_2^*\}$
airplane
automobile
bird
deer
dog
truck
$f(\cdot)$
Target sample
[horse]
2. Oracle-side Perturbation
s_1^*
s_2^*
s_3^*
$h(\cdot)$
$\bar{O}_1^*$
$\bar{O}_2^*$
$\bar{O}_3^*$
Avg.
Auxiliary
Oracle
3. Target-side Perturbation
s_1
s_2
s_3
O_1
O_2
O_3
Target
4. Comparison of Perturbed Sample's Outputs
MixDiff $= \frac{1}{N}\sum_{i=1}^{N}(s_i - s_i^*)$
$f(\cdot)$: Classification model.
$h(\cdot)$: OOD scoring function.
O_i : Prediction scores of the target sample mixed with i-th auxiliary sample.
$\bar{O}_i^*$: Average prediction scores of the oracles mixed with i-th auxiliary sample.
s_i^* : The oracle sample's OOD score perturbed with i-th auxiliary sample.
s_i : The target sample's OOD score perturbed with i-th auxiliary sample.
: Auxiliary samples

Figure 2: The overall figure of MixDiff with the number of Mixup ratios, $R = 1$, the number of classes, $K = 6$, the number of auxiliary samples, $N = 3$, and the number of oracle instances, $M = 2$. We omit Mixup ratio subscript r for simplicity.

3 Methodology

In this section, we describe the working mechanism of MixDiff framework. MixDiff is comprised of the following three procedures: (1) find ID samples that are similar to the target sample and perturb these samples by performing Mixup with an auxiliary sample; (2) perturb the target sample by performing Mixup with the same auxiliary sample; (3) measure the model's uncertainty of the perturbed target sample *relative* to the perturbed ID samples. We now provide a detailed description of each procedure.

Oracle-side perturbation We feed the given target sample, x_t, to a classification model $f(\cdot)$ and get its prediction scores for K classes, O_t, and the predicted class label, $\widehat{y}_t$, as shown in Equation 1.

$$O_t = f(x_t) \in \mathbb{R}^K, \quad \widehat{y}_t = \arg\max(O_t) \tag{1}$$

Next, we assume a small set of M labeled samples, $\Omega_k = \{(x_m^*, y_k^*)\}_{m=1}^{M}$, for each class label k. We refer to these samples as the oracle samples. From these, we take the samples that are of the same label as the predicted label $\widehat{y}_t$. Then, we perturb each oracle sample, x_m^*, by performing Mixup with an auxiliary sample, $x_i \in \{x_i\}_{i=1}^{N}$, with Mixup rate λ_r.

$$x_{mir}^* = \lambda_r x_m^* + (1 - \lambda_r) x_i, \text{ where } y_k^* = \widehat{y}_t \tag{2}$$

We feed the perturbed oracle sample to the classification model $f(\cdot)$ and get the model's prediction scores, $O_{mir}^* = f(x_{mir}^*) \in \mathbb{R}^K$. Then, we average the perturbed oracle samples' model outputs, to get $\bar{O}_{ir}^* = \frac{1}{M}\sum_{m=1}^{M} O_{mir}^*$. Finally, we compute the perturbed oracle samples' OOD score, $s_{ir}^* \in \mathbb{R}$, with an arbitrary output-based OOD scoring function $h(\cdot)$ such as MSP or MLS, *i.e.*, $s_{ir}^* = h\left(\bar{O}_{ir}^*\right) \in \mathbb{R}$.

Target-side perturbation We perturb the target sample x_t with the same auxiliary samples $\{x_i\}_{i=1}^{N}$, as $x_{ir} = \lambda_r x_t + (1 - \lambda_r) x_i$, and compute the OOD scores of the perturbed target sample as follows:

$$O_{ir} = f(x_{ir}) \in \mathbb{R}^K, \quad s_{ir} = h(O_{ir}) \in \mathbb{R} \tag{3}$$

Comparison of perturbed samples' outputs From the perturbed target's and oracles' uncertainty scores, (s_{ir}^*, s_{ir}), we calculate the MixDiff score for the target sample, x_t, as shown in Equation 4. It measures the model's uncertainty score of the target sample relative to similar ID samples when both undergo the same Mixup operation with an auxiliary sample x_i, then takes the average of the differences over the auxiliary samples and the Mixup ratios. We provide descriptions and illustrations of the overall procedure in Algorithm 1 and Figure 2.

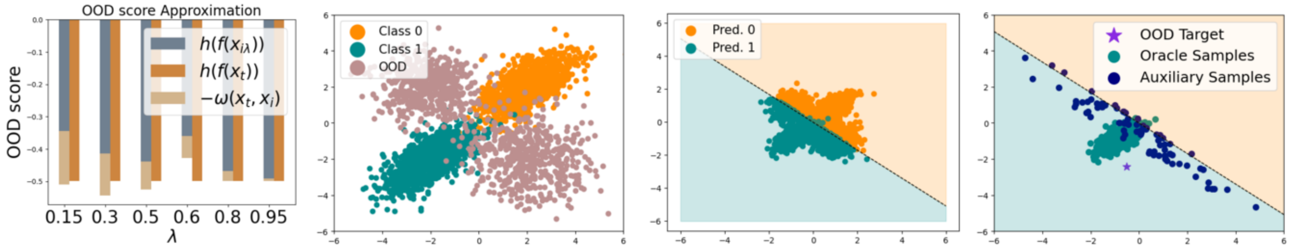
$$\text{MixDiff} = \frac{1}{RN}\sum_{r=1}^{R}\sum_{i=1}^{N}(s_{ir} - s_{ir}^*) \tag{4}$$

We calibrate the base OOD score for the target sample, $h(f(x_t))$, by adding the MixDiff score with a scaling hyperparameter γ to it so as to mitigate the model's over- or underconfidence issue.

Practical implementation The oracle-side procedure can be precomputed since it does not depend on the target sample. The target-side computations can be effectively parallelized since each perturbed target sample can be processed by the model, independent of the others. We organize the perturbed target samples in a single batch in our implementation (see Appendix F [16] for details on practical implementation). Further speedup can be gained in remote API environments as API calls are often handled by multiple nodes.

3.1 Theoretical analysis

To better understand how and when our method ensures performance improvements, we present a theoretical analysis of MixDiff. We use a similar theoretical approach to Zhang et al. [41], but towards a distinct direction for analyzing a post hoc OOD scoring function. Proposition 1 reveals the decomposition of the OOD score function into two components: the OOD score of the unmixed clean target sample and the supplementary signals introduced by Mixup.



(a) (b) (c) (d)

Figure 3: **(a)** Approximation error for Equation 5 on synthetic data. Without higher-order terms, we can reasonably approximate the OOD score of mixed sample with decomposed terms. **(b)** The syntactic data distribution. Data is sampled from four independent Gaussian distributions, with two considered as ID samples for each class and the other two as OOD samples. We train a logistic regression model with this dataset. **(c)** The prediction results of the trained model. **(d)** Although the target sample is a hard OOD sample, there are auxiliary samples (blue dot) that guarantee that MixDiff is positive under some reasonable conditions introduced in Theorem 1.

Algorithm 1 Computation of MixDiff Score

```
Require: target sample $x_t$, set of auxiliary samples $\{x_i\}_{i=1}^N$, set of Mixup rates $\{\lambda_r\}_{r=1}^R$, set of oracle samples for all $K$ classes $\{\Omega_k\}_{k=1}^K$ where $\Omega_k = \{(x_m^*, y_k^*)\}_{m=1}^M$, classifier model $f(\cdot)$, OOD scoring function $h(\cdot)$
$O_t = f(x_t)$
$\widehat{y_t} = \arg\max(O_t)$
$\{(x_m^*, y_k^*)\}_{m=1}^M \leftarrow \Omega_k$, where $y_k^* = \widehat{y_t}$
for $i \in \{1, \dots, N\}$ do
    for $r \in \{1, \dots, R\}$ do
        for $m \in \{1, \dots, M\}$ do
            $O_{mir}^* \leftarrow f(\lambda_r x_m^* + (1 - \lambda_r) x_i)$
        end for
        $s_{ir}^* \leftarrow h\left(\frac{1}{M} \sum_{m=1}^M O_{mir}^*\right)$
        $O_{ir} \leftarrow f(\lambda_r x_t + (1 - \lambda_r) x_i)$
        $s_{ir} \leftarrow h(O_{ir})$
    end for
end for
MixDiff $\leftarrow \frac{1}{RN} \sum_{r=1}^R \sum_{i=1}^N (s_{ir} - s_{ir}^*)$
```

Proposition 1 (OOD scores for mixed samples)**.** *Let pre-trained model $f(\cdot)$ and base OOD score function $h(\cdot)$ be twice-differentiable functions, and $x_{i\lambda} = \lambda x_t + (1 - \lambda) x_i$ be a mixed sample with ratio $\lambda \in (0, 1)$. Then OOD score function of mixed sample, $h(f(x_{i\lambda}))$, is written as:*

$$h(f(x_{i\lambda})) = h(f(x_t)) + \sum_{l=1}^{3} \omega_l(x_t, x_i) + \varphi_t(\lambda)(\lambda - 1)^2, \quad (5)$$

where $\lim_{\lambda \to 1} \varphi_t(\lambda) = 0,$

$$\omega_1(x_t, x_i) = (\lambda - 1)(x_t - x_i)^T f'(x_t) h'(f(x_t))$$

$$\omega_2(x_t, x_i) = \frac{(\lambda - 1)^2}{2} (x_t - x_i)^T f''(x_t)(x_t - x_i) h'(f(x_t))$$

$$\omega_3(x_t, x_i) = \frac{(\lambda - 1)^2}{2} (x_t - x_i)^T f'(x_t)(x_t - x_i)^T f'(x_t) h''(f(x_t)).$$

We analyze MixDiff using the quadratic approximation of $h(f(x_{i\lambda}))$, omitting the higher order terms denoted as $\varphi_t(\lambda)$ in Equation 5. In Figure 3a, we experimentally verify that the sum of the OOD score of the pure sample and ω terms, denoted as $\omega(x_t, x_i) = \sum_{l=1}^{3} \omega_l(x_t, x_i)$, reasonably approximates the OOD score of the mixed sample in Equation 5. $\omega(x_t, x_i)$ represents the impact caused by Mixup as can be seen from its increase when λ decreases. Hence, the additional signal from the Mixup can be derived from the first and second derivatives of $f(\cdot)$ and $h(\cdot)$ and the difference between the target and auxiliary samples.

We argue that perturbing both the target and oracle samples and then comparing the model outputs of the two can help OOD detection even when the target induces a relatively high confidence score from the model, in which case existing output-based OOD scoring functions would result in detection failure. Through Theorem 1, we show the effectiveness of MixDiff by demonstrating the existence of an auxiliary sample with which MixDiff can calibrate the overconfidence of a high confidence OOD sample on a simple linear model setup.

Theorem 1. *Let h(x) represent MSP and f(x) represent a linear model, described by $w^T x + b$, where $w, x \in \mathbb{R}^d$ and $b \in \mathbb{R}$. We consider the target sample, x_t, to be a hard OOD sample, defined as a sample that is predicted to be of the same class as the oracle sample, x_m, but with a higher confidence score than the oracle sample. For binary classification, x_t is a hard OOD sample when $0 < f(x_m) < f(x_t)$ or $f(x_t) < f(x_m) < 0$. There exists an auxiliary sample x_i such that*

$$h(f(x_t)) - h(f(x_m)) + \sum_{l=1}^{3} (\omega_l(x_t, x_i) - \omega_l(x_m, x_i)) > 0.$$

Theorem 1 provides a theoretical ground for our approach's effectiveness in discerning OOD samples that may not be detected by existing output-based OOD scores. Figures 3b to 3d illustrate examples of such auxiliary samples using synthetic data. Proof and details of Proposition 1 and Theorem 1 are in Appendix B and C [16], respectively. We also show that Theorem 1 holds for MLS and Entropy in Appendix C [16]. While we take a linear model as the classifier for simplicity of analysis, the prevalence of linear probing from foundation models' embeddings brings our analysis closer to real-world setups (see Section 4.5 for experimental validation).

4 Experiments

4.1 Experimental setup

We elaborate on the implementation details and present the descriptions on baselines. Other details on datasets and evaluation metrics are provided in Appendix G [16]. See Appendix O [16] for code.

Implementation details Following a recent OOD detection approach [7, 22, 35] that utilizes vision-language foundation models' zero-shot classification capability, we employ CLIP ViT-B/32 model

[26] as our classification model without any finetuning on ID samples. We construct the oracle set by randomly sampling M samples per class from the train split of each dataset. For a given target sample, we simply use the other samples in the same batch as the auxiliary set. Instead of searching hyperparameters for each dataset, we perform one hyperparameter search on Caltech101 [8] and use the same hyperparameters across all the other datasets, which is in line with a more realistic OOD detection setting [18]. We provide full description of the implementation details in Appendix G [16].

Baselines We take MSP [10], MLS [11], energy score [19], Shannon entropy [32] and MCM [22] as output-based training-free baselines. We also include methods that require extra training for comparison. ZOC [7] is a zero-shot OOD detection method based on CLIP [26] that requires training a separate candidate OOD class name generator. CAC [21] relies on train-time loss function modification and shows the best performance among the train-time modification methods compatible with CLIP [7]. We take CAC trained with the same CLIP ViT-B/32 backbone as a baseline (CLIP+CAC).

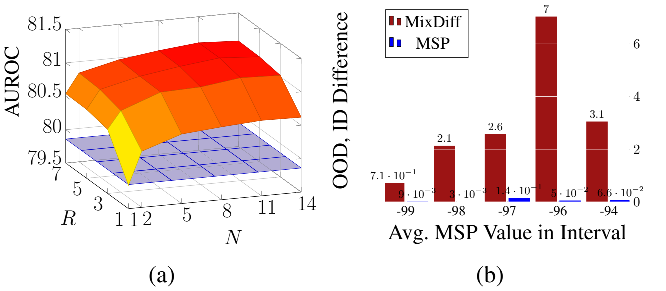
4.2 *Logits as model outputs*

First, we assume a more lenient access constraint whereby logits are provided as the model $f(\cdot)$'s outputs. This setup facilitates validation of MixDiff's OOD score enhancement ability on both the logit-based and probability-based scores. Note that, in this setup, the perturbed oracle samples' probability-based OOD scores are computed after averaging out M perturbed oracle samples in the logit-space, *i.e.*, $\bar{O}^*_{ir} = \frac{1}{M}\sum_{m=1}^{M} O^*_{mir}$. The consistent improvements across all datasets and methods in Table 1 indicate that MixDiff is effective in enhancing output-based OOD scores, to a degree where one of the training-free methods, MixDiff+MCM, outperforming a training-based method CLIP+CAC. Equipping MixDiff with the best performing non-training-free method, ZOC, also yields performance improvements.

4.3 *Prediction probabilities as model outputs*

We now take a more restricted environment where the only accessible part of the model is its output prediction probabilities. To the best of our knowledge, none of the existing OOD score enhancement methods are applicable in this environment. Logits are required in the case of Softmax temperature scaling [18]. ODIN's gradient-based input preprocessing [18] or weight pruning methods [28] assume an access to the model's parameters. The model's internal activations are required in the case of activation clipping [29] and activation pruning [6].

We take a linear combination of entropy and MSP scores with a scaling hyperparameter tuned on the Caltech101 dataset as a baseline (Entropy+MSP). The results are presented in Table 2. Even in this constrained environment, MixDiff effectively enhances output-based OOD scores, as evidenced by MixDiff+Entropy outperforming MCM (in Table 1), a method that assumes an access to the logit space, while MSP score fails to provide entropy score any meaningful performance gain. Figure 4a shows that MixDiff's performance gain can be enjoyed with as little as two additional forward passes ($R = 1$, $N = 2$). Figure 4b illustrates the discriminative edge provided by MixDiff score when the base OOD score's values are almost identical. We observe that the performance gain is more pronounced when the outputs contain more limited information as can be seen in the case of MSP where only the predicted class's probability value is utilized.



(a) (b)

Figure 4: Additional analyses on CIFAR100. **(a)** AUROC scores of MixDiff+Entropy with varying values of N and R (top). AUROC score of Entropy (bottom). We also provide processing time analysis in Appendix K [16]. **(b)** Difference of the OOD, ID samples' average uncertainty scores belonging to a given interval of MSP score. None-overlapping five consecutive intervals whose values lie below the threshold set by FPR95 are constructed. MixDiff scores can discriminate OOD, ID samples even when its base score values are almost identical.

Ablations We present the ablation results of MixDiff framework in Table 2 to illuminate each component's effect on performance. We take MixDiff+Entropy for these experiments. MixDiff's performance improvements are consistent when the homogeneity of auxiliary samples is gradually increased by changing the in-batch auxiliary samples, which may contain OOD samples, to random ID samples (*random ID as auxiliary*), and to the other oracle samples with the same predicted label as the target (*oracle as auxiliary*), suggesting that MixDiff is robust to the choice of auxiliary samples. Eliminating the comparison part in the perturb-and-compare approach by using only the perturbed target's scores without comparing with the perturbed oracles' scores (*without compare part*), and randomly choosing oracle samples from a set of ID sample instead of finding similar oracle samples using the predicted class label (*without oracle selection*) result in performance degradation. These observations suggest that comparing the relative change from a similar ID sample is crucial. We show that MixDiff is applicable even when there is no labeled oracle set by selecting top-M most similar samples from $M \times K$ unlabeled ID samples with similarity calculated from the dot product of the prediction probabilities of the target and the unlabeled oracle samples (*unlabeled oracle*).

4.4 *Prediction labels as model outputs*

We push the limits of the model access by assuming that only the predicted class labels are available without any scores attached to them. We apply MixDiff by representing the model's predictions as one-hot vectors and taking the difference between the perturbed target's predicted label and the corresponding perturbed oracles' average score for that label in Equation 4. As there is no base OOD score applicable in the environment, we use the MixDiff score alone. The results in Table 2 show that MixDiff is applicable even in this extremely constrained access environment.

4.5 *Last layer activations as model outputs*

We relax the model access constraint by permiting access to the model's activations from the last layer, *i.e.*, image embeddings in CLIP model. In this setup, instead of input-level Mixup, we utilize embedding-level Mixup. More specifically, embeddings of target (or oracle) are perturbed by mixing them with auxiliary sample's embeddings, after which logits are computed from the perturbed embed-

Table 1: Average AUROC scores for five datasets. The highest and second highest AUROC scores from each block are highlighted with **bold** and underline. The value on the right side of ± denotes the standard deviation induced from 5 different OOD, ID class splits. Statistically significant differences compared to the corresponding base score (indicated by background color) are *italicised* (one-tailed paired t-test with $p < 0.1$). Δ represents difference from the corresponding base score. † indicates the reduced evaluation setting described in Appendix G.4 [16]. We report AUCPR, FPR95 scores in Appendix H [16]. We report results on other CLIP backbones in Appendix M [16]. Best viewed in color.

Method	**Training-free**	**CIFAR10**	**CIFAR100**	**CIFAR+10**	**CIFAR+50**	**TinyImageNet**	**Avg.**	**Δ**
CSI [31]	✗	87.0±4.0	80.4±1.0	94.0±1.5	97.0	76.9±1.2	87.0	-
CAC [21]	✗	80.1±3.0	76.1±0.7	87.7±1.2	87.0	76.0±1.5	84.9	-
CLIP+CAC [21]	✗	89.3±2.0	**83.5**±1.2	96.5±0.5	95.8	**84.6**±1.7	89.9	-
ZOC † [7]	✗	91.5±2.5	82.7±2.8	97.6±1.1	97.1	82.6±3.1	90.3	-
MixDiff+ZOC †	✗	**92.2**±2.5	82.8±2.4	**98.2**±1.2	**98.5**	82.9±3.3	**90.9**	+0.6
MSP [10]	✓	88.7±2.0	78.2±3.1	95.0±0.8	95.1	80.4±2.5	87.5	-
MLS [11]	✓	87.8±3.0	80.0±3.1	96.1±0.8	96.0	84.0±1.2	88.8	-
Energy [19]	✓	85.4±3.0	77.6±3.7	94.9±0.9	94.8	83.2±1.2	87.2	-
Entropy [32]	✓	89.9±2.6	79.9±2.5	96.8±0.8	96.8	82.2±2.3	89.1	-
MCM [22]	✓	90.6±2.9	80.3±2.1	96.9±0.8	97.0	83.1±2.2	89.6	-
MixDiff+MSP	✓	*89.2±1.6*	*80.1±2.8*	*96.7±0.8*	96.9	*81.6±2.6*	88.9	+1.4
MixDiff+MLS	✓	87.9±2.1	80.5±2.2	*96.5±0.7*	96.9	**84.5**±0.9	89.3	+0.5
MixDiff+Energy	✓	85.6±2.2	78.3±2.7	*95.4±0.8*	95.9	83.6±1.1	87.8	+0.6
MixDiff+Entropy	✓	*90.7±1.8*	81.0±2.6	***97.6****±0.8*	97.6	82.9±2.4	90.0	+0.9
MixDiff+MCM	✓	**91.4**±1.8	**81.4**±2.6	*97.5*±0.9	**97.7**	83.9±2.2	**90.4**	+0.8

Table 2: AUROC scores on various degrees of model access scenarios. The methods in the bottom block require the model's inner activations and are evaluated with the same CLIP ViT-B/32 backbone and entropy as OOD scoring function. Best viewed in color.

Method	**Access**	**CIFAR10**	**CIFAR100**	**CIFAR+10**	**CIFAR+50**	**TinyImageNet**	**Avg.**	**Δ**
MSP [10]	Prediction prob.	88.7±2.0	78.2±3.1	95.0±0.8	95.1	80.4±2.5	87.5	-
Entropy [32]	Prediction prob.	89.9±2.6	79.9±2.5	96.8±0.8	96.8	82.2±2.3	89.1	-
Entropy+MSP	Prediction prob.	89.9±2.6	79.9±2.5	96.8±0.8	96.8	82.2±2.3	89.1	+0.0
MixDiff+MSP (Prediction prob.)	Prediction prob.	*89.4±1.3*	*80.0±2.8*	*96.5±0.8*	96.8	*81.8±2.4*	88.9	+1.4
MixDiff+Entropy (Prediction prob.)	Prediction prob.	***91.1****±1.6*	80.9±2.6	*97.1±0.8*	97.3	82.9±2.3	**89.9**	+0.8
with oracle as auxiliary	Prediction prob.	90.6±1.7	**81.1**±2.0	**97.3**±0.7	**97.4**	82.9±2.2	**89.9**	+0.8
with random ID as auxiliary	Prediction prob.	90.8±1.5	**81.1**±2.1	96.8±1.0	96.8	82.9±2.3	89.7	+0.6
with unlabeled oracle	Prediction prob.	91.0±1.6	80.5±2.9	97.1±0.8	97.3	82.7±2.1	89.7	+0.6
without compare part	Prediction prob.	89.4±2.9	79.5±2.7	97.1±0.9	97.2	81.6±2.5	89.0	-0.1
without oracle selection	Prediction prob.	89.5±2.8	79.6±2.7	97.1±0.9	97.3	81.7±2.5	89.0	-0.1
Random score from uniform dist.	Prediction label	49.6±0.5	49.8±1.1	49.8±0.7	50.1	49.8±0.4	49.8	-
MixDiff with random ID as auxiliary	Prediction label	**62.4**±4.1	**59.4**±6.2	**65.6**±1.5	**65.4**	**63.3**±2.8	**63.2**	+13.4
MixDiff with oracle as auxiliary	Prediction label	61.9±3.7	55.1±7.1	59.9±1.1	59.8	55.6±2.7	58.4	+8.6
MixDiff+MSP (Embedding Mixup)	Activation	*90.0*±1.8	80.0±3.6	*95.6*±0.8	95.7	*82.2*±2.3	88.7	+1.2
MixDiff+Entropy (Embedding Mixup)	Activation	***91.1***±2.0	**81.1**±3.2	***97.1***±0.7	**97.1**	*83.7*±2.2	**90.0**	+0.9
DML [42]	Activation	87.8±3.0	80.0±3.1	96.1±0.8	96.0	**84.0**±1.2	88.8	-
ASH [6]	Activation	85.2±3.8	75.4±4.4	92.5±0.9	92.4	77.2±3.1	84.5	-

dings and fed to an output-based OOD scoring function $h(\cdot)$ such as entropy. As auxiliaries' and oracles' embeddings are precomputed, the computational overhead introduced by MixDiff is almost nil. The assumption of linear model in theoretical analysis is more closely followed in embedding-level Mixup since they can be viewed as linear probing of foundation models' activations. Bottom block of Table 2 shows that MixDiff can enhance OOD detection performance even with negligible compute overhead in this relaxed setup. We use random ID samples as auxiliaries in the embedding Mixup experiments.

4.6 *Robustness to adversarial attacks*

In adversarial attack on an OOD detector, the attacker creates a small, indistinguishable modification to the input sample with the purpose of increasing the model's confidence of a given OOD sample or decreasing the model's confidence of a given ID sample [2]. These modifications can be viewed as injection of certain artificial features, specifically designed to induce more confident or uncertain outputs from the model. Our motivation in Section 1 suggests that these artificial features may also be less robust to perturbations. We test this by evaluating MixDiff under adversarial attack. The results in Table 3 indicate that the contributing features that induce ID/OOD misclassification are less robust to perturbations and that MixDiff can effectively exploit such brittleness. Detailed description of the experimental setup is in Appendix G.5 [16].

Table 3: AUROC scores on various attack scenarios. "In" (or "Out") indicates all of the ID (or OOD) samples are adversarially modified. "Both" indicates all of the ID, OOD samples are adversarially modified. "MixDiff Only" refers to the score in Equation 4 with entropy as the OOD scoring function $h(\cdot)$.

Method	**CIFAR10**				**CIFAR100**			
	Clean	**In**	**Out**	**Both**	**Clean**	**In**	**Out**	**Both**
Entropy	89.88	47.42	13.77	2.68	79.87	36.86	14.38	2.21
MixDiff+Entropy	**90.64**	54.71	31.77	8.84	**81.11**	50.31	31.40	9.08
MixDiff Only	88.16	**61.00**	**40.28**	**20.45**	78.05	**58.84**	**44.19**	**27.48**

4.7 *Experiments on out-of-scope detection task*

Out-of-scope detection We take the MixDiff framework to out-of-scope (OOS) detection task to check its versatility in regard to the modality of the input. To reliably fulfill users' queries or instructions, understanding the intent behind a user's utterance forms a crucial aspect of dialogue systems. In intent classification task, models are tasked to extract the intent behind a user utterance. Even though there has been an inflow of development in the area for the improvement of classification performance, there is no guarantee that a given query's intent is in the set of intents that the model is able to classify. OOS detection task [3], concerns with detection of such user utterances.

MixDiff with textual input Unlike images whose continuousness lends itself to a simple Mixup operation, the discreteness of texts renders Mixup of texts not as straightforward. While there are several works that explore interpolation of texts, most of these require access to the model parameters [15]. This limits the MixDiff framework's applicability in an environment where the model is served as an API [27], which is becoming more and more prevalent with the rapid development of large language models [25]. Following this trend, we assume a more challenging environment with the requirement that Mixup be performed on the input level. To this end, we simply concatenate the text pair and let the interpolation happen while the pair is inside the model [9].

Table 4: Average AUROC scores for out-of-scope detection task.

Method	CLINC150	Banking77	ACID	TOP	Average
MSP	93.02	85.43	88.98	90.01	89.36
MLS	93.56	85.02	88.91	90.06	89.39
Energy	93.61	84.99	88.83	90.06	89.37
Entropy	93.29	85.59	88.87	90.02	89.44
MixDiff+MSP	93.42	85.75	89.18	**90.68**	89.76
MixDiff+MLS	93.88	85.46	**89.24**	90.35	89.73
MixDiff+Energy	**93.89**	85.51	89.18	90.35	89.73
MixDiff+Entropy	93.67	**85.98**	89.13	**90.68**	**89.87**

Experimental setup We run OOS detection experiments using 4 intent classification datasets: CLINC150, Banking77, ACID, TOP. Following Zhan et al. [38], we randomly split the provided classes into in-scope and OOS intents, with in-scope intent class ratios of 25%, 50%, 75%. For the intent classification model, we finetune BERT-base model [5] on the in-scope split of each dataset's train set. For each in-scope ratio, we construct 10 in-scope, OOS splits with different random seeds. Detailed experimental setup is in Appendix G.6 [16].

Results We report the average AUROC scores in Table 4, each of which is averaged over the in-scope class ratios as well as the class splits. Even with a simple Mixup method that simply concatenates the text pair, MixDiff consistently improves the performance across diverse datasets. The results suggest that the MixDiff framework's applicability is not limited to images and that the framework can be applied to other modalities with an appropriate perturbation method.

5 Liminations and future work

Dependency on model's performance We construct a low-confidence oracle set by limiting the oracle pool to contain the top $p\%$ of most uncertain ID samples. Fig. 5 shows MixDiff's dependency on the model's ability to assign minimal confidence on the oracle. The experiments are performed with CIFAR10 dataset using the other oracle samples of the predicted class of the target as auxiliaries.

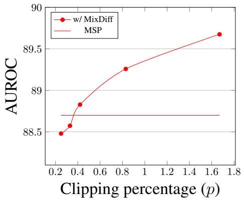


Figure 5: Effect of low-confidence oracles.

Time and space complexity MixDiff is effective at bypassing a black-box model's access restriction for OOD detection, but bypassing the access restriction comes with a certain computational overhead. For each target sample x_t, MixDiff requires processing of $N \times R$ mixed samples. While these samples can be effectively processed in parallel and the MixDiff framework outperforming the baselines only with small values of R and N, it nonetheless remains as a drawback of the MixDiff framework. Further research is called for reducing the computational and space complexity of MixDiff framework.

Selection of auxiliary samples In Section 4, we experiment with three auxiliary sample selection methods, one using the in-batch samples and the other two using the oracle or random ID samples as the auxiliary samples. Figure 4a shows reduced performance gain when the number of auxiliary samples, N, is too small. We hypothesize that this is due to the fact that while on average MixDiff can effectively discern the overemphasized features, there is a certain degree of variance in the MixDiff score, requiring N and, to some degree, R to be over a certain value for reliable performance. There may be an auxiliary sample that is more effective at discerning an overemphasized feature of a given target sample, but this is subject to change depending on the target sample. We leave the exploration of better auxiliary sample selection methods, either by careful curation of auxiliary samples or by making the procedure more instance-aware and possibly learnable, as future work.

Other forms of inputs MixDiff framework can be easily extended to incorporate inputs from other modalities. The experiments on the out-of-scope detection task serve as an example of these kinds of extensions. This input-level Mixup makes the framework applicable to environments where the access to the model parameters cannot be assumed. It also grants the freedom to design better Mixup methods that are specific to the format of the input or the task at hand. But this freedom comes at the cost of having to devise a Mixup mechanism for each input format and task. For example, the simple concatenation of samples that we have utilized on out-of-scope detection task has the limitation that it cannot be applied if the input sequence is too long due to the quadratic time and space complexity of Transformers [33].

Other types of distribution shifts and broader categories of models This work deals with detecting label shift with classifier models. However, there are other types of distribution shifts such as domain shift and broader range of models other than classifiers, *e.g.*, image segmentation models. Extensions of the perturb-and-compare mechanism to more diverse types of shifts and tasks would be a valuable addition to the black-box OOD detection field.

6 Conclusion

In this work, we present a new OOD detection framework, MixDiff, that boosts OOD detection performance in constrained access scenarios. MixDiff is based on the perturb-and-compare approach that measures how the model's confidence in the target sample behaves compared to a similar ID sample when both undergo an identical perturbation. This provides an additional signal that cannot be gained from the limited information of the target sample's model output alone. We provide theoretical grounds for the framework's effectiveness and empirically validate our approach on multiple degrees of restricted access scenarios. Our experimental results show that MixDiff is an effective OOD detection method for constrained access scenarios where the applicability of existing methods is limited.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2021R1F1A1060117, 2022R1A4A3033874) and ICT Creative Consilience Program through the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (IITP-2024-2020-0-01821).

References

[1] Y. H. Ahn, G.-M. Park, and S. T. Kim. Line: Out-of-distribution detection by leveraging important neurons. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[2] M. Azizmalayeri, A. S. Moakar, A. Zarei, R. Zohrabi, M. T. Manzuri, and M. H. Rohban. Your out-of-distribution detection method is not robust! In *Advances in Neural Information Processing Systems*, 2022.

[3] D. Chen and Z. Yu. GOLD: Improving out-of-scope detection in dialogues using data augmentation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021.

[4] P. Chen, Q. Li, S. Biaz, T. Bui, and A. Nguyen. gscorecam: What objects is clip looking at? In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2022.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019.

[6] A. Djurisic, N. Bozanic, A. Ashok, and R. Liu. Extremely simple activation shaping for out-of-distribution detection. In *The Eleventh International Conference on Learning Representations*, 2023.

[7] S. Esmaeilpour, B. Liu, E. Robertson, and L. Shu. Zero-shot out-of-distribution detection based on the pre-trained model clip. In *Proceedings of the AAAI conference on artificial intelligence*, number 6, 2022.

[8] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*. IEEE, 2004.

[9] X. Hao, Y. Zhu, S. Appalaraju, A. Zhang, W. Zhang, B. Li, and M. Li. Mixgen: A new multi-modal data augmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*, 2023.

[10] D. Hendrycks and K. Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2017.

[11] D. Hendrycks, S. Basart, M. Mazeika, M. Mostajabi, J. Steinhardt, and D. X. Song. Scaling out-of-distribution detection for real-world settings. In *International Conference on Machine Learning*, 2022.

[12] Y. C. Hsu, Y. Shen, H. Jin, and Z. Kira. Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[13] R. Huang, A. Geng, and Y. Li. On the importance of gradients for detecting distributional shifts in the wild. In *Advances in Neural Information Processing Systems*, 2021.

[14] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2015.

[15] F. Kong, R. Zhang, X. Guo, S. Mensah, and Y. Mao. DropMix: A textual data augmentation combining dropout with mixup. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022.

[16] H. Lee, H. Byun, C. Oh, J. Bak, and K. Song. Perturb-and-compare approach for detecting out-of-distribution samples in constrained access environments. *arXiv*, abs/2408.10107, 2024.

[17] K. Lee, K. Lee, H. Lee, and J. Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, 2018.

[18] S. Liang, Y. Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*, 2018.

[19] W. Liu, X. Wang, J. Owens, and Y. Li. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems*, 2020.

[20] X. Liu, Y. Lochman, and Z. Chrsitopher. Gen: Pushing the limits of softmax-based out-of-distribution detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[21] D. Miller, N. Sunderhauf, M. Milford, and F. Dayoub. Class anchor clustering: A loss for distance-based open set recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021.

[22] Y. Ming, Z. Cai, J. Gu, Y. Sun, W. Li, and Y. Li. Delving into out-of-distribution detection with vision-language representations. *Advances in Neural Information Processing Systems*, 2022.

[23] A. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.

[24] B. Olber, K. Radlak, A. Popowicz, M.Szczepankiewicz, and K. Chachula. Detection of out-of-distribution samples using binary neuron activation patterns. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[25] OpenAI. Gpt-4 technical report, 2023.

[26] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*. PMLR, 2021.

[27] T. Sun, Y. Shao, H. Qian, X. Huang, and X. Qiu. Black-box tuning for language-model-as-a-service. In *Proceedings of ICML*, 2022.

[28] Y. Sun and Y. Li. Dice: Leveraging sparsification for out-of-distribution detection. In *European Conference on Computer Vision*, 2022.

[29] Y. Sun, C. Guo, and Y. Li. React: Out-of-distribution detection with rectified activations. In *Advances in Neural Information Processing Systems*, 2021.

[30] Y. Sun, Y. Ming, X. Zhu, and Y. Li. Out-of-distribution detection with deep nearest neighbors. *ICML*, 2022.

[31] J. Tack, S. Mo, J. Jeong, and J. Shin. Csi: Novelty detection via contrastive learning on distributionally shifted instances. *Advances in neural information processing systems*, 2020.

[32] S. Thulasidasan, S. Thapa, S. Dhaubhadel, G. Chennupati, T. Bhattacharya, and J. A. Bilmes. An effective baseline for robustness to distributional shift. *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2021.

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

[34] H. Wang, Z. Li, L. Feng, and W. Zhang. Vim: Out-of-distribution with virtual-logit matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[35] H. Wang, Y. Li, H. Yao, and X. Li. Clipn for zero-shot ood detection: Teaching clip to say no. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[36] J. Yang, K. Zhou, Y. Li, and Z. Liu. Generalized out-of-distribution detection: A survey. *arXiv preprint arXiv:2110.11334*, 2021.

[37] E. Yilmaz and C. Toraman. D2U: Distance-to-uniform learning for out-of-scope detection. In *2022 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2022.

[38] L.-M. Zhan, H. Liang, B. Liu, L. Fan, X.-M. Wu, and A. Y. Lam. Out-of-scope intent detection with self-supervision and discriminative training. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021.

[39] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

[40] J. Zhang, Q. Fu, X. Chen, L. Du, Z. Li, G. Wang, xiaoguang Liu, S. Han, and D. Zhang. Out-of-distribution detection based on in-distribution data patterns memorization with modern hopfield energy. In *The Eleventh International Conference on Learning Representations*, 2023.

[41] L. Zhang, Z. Deng, K. Kawaguchi, A. Ghorbani, and J. Zou. How does mixup help with robustness and generalization? In *International Conference on Learning Representations*, 2021.

[42] Z. Zhang and X. Xiang. Decoupling maxlogit for out-of-distribution detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[43] Y. Zhu, Y. Chen, C. Xie, X. Li, R. Zhang, H. Xue', X. Tian, bolun zheng, and Y. Chen. Boosting out-of-distribution detection with typical features. In *Advances in Neural Information Processing Systems*, 2022.