Fast Decision Boundary based Out-of-Distribution Detector

Litian Liu [1] **Yao Qin** [2]

Abstract

Efficient and effective Out-of-Distribution (OOD) detection is essential for the safe deployment of AI systems. Existing feature space methods, while effective, often incur significant computational overhead due to their reliance on auxiliary models built from training features. In this paper, we propose a computationally-efficient OOD detector without using auxiliary models while still leveraging the rich information embedded in the feature space. Specifically, we detect OOD samples based on their feature distances to decision boundaries. To minimize computational cost, we introduce an efficient closed-form estimation, analytically proven to tightly lower bound the distance. Based on our estimation, we discover that In-Distribution (ID) features tend to be further from decision boundaries than OOD features. Additionally, ID and OOD samples are better separated when compared at equal deviation levels from the mean of training features. By regularizing the distances to decision boundaries based on feature deviation from the mean, we develop a hyperparameter-free, auxiliary model-free OOD detector. Our method matches or surpasses the effectiveness of state-of-the-art methods in extensive experiments while incurring negligible overhead in inference latency. Overall, our approach significantly improves the efficiency-effectiveness trade-off in OOD detection. Code is available at: https://github.com/litianliu/fDBD-OOD.

1. Introduction

As machine learning models are increasingly deployed in the real world, it is inevitable to encounter samples out of the training distribution. Since a classifier cannot make meaningful predictions on test samples from classes unseen during training, the detection of Out-of-Distribution (OOD) samples is crucial for taking necessary precautions. The field of OOD detection, which has recently seen a surge in research interest (Yang et al., 2021a), divides into two main areas. One area investigates the training time regularization to enhance OOD detection (Wei et al., 2022; Huang & Li, 2021; Ming et al., 2023), while our work, along with others, delves into *post-hoc* methods, which are training-agnostic and suitable for ready implementation on pre-trained models. OOD detectors can be designed over model output space (Liang et al., 2018; Liu et al., 2020; Hendrycks et al., 2019). Additionally, Tack et al. (2020); Lee et al. (2018); Sun et al. (2022) and Sastry & Oore (2020) use the clustering of In-Distribution (ID) samples in the feature space for OOD detection. For example, Lee et al. (2018) fit a multivariate Gaussian over the training features and detect OOD based on the Mahalanobis distance, and Sun et al. (2022) detect OOD based on the k-th nearest neighbor (KNN) distance to the training features. While existing feature-space methods are highly effective, their reliance on auxiliary models built from training features incurs additional computational costs. This poses a challenge for time-critical real-world applications, such as autonomous driving, where the latency of OOD detection becomes a top priority.

In this work, we focus on designing post-hoc OOD detectors for pre-trained classifiers. We aim to leverage the rich

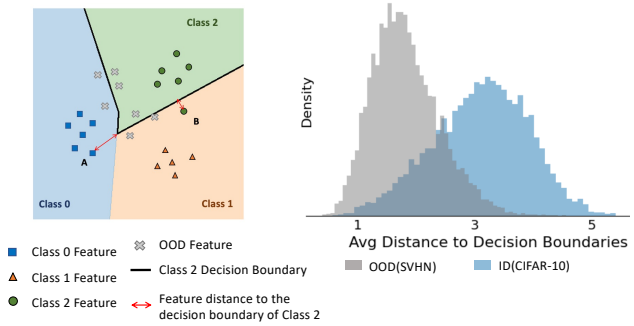

Figure 1. **Overview.** *Left: Conceptual Illustration.* The feature distance to decision boundaries on a multi-class classifier's penultimate layer, quantifying the perturbation magnitude needed to alter the model prediction to a class (see formal definition in Section 3.1). *Right: Empirical Observation.* Features of ID samples (CIFAR-10) tend to reside further from decision boundaries than OOD samples (SVHN). The distances are measured using our method (see Section 3.1) and averages are per sample.

[1]MIT [2]UC Santa Barbara. Correspondence to: Litian Liu <litianl@mit.edu>, Yao Qin <yaoqin@ucsb.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

information in the feature space while optimizing computational efficiency and avoiding the need for auxiliary models built from training statistics. To this end, we study from the novel perspective of decision boundaries, which naturally summarizes the training statistics. We begin by asking:

Where do features of ID and OOD samples reside with respect to the decision boundaries?

To answer the question, we first formalize the concept of the feature distance to a class's decision boundary. We define the distance as the minimum perturbation in the feature space to change the classifier's decision to the class, visually explained in Figure 1 *Left*. In particular, we focus on the penultimate layer, *i.e.*, the layer before the linear classification head. Due to non-convexity, the distance on the penultimate layer cannot be readily computed. To minimize the cost of measuring the distance, we introduce in Section 3.1 an efficient closed-form estimation, analytically proven to tightly lower bound the distance. Intuitively, feature distances to decision boundaries reflects the difficulty of changing model decisions and can quantify model uncertainty in the feature space. Unlike output space *softmax* confidence, our feature-space distance uses the rich information embedded in the feature space for OOD detection.

Based on our closed-form distance estimation, we pioneeringly explore OOD detection from the perspective of decision boundaries. Intuitively, features of ID samples would reside further away from the decision boundaries than OOD samples, since a classifier is likely to be more decisive in ID samples. We empirically validate our intuition in Figure 1 (*Right*). Further, we observe that ID and OOD can be better separated when compared at equal deviation levels from the mean of training features. Using the deviation level as a regularizer, we design our detection score as a regularized average feature distance to decision boundaries. The lower the score is, the closer the feature is to decision boundaries, and the more likely the sample is OOD.

Thresholding on the detection scores, we have **f**ast **D**ecision **B**oundary based OOD **D**etector (`fDBD`). Our detector is hyperparameter-free and auxiliary model-free, eliminating the cost of tuning parameters and reducing the inference overhead. Moreover, `fDBD` scales linearly with the number of classes and the feature dimension, theoretically guaranteed to be computationally scalable for large-scale tasks. In addition, `fDBD` incooperates class-specific information from the class decision boundary perspective to improve OOD detection effectiveness.

With extensive experiments, we demonstrate the superior efficiency and effectiveness of our method across various OOD benchmarks on different classification tasks (ImageNet (Deng et al., 2009), CIFAR-10 (Krizhevsky et al., 2009)), diverse training objectives (cross-entropy & supervised contrastive loss (Khosla et al., 2020)), and a range of network architectures (ResNet (He et al., 2016) & ViT (Dosovitskiy et al., 2020) & DenseNet (Huang et al., 2017)). Notably, our `fDBD` consistently achieves or surpasses state-of-the-art OOD detection performance. In the meantime, `fDBD` maintains inference latency comparable to the vanilla *softmax-confidence* detector, inducing practically negligible overhead in inference latency. Overall, our method significantly improves upon the efficiency-effectiveness trade-off of existing methods. We summarize our main contributions below:

- **Closed-form Estimation of the Feature Distance to Decision Boundaries** In Section 3.1, we formalize the concept of the feature distance to decision boundaries. We introduce an efficient and effective closed-form estimation method to measure the distance, providing a beneficial tool for the community.
- **Fast Decision Boundary based OOD Detector:** Using our estimation method in Section 3.1, we establish in Section 3.2 the first empirical observation that ID features tend to reside further from decision boundaries than OOD features. This ID/OOD separation is enhanced when regularized by the feature deviation from the training feature mean. Based on the observation, we propose a hyperparameter-free, auxiliary model-free, and computationally efficient OOD detector from the novel perspective of decision boundaries.
- **Experimental analysis:** In Section 4, we demonstrate across extensive experiments that fDBD achieves or surpasses the state-of-the-art OOD detection effectiveness with negligible latency overhead.
- **Theoretical analysis:** We theoretically guarantee the computational efficiency of `fDBD` through complexity analysis. Additionally, we support the effectiveness of our `fDBD` through theoretical analysis in Section 5.

2. Problem Setting

We consider a data space $\mathcal{X}$, a class set $\mathcal{C}$, and a classifier $f : \mathcal{X} \to \mathcal{C}$, which is trained on samples *i.i.d.* drawn from joint distribution $\mathbb{P}_{\mathcal{X}\mathcal{C}}$. We denote the marginal distribution of $\mathbb{P}_{\mathcal{X}\mathcal{C}}$ on $\mathcal{X}$ as $\mathbb{P}^{in}$. And we refer to samples drawn from $\mathbb{P}^{in}$ as In-Distribution (ID) samples. In practice, the classifier f may encounter $\boldsymbol{x} \in \mathcal{X}$ which is not drawn from $\mathbb{P}^{in}$. We say such samples are Out-of-Distribution (OOD).

Since a classifier cannot make meaningful predictions on OOD samples from classes unseen during training, it is important to distinguish between such OOD samples and ID samples for deployment reliability. Additionally, for time-critical applications, it is crucial to detect OOD samples promptly to take precautions. Instead of using the clustering of ID features and building auxiliary models as in prior art (Lee et al., 2018; Sun et al., 2022), we alternatively investigate OOD-ness from the perspective of decision boundaries, which inherently captures the training ID statistics.

3. Detecting OOD using Decision Boundaries

To understand the potential of detecting OOD from the decision boundaries perspective, we ask:

Where do features of ID and OOD samples reside with respect to the decision boundaries?

To this end, we first define the feature distance to decision boundaries in a multi-class classifier. We then introduce an efficient and effective method for measuring the distance using closed-form estimation. Using our method, we observe that the ID features tend to reside further away from the decision boundaries. Accordingly, we propose a decision boundary-based OOD detector. Our detector is post-hoc and can be built on top of any pre-trained classifiers, agnostic to model architecture, training procedure, and OOD types. In addition, our detector is hyperparameter-free, auxiliary model-free, and computationally efficient.

3.1. Measuring Feature Distance to Decision Boundaries

We now formalize the concept of the feature distance to the decision boundaries. We denote the last layer function of f as $f_{-1}: \mathcal{Z} \to \mathcal{C}$, which maps a penultimate feature vector $\boldsymbol{z}$ into a class c. Since f_{-1} is linear, we can express f_{-1} as:

$$f_{-1}(\boldsymbol{z}) = \arg\max_{c \in \mathcal{C}} \boldsymbol{w}_c^T \boldsymbol{z} + b_c,$$

where $\boldsymbol{w}_c$ and b_c are parameters corresponding to class c.

Definition 3.1. On the penultimate space of classifier f, we define the $L2$-distance of feature embedding $\boldsymbol{z}_{\boldsymbol{x}}$ for sample $\boldsymbol{x}$ to the decision boundary of class c, where $c \neq f(\boldsymbol{x})$, as:

$$D_f(\boldsymbol{z}_{\boldsymbol{x}}, c) = \inf_{\{\boldsymbol{z}': f_{-1}(\boldsymbol{z}')=c\}} \|\boldsymbol{z}_{\boldsymbol{x}} - \boldsymbol{z}'\|_2 .$$

Here, $\{\boldsymbol{z}' : f_{-1}(\boldsymbol{z}') = c\}$ is the decision region of class c in the penultimate space. Therefore, the distance we defined is the minimum perturbation required to change the model's decision to class c. Intuitively, the metric quantifies the difficulty of altering the model's decision.

As the decision region is *non-convex* in general as shown in Figure 1, the feature distance to a decision boundary in Definition 3.1 does not have a closed-form solution and cannot be readily computed. To circumvent computationally expensive iterative estimation, we relax the decision region and propose an efficient and effective estimation method for measuring the distance.

Theorem 3.2. *On the penultimate space of classifier f, the $L2$-distance between feature embedding $\boldsymbol{z}_{\boldsymbol{x}}$ of sample $\boldsymbol{x}$ and the decision boundary of class c, where $c \neq f(\boldsymbol{x})$, i.e. $D_f(\boldsymbol{z}_{\boldsymbol{x}}, c)$, is tightly lower bounded by*

$$\tilde{D}_f(\boldsymbol{z}_{\boldsymbol{x}}, c) \coloneqq \frac{|(\boldsymbol{w}_{f(\boldsymbol{x})} - \boldsymbol{w}_c)^T \boldsymbol{z}_{\boldsymbol{x}} + (b_{f(\boldsymbol{x})} - b_c)|}{\left\|\boldsymbol{w}_{f(\boldsymbol{x})} - \boldsymbol{w}_c\right\|_2}, \quad (1)$$

where $\boldsymbol{z}_{\boldsymbol{x}}$ is the penultimate space feature embedding of $\boldsymbol{x}$ under classifier f, $\boldsymbol{w}_{f(\boldsymbol{x})}$ and $b_{f(\boldsymbol{x})}$ are parameters of the linear classifier corresponding to the predicted class $f(\boldsymbol{x})$.

Proof. For any class c, $c \neq f(x)$, let

$$\begin{aligned}\mathcal{Z}_c :=& \{\boldsymbol{z} : f_{-1}(\boldsymbol{z}') = c\} \\ =& \{\boldsymbol{z} : \boldsymbol{w}_c^T \boldsymbol{z} + b_c > \boldsymbol{w}_{c'}^T \boldsymbol{z} + b_{c'} \ \forall c' \neq c\}; \\ \mathcal{Z}'_c :=& \{\boldsymbol{z}' : \boldsymbol{w}_c^T \boldsymbol{z}' + b_c > \boldsymbol{w}_{f(\boldsymbol{x})}^T \boldsymbol{z}' + b_{f(\boldsymbol{x})}\}.\end{aligned}$$

Observe that $\mathcal{Z}_c \subseteq \mathcal{Z}'_c$. Therefore, we have

$$D_f(\boldsymbol{z}_{\boldsymbol{x}}, c) = \inf_{\boldsymbol{z} \in \mathcal{Z}_c} \|\boldsymbol{z} - \boldsymbol{z}_{\boldsymbol{x}}\|_2 \geq \inf_{\boldsymbol{z}' \in \mathcal{Z}'_c} \|\boldsymbol{z}' - \boldsymbol{z}_{\boldsymbol{x}}\|_2 . \quad (2)$$

Note that geometrically $\inf_{\boldsymbol{z}' \in \mathcal{Z}'_c} \|\boldsymbol{z}' - \boldsymbol{z}_{\boldsymbol{x}}\|_2$ represents the l_2 distance from $\boldsymbol{z}_{\boldsymbol{x}}$ to hyperplane

$$(\boldsymbol{w}_{f(\boldsymbol{x})} - \boldsymbol{w}_c)^T \boldsymbol{z} + (b_{f(\boldsymbol{x})} - b_c) = 0, \quad (3)$$

and thus

$$\inf_{\boldsymbol{z}' \in \mathcal{Z}'_c} \|\boldsymbol{z}' - \boldsymbol{z}_{\boldsymbol{x}}\|_2 = \frac{|(\boldsymbol{w}_{f(\boldsymbol{x})} - \boldsymbol{w}_c)^T \boldsymbol{z}_{\boldsymbol{x}} + (b_{f(\boldsymbol{x})} - b_c)|}{\left\|\boldsymbol{w}_{f(\boldsymbol{x})} - \boldsymbol{w}_c\right\|_2}. \quad (4)$$

Combining Eqn. (4) with Eqn. (2), we conclude that Eqn. (1) lower bounds $D_f(\boldsymbol{z}_{\boldsymbol{x}}, c)$.

We now show that equality in Eqn. (2) holds for class c_2, corresponding to the nearest hyperplane to the sample embedding $\boldsymbol{z}_{\boldsymbol{x}}$, i.e.,

$$c_2 := \arg \min_{c \in \mathcal{C}, c \neq f(\boldsymbol{x})} \inf_{\boldsymbol{z}' \in \mathcal{Z}'_c} \|\boldsymbol{z}' - \boldsymbol{z}_{\boldsymbol{x}}\|_2 . \quad (5)$$

Let the projection of $\boldsymbol{z}_{\boldsymbol{x}}$ on the nearest hyperplane be $\boldsymbol{p}_{\boldsymbol{x}}$. From Eqn. (5), for all $c \notin \{c_2, f(\boldsymbol{x})\}$, we have

$$\|\boldsymbol{p}_{\boldsymbol{x}} - \boldsymbol{z}_{\boldsymbol{x}}\|_2 = \inf_{\boldsymbol{z}' \in \mathcal{Z}'_{c_2}} \|\boldsymbol{z}' - \boldsymbol{z}_{\boldsymbol{x}}\|_2 \leq \inf_{\boldsymbol{z}' \in \mathcal{Z}'_c} \|\boldsymbol{z}' - \boldsymbol{z}_{\boldsymbol{x}}\|_2 . \quad (6)$$

Consequently, we have $\boldsymbol{p}_{\boldsymbol{x}} \in \mathcal{Z}'^{\complement}_c$, i.e. $\boldsymbol{w}_c^T \boldsymbol{p}_{\boldsymbol{x}} + b_c \leq \boldsymbol{w}_{f(\boldsymbol{x})}^T \boldsymbol{p}_{\boldsymbol{x}} + b_{f(\boldsymbol{x})}$ for any $c \notin \{f(\boldsymbol{x}), c_2\}$. Intuitively, as all other hyperplanes are further away from $\boldsymbol{z}_{\boldsymbol{x}}$ than $\boldsymbol{p}_{\boldsymbol{x}}$, $\boldsymbol{p}_{\boldsymbol{x}}$ and $\boldsymbol{z}_{\boldsymbol{x}}$ must fall on the same side of each hyperplane. Therefore, $p_{\boldsymbol{x}}$ falls within the closure of $\mathcal{Z}_{c_2}$, i.e. $p_{\boldsymbol{x}} \in \overline{\mathcal{Z}}_{c_2}$. It follows that

$$\|\boldsymbol{p}_{\boldsymbol{x}} - \boldsymbol{z}_{\boldsymbol{x}}\|_2 \geq \inf_{\boldsymbol{z} \in \mathcal{Z}_{c_2}} \|\boldsymbol{z} - \boldsymbol{z}_{\boldsymbol{x}}\|_2 . \quad (7)$$

Combining Eqn. (6) and Eqn. (7), we see that equality holds in Eqn. (2) for $c = c_2$. Therefore, we conclude that Eqn. (1) tightly lower bounds $D_f(\boldsymbol{z}_{\boldsymbol{x}}, c)$ □

Effectiveness of Distance Measure Our Theorem 3.2 analytically guarantees the effectiveness of our method. In addition, we empirically validate that our estimation method achieves high precision with a relative error of less than 1.5%. See details in Appendix H.

Efficiency of Distance Measure Analytically, Eqn. (1) can be computed in constant time on top of the inference process. Specifically, the numerator in Eqn. (1) calculates the absolute difference between corresponding logits generated during model inference. And the denominator takes a finite number of $|\mathcal{C}| \times (|\mathcal{C}| - 1)$ possible values, which can be pre-computed and retrieved in constant time during inference. Empirically, our method incurs negligible inference

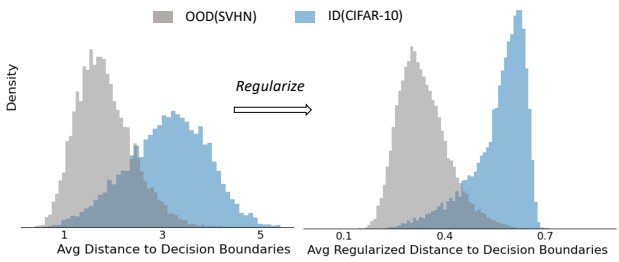

Figure 2. **Regularization enhances ID/OOD separation.** *Left:* Histograms of ID/OOD features based on the average distance to decision boundaries. *Right:* Histograms of ID/OOD features based on the *regularized* average distance to decision boundaries, which effectively compares ID and OOD features at equal deviation levels from the mean of training features.

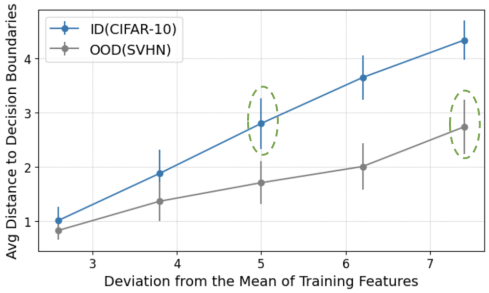

Figure 3. **ID and OOD are better separated at Equal Deviation Levels.** Features are grouped by deviation levels with group mean and variance displayed. Since the average feature distance to decision boundaries increases as features deviate from the mean of training features, the *circled* ID/OOD groups cannot be distinguished based on their average distance to decision boundaries while being effectively separable at their own deviation levels.

overhead. In particular, on a Tesla T4 GPU, the average inference time on the CIFAR-10 classifier is 0.53ms per image *with* or *without* computing the distance using our method. In contrast, the alternative way of estimating the distance through iterative optimization takes 992.2ms under the same setup. This empirically validates the efficiency of our proposed estimation. See details in Appendix H.

For the rest of the paper, we use our closed-form estimation in Eqn. (1) to empirically study the relation between OOD-ness and the feature distance to decision boundaries, and to design our OOD detector.

3.2. Fast Decision Boundary based OOD Detector

We now study OOD detection from the perspective of decision boundaries. Recall that the feature distance to decision boundaries measures the minimum perturbation required to change the classification result. Intuitively, the distance reflects the difficulty of changing the model's decision. Given that a model tends to be more certain on ID samples, we hypothesize that ID features are more likely to reside further away from the decision boundaries compared to OOD features. We extensively validate our hypothesis in Appendix J with plots showing ID/OOD feature distance to decision boundaries. And we spotlight our empirical study by visualizing the per-sample average feature distance to decision boundaries for ID/OOD set in Figure 2 (*Left*).

Going one step further, we investigate the overlapping region of ID/OOD under the metric of the average distance to decision boundaries. To this end, we present Figure 3, where we group ID and OOD samples into buckets based on their deviation levels from the mean of training features. For each group, we plot the mean and variance of the average distance to decision boundaries. Examining Figure 3, we discover that the average feature distance to decision boundaries of both ID and OOD samples increases as features deviate from the mean of training features. We provide theoretical insights into this observation in Section 5. Consequently, OOD samples with a higher deviation level cannot be well distinguished from ID samples that fall into a lower deviation level. In contrast, within the same deviation level, OOD can be much better separated from ID samples.

Based on the understanding, we design our OOD detection score as the average feature distances to decision boundaries, *regularized* by the feature distance to the mean of training features:

$$\texttt{regDistDB} := \frac{1}{|\mathcal{C}|-1} \sum_{c\in\mathcal{C},\ c\neq f(\boldsymbol{x})} \frac{\tilde{D}_f(\boldsymbol{z}_{\boldsymbol{x}}, c)}{\|\boldsymbol{z}_{\boldsymbol{x}} - \boldsymbol{\mu}_{train}\|_2}, \quad (8)$$

where $\tilde{D}_f(\boldsymbol{z}_{\boldsymbol{x}}, c)$ is the estimated distance defined in Eqn. (1) and $\boldsymbol{\mu}_{train}$ denotes the mean of training features. The score approximately compares ID and OOD samples at the same deviation levels. As demonstrated in Figure 2, the regularized distance score enhances the ID/OOD separation, which we explain theoretically in Appendix B. By applying a threshold on `regDistDB`, we introduce the **f**ast **D**ecision **B**oundary based OOD **D**etector (`fDBD`), which identifies samples below the threshold as OOD.

It's worth noticing that our `fDBD` is *hyperparameter-free* and *auxiliary-model-free*. In contrast to many existing approaches (Liang et al., 2018; Lee et al., 2018; Sun et al., 2022), our `fDBD` eliminates the pre-inference cost of tuning hyper-parameter and the potential requirement for additional data. Benefiting from our closed-form distance measuring method, `fDBD` is computationally efficient. Specifically, computing $\tilde{D}_f(\boldsymbol{z}_{\boldsymbol{x}}, c)$ takes constant time (Section 3.1) and computing $\|\boldsymbol{z}_{\boldsymbol{x}} - \boldsymbol{\mu}_{\boldsymbol{G}}\|_2$ in Equation 8 has time complexity $O(P)$, where P is the dimension of penultimate layer. Overall, `fDBD` has time complexity $O(|\mathcal{C}| + P)$, which scales linearly with the number of training classes $|\mathcal{C}|$ and the dimension P, indicating computational scalability for larger datasets and models. We will further demonstrate the efficiency of `fDBD` through experiments in Section 4.

Table 1. **fDBD achieves superior performance with negligible latency overhead on CIFAR-10 OOD benchmarks.** Evaluated on ResNet-18 with FPR95, AUROC, and inference latency. ↑ indicates that larger values are better and vice versa. Best performance highlighted in **bold**. Methods with ∗ are hyperparameter-free.

Method	Latency ↓	SVHN		iSUN		Place365		Texture		AVG	
		FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑
with Cross-entropy Loss											
MSP *	0.53	59.51	91.29	54.57	92.12	62.55	88.63	66.49	88.50	60.78	90.14
ODIN	1.34	61.71	89.12	15.09	97.37	41.45	91.85	52.62	89.41	42.72	91.94
Energy *	0.53	53.96	91.32	27.52	95.59	42.80	91.03	55.23	89.37	44.88	91.83
ViM	0.70	25.38	95.40	30.52	95.10	47.36	90.68	25.69	95.01	32.24	94.05
MDS	2.83	16.77	95.67	7.56	97.93	85.87	68.44	35.21	85.90	36.35	86.99
KNN	1.95	27.85	95.52	24.67	95.52	44.56	90.85	37.57	94.71	33.66	94.15
fDBD *	0.53	22.58	96.07	23.96	95.85	46.59	90.40	31.24	94.48	31.09	94.20
with Supervised Contrastive Loss											
CSI	NA	37.38	94.69	10.36	98.01	38.31	93.04	28.85	94.87	28.73	95.15
SSD+	1.12	1.35	99.72	33.60	95.16	26.09	95.48	12.98	97.70	18.51	97.02
KNN+	1.93	2.20	99.57	20.06	96.74	18.38	96.57	8.09	98.56	12.18	**97.86**
fDBD *	0.55	4.59	99.00	10.04	98.07	23.16	95.09	9.61	98.22	**11.85**	97.60

4. Experiments

In this section, we demonstrate the superior efficiency and effectiveness of `fDBD` across OOD benchmarks. We use two widely recognized metrics in the literature: the False Positive rate at 95% true positive rate (FPR95) and the Area Under the Receiver Operating Characteristic Curve (AUROC). A lower FPR95 score indicates better performance, whereas a higher AUROC value indicates better performance. In addition, we report the per-image inference latency (in milliseconds) evaluated on a Tesla T4 GPU. We refer readers to Appendix F for implementation details.

4.1. Evaluation on CIFAR-10 Benchmarks

In Table 1[1], we present the evaluation of baselines and our `fDBD` across CIFAR-10 OOD benchmarks on ResNet-18.

Training Schemes We evaluate OOD detection performance on a model trained under the standard cross-entropy loss, achieving an accuracy of 94.21%. Moreover, we experiment with a model whose representation mapping is trained using supervised contrastive loss (SupCon) (Khosla et al., 2020). With a linear classifier trained on top of the representation mapping, the model achieves an accuracy of 94.64%. We note that classifiers trained with SupCon loss reach competitive accuracy, making them essential for real-world deployment and highlighting the importance of studying OOD detection performance on such models. As shown by Sun et al. (2022), clustering-based OOD detectors excel for models trained with SupCon loss. Thus, we aim to assess if `fDBD` can achieve state-of-art performance in such competitive scenarios.

[1] `CSI` results copied from Table 4 in Sun et al. (2022).

Datasets On the CIFAR-10 OOD benchmark, we use the standard CIFAR-10 test set with 10,000 images as ID test samples. For OOD samples, we consider common OOD benchmarks: SVHN (Netzer et al., 2011), iSUN (Xu et al., 2015), Places365 (Zhou et al., 2017), and Texture (Cimpoi et al., 2014). All images are of size 32×32.

Baselines We compare our method with six baseline methods on the model trained with standard cross-entropy loss. In particular, MSP (Hendrycks & Gimpel, 2016), ODIN (Liang et al., 2018), and Energy (Liu et al., 2020) design OOD score functions on the model output. Conversely, MDS (Lee et al., 2018) and KNN (Sun et al., 2022) utilize the clustering of ID samples in the feature space and build auxiliary models for OOD detection. ViM (Wang et al., 2022) combines feature null space information with the output space Energy score. In addition, we consider four baseline methods particularly competitive under contrastive loss, `CSI` (Tack et al., 2020), `SSD+` (Sehwag et al., 2020), and `KNN+`. All four methods utilize feature space clustering through building auxiliary models. Our method, `fDBD`, is training-agnostic and applicable across training schemes. We eliminate auxiliary models and incorporate class-specific information from the decision boundaries perspective. Notably, `fDBD`, MSP, and Energy are hyperparameter free, while the other baselines require hyperparameter fine-tuning.

OOD Detection Performance In Table 1, we compare `fDBD` with the baselines. Overall, `fDBD` achieves state-of-art performance in terms of FPR95 and AUROC scores across training schemes. In addition, thanks to our efficient distance estimation method in Section 3.1, `fDBD` has minimal computational overhead: the original classifier takes

Table 2. **fDBD achieves superior performance with negligible latency overhead on ImageNet OOD benchmark.** Evaluated on ResNet-50 with FPR95, AUROC, and inference latency. ↑ indicates that larger values are better and vice versa. Best performance highlighted in **bold**. Methods with ∗ are hyperparameter-free.

Method	Latency↓	iNaturalist		SUN		Places		Texture		Avg	
		FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑
with Cross-entropy Loss											
MSP *	7.04	54.99	87.74	70.83	80.63	73.99	79.76	68.00	79.61	66.95	81.99
ODIN	7.05	47.66	89.66	60.15	84.59	67.90	81.78	50.23	85.62	56.48	85.41
Energy *	7.04	55.72	89.95	59.26	85.89	64.92	82.86	53.72	85.99	58.41	86.17
ViM	9.55	71.85	87.42	81.79	81.07	83.12	78.40	14.88	96.83	62.91	85.93
MDS	35.83	97.00	52.65	98.50	42.41	98.40	41.79	55.80	85.01	87.43	55.17
KNN ($\alpha = 100\%$)	10.31	59.00	86.47	68.82	80.72	76.28	75.76	11.77	97.07	53.97	85.01
KNN ($\alpha = 1\%$)	7.04	59.08	86.20	69.53	80.10	77.09	74.87	11.56	97.18	54.32	84.59
fDBD *	6.81	40.24	93.67	60.60	86.97	66.40	84.27	37.50	92.12	51.19	89.26
with Supervised Contrastive Loss											
SSD+	28.31	57.16	87.77	78.23	73.10	81.19	70.97	36.37	88.53	63.24	80.09
KNN+($\alpha = 100\%$)	10.47	30.18	94.89	48.99	88.63	59.15	84.71	15.55	95.40	38.47	90.91
KNN+($\alpha = 1\%$)	7.04	30.83	94.72	48.91	88.40	60.02	84.62	16.97	94.45	39.18	90.55
fDBD *	6.82	17.27	96.68	42.30	90.90	49.77	88.36	21.83	95.43	**37.79**	**92.84**

0.53 milliseconds per image, and with `fDBD`, the processing time remains the same. Furthermore, we observe that OOD detection significantly improves under contrastive learning. This aligns with the study by Sun et al. (2022), showing that contrastive learning better separates ID and OOD features.

We highlight three groups of comparisons:

- **`fDBD` v.s. MSP / Energy**: All three methods are hyperparameter-free and detect OOD based on model uncertainty: MSP and Energy use softmax confidence and Energy score in the output space, respectively, whereas `fDBD` utilizes the feature-space distance *w.r.t.* decision boundaries. Looking into the performance in Table 1, on the model trained with cross-entropy loss, our `fDBD` reduces the average FPR95 of MSP by 29.69%, which is a relatively **48.85%** reduction in error. Additionally, `fDBD` reduces the average FPR95 of Energy by 13.78%, resulting in a relatively **30.73%** reduction in error. The substantial improvement aligns with our intuition that the feature space contains crucial information for OOD detection, which we leverage in both our uncertainty metric and our regularization scheme.

- **`fDBD` v.s. KNN** We benchmark against KNN under the same hyperparameter setup in Sun et al. (2022), using $k = 50$ nearest neighbors across the entire training set. While both `fDBD` and KNN achieve superior detection effectiveness on CIFAR-10 OOD benchmark, KNN reports an average inference time of $1.93ms$ per image, inducing a noticeable overhead in comparison to `fDBD` due to the use of the auxiliary model. In addition, `fDBD` significantly outperforms KNN on ImageNet OOD benchmark in Table 2, highlighting the benefit of incorporating the class-specific information from the class decision boundary perspective.

- **`fDBD` v.s.ViM** `fDBD` and ViM (Wang et al., 2022) both integrate class-specific information into feature space representation. Specifically, ViM algebraically adds the output space energy score to the feature null space score. Due to the use of null space, ViM requires expensive matrix multiplication during inference, resulting in a noticeable latency increase of $0.70ms$ compared to `fDBD`. Moreover, `fDBD` outperforms ViM, especially on ImageNet OOD benchmark in Table 2. This suggests the effectiveness of our geometrically motivated integration of class-specific information from the perspective of feature-space class decision boundaries, compared to simplly algebraically adding output-space scores to feature-space scores, as done in ViM.

4.2. Evaluation on ImageNet Benchmarks

In Table 2 [2], we further compare the efficiency and effectiveness of our `fDBD` and baselines on larger scale ImageNet OOD Benchmarks on ResNet-50.

Training Schemes & **Datasets** & **Baselines** We consider the training schemes discussed in Section 4.1 and examine models trained with cross-entropy loss and supervised contrastive loss. The ResNet-50 trained under cross-entropy

[2]Results in Table 2 except ours and ViM are from Table 4 by Sun et al. (2022). MDS here refers to Mahalanobis there. Following the reference table, we exclude CSI, since Sun et al. (2022) note that training of CSI on ImageNet is notably resource-intensive, requiring three months on 8 Nvidia 2080Tis.

Table 3. **fDBD achieves competitive performance on ViT-B/16 model fine-tuned on ImageNet-1k.** Evaluated under AUROC. Best performance highlighted in **bold**.

Method	iNaturalist	SUN	Places	Texture	Avg
ViM	98.98	92.13	89.22	92.13	92.77
KNN	98.67	90.42	87.13	90.82	91.76
fDBD	98.76	92.20	89.88	90.71	**92.89**

Table 4. fDBD is compatible with activation shaping algorithms ReAct, ASH, and Scale. Evaluated under AUROC and FPR95 on ImageNet OOD Benchmark. Best performance highlighted in **bold**.

Method	iNaturalist		SUN		Places		Texture		Avg	
	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑
fDBD w/ ReLU	40.24	93.67	60.60	86.97	66.40	84.27	37.50	92.12	51.19	89.26
fDBD w/ ReAct	20.85	96.37	32.37	93.31	41.24	90.78	27.11	94.56	30.39	93.76
fDBD w/ ASH	12.89	97.67	30.28	93.66	42.40	90.53	12.18	97.61	24.44	94.87
fDBD w/ Scale	10.19	98.07	24.58	94.87	36.12	92.00	12.52	97.48	**20.85**	**95.61**

loss achieves an accuracy of 76.65% and the ResNet-50 trained under supervised contrastive loss achieves an accuracy of 77.30%.

We use 50,000 ImageNet validation images in the standard split as ID test samples. Following Huang & Li (2021) and Sun et al. (2022), we remove classes in Texture, Places365 (Zhou et al., 2017), iNaturalist (Van Horn et al., 2018), SUN (Xiao et al., 2010) that overlap with ImageNet and use the remaining datasets as OOD samples. All images are of size 224×224.

We compare to the same baselines in Section 4.1 except for CSI. For KNN, we consider two sets of hyper-parameters reported in the original paper (Sun et al., 2022): $\alpha = 100\%$ refers to searching through all training data for $k = 1000$ nearest neighbors; $\alpha = 1\%$ refers to searching through sampled 1% of training data for 10 nearest neighbors.

OOD Detection Performance Table 2 shows that `fDBD` outperforms all baselines in both average FPR95 and average AUROC on ImageNet OOD benchmarks. This demonstrates `fDBD` consistently maintains its superior effectiveness in OOD detection on large-scale datasets. In addition, `fDBD` remains computationally efficient for ImageNet OOD detection. This aligns with our observation on CIFAR-10 benchmarks and supports our analysis that `fDBD` scales linearly with the class number and the dimension, ensuring manageable computation for large models and datasets.

4.3. Evaluation on Alternative Architectures

To examine the generalizability of our proposed method beyond ResNet, we further experiment with transformer-based ViT model (Dosovitskiy et al., 2020) and DenseNet (Huang et al., 2017). In Table 3, we evaluate our `fDBD`, as well as strong competitors ViM and KNN on a ViT-B/16 fine-tuned with ImageNet-1k using cross-entropy loss. The classifier achieves an accuracy of 81.14%. We consider the same OOD test sets as in Section 4.2 for Imagenet. In Appendix C, we extend our experiments to DenseNet. The performance on ViT and DenseNet demonstrates the effectiveness of `fDBD` across different network architectures.

4.4. Evaluation under Activation Shaping

Orthogonal to the effort of designing standalone detection scores, Sun et al. (2021); Djurisic et al. (2022) and Xu et al. (2023) propose to shape the feature activation to improve ID/OOD separation. The proposed algorithms, ReAct (Sun et al., 2021), ASH (Djurisic et al., 2022), and Scale (Xu et al., 2023), serve as alternative operations to the standard ReLU activation in our experiments so far. With proper hyper-parameter selection, such algorithms have been shown to enhance the performance of standalone scores such as Energy, as detailed in Appendix G. As a hyperparameter-free method, our `fDBD` can be seamlessly combined with hyperparameter-dependent activation shaping algorithms without intricate tuning interactions. In Table 4, we compare `fDBD` performance under standard ReLU activation and under activation shaping algorithms ReAct, ASH, and Scale. Specifically, we evaluate ImageNet OOD Benchmarks on a ResNet-50 trained under cross-entropy loss following detailed setups in Section 4.2. For hyperparameter selection, we adhere to the original papers and set the percentile values to 80, 90, 90 for ReAct, ASH, and Scale, respectively. With activation shaping applied both to test features and the mean of training feature in Equation 8, we observe improved performance across OOD datasets, validating the compatibility of `fDBD` with ReAct, ASH, and Scale. We remark that `fDBD` with Scale achieves the state-of-art performance on this benchmark, comparable to Energy with Scale, as detailed in Appendix D.

Table 5. **Regularization enhances the effectiveness of OOD detection.** AUROC scores reported on ImageNet Benchmarks (higher is better). `regDistDB` outperforms `avgDistDB`.

	iNaturalist	SUN	Places	Texture
avgDistDB	90.51	85.55	83.05	86.79
$\|z_x - \mu_{train}\|_2$	47.84	58.59	58.95	41.92
regDistDB	93.67	86.97	84.27	92.12

4.5. Ablation Study

4.5.1. Effect of Regularization

Previously, we illustrate in Figure 2 that regularization enhances the ID/OOD separation under the metric of feature distances to decision boundaries. We now quantitatively study the regularization effect. Specifically, we compare the performance of OOD detection using the regularized average distance `regDistDB`, the regularization term $\|z - \mu_{train}\|_2$, and the un-regularized average distance

$$\texttt{avgDistDB} \coloneqq \|z - \mu_{train}\|_2 \, \texttt{regDistDB}$$

as detection scores respectively. Experiments are conducted on a ResNet-50 trained under cross-entropy loss following detailed setups in Section 4.2. We report the performance in AUROC scores in Table 5 and FPR95 in Appendix I. Aligning with Figure 3, $\|z - \mu_{train}\|_2$ alone does not necessarily distinguish between ID and OOD samples, as indicated by AUROC scores around 50. However, regularization with respect to $\|z - \mu_{train}\|_2$ enhances ID/OOD separation. Consequently, `regDistDB` improves over `avgDistDB` and achieves higher AUROC, as shown in Table 5. This supports our intuition in Section 3 to compare ID/OOD at equal deviation levels through regulirization. We further theoretically explain the observed enhancement in Appendix B.

4.5.2. Effect of Individual Distances

For `fDBD`, we design the detection score as the feature distances to the decision boundaries, averaged over *all* unpredicted classes. Notably, `fDBD` operates as a hyperparameter-free method, and we do **not** tune the number of distances in our experiments. Nevertheless, we perform an ablation study to understand the effect of individual distances.

To align across samples predicted as different classes, we sort per sample the feature distances to decision boundaries. We then detect OOD using the average of top-k smallest distance values. Specifically, $k = 1$ corresponds to the detection score being the ratio between the feature distance to the closest decision boundary and the feature distance to the mean of training features. And $k = 9$ on CIFAR-10 and $k = 999$ on ImageNet recover our detection score `regDistDB` (see Eqn. (8)), where we average over all distances for OOD detection.

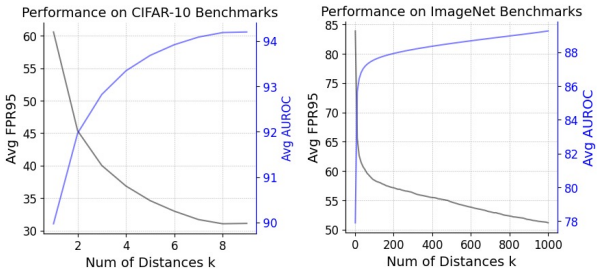

Figure 4. **Ablation on Individual Distances.** *Left:* CIFAR-10 Benchmark performance improves with an increasing number of distances. *Right:* ImageNet Benchmark performance improves with an increasing number of distances. The performance supports the use of all distances in our hyperparameter-free `fDBD`.

We experiment with CIFAR-10 and ImageNet benchmarks on ResNets trained with cross-entropy loss, following the setups in Section 4.1 and Section 4.2. In Figure 4, we present the average FPR95 and AUROC score across OOD datasets, using k distances for detection. Looking into Figure 4, the performance improves with increasing number of k. This justifies our design of `fDBD` as a hyper-parameter-free method, utilizing all distances for OOD detection.

5. Theoretical Analysis

In this section, we give theoretical analysis to shed light on our observation and algorithm design in Section 3.

Setups We consider a general classifier for a class set $\mathcal{C}$ with a penultimate layer of dimension P. Following Lee et al. (2018), we model the ID feature distribution as a Gaussian mixture. Specifically, we consider $|\mathcal{C}|$ equally-weighted components, where each component corresponds to a class $i \in \mathcal{C}$ and follows a Gaussian distribution $N(\boldsymbol{\mu}_i, \sigma^2 \boldsymbol{I})$, where $\boldsymbol{I}$ is the identity matrix. Without loss of generality, we assume the distribution is zero-centered, i.e. $\boldsymbol{\mu} \doteq \frac{1}{|\mathcal{C}|}\sum_{i\in\mathcal{C}} \boldsymbol{\mu}_i = \mathbf{0}$. Following the empirical observation by Papyan et al. (2020), we model the geometry of class means $\{\mu_i\}$ as a simplex Equiangular Tight Framework (ETF):

$$\|\boldsymbol{\mu}_i\|_2 = \|\boldsymbol{\mu}_j\|_2 \;\; \forall i, j,$$

$$\langle \frac{\boldsymbol{\mu}_i}{\|\boldsymbol{\mu}_i\|_2}, \frac{\boldsymbol{\mu}_j}{\|\boldsymbol{\mu}_j\|_2} \rangle = \frac{|\mathcal{C}|}{|\mathcal{C}|-1}\delta_{i,j} - \frac{1}{|\mathcal{C}|-1},$$

where δ_{ij} is the Kronecker delta symbol.

Under the modeling, the optimal decision region of class i can be defined as:

$$\mathcal{V}_i \doteq \{\boldsymbol{z} : \langle \boldsymbol{\mu}_i, \boldsymbol{z} \rangle \geq \max_{j\neq i} \langle \boldsymbol{\mu}_j, \boldsymbol{z} \rangle\}.$$

Correspondingly, the decision boundary between class i and j is:

$$\mathcal{S}_{ij} \doteq \{\boldsymbol{z} : \langle \boldsymbol{\mu}_i, \boldsymbol{z} \rangle = \langle \boldsymbol{\mu}_j, \boldsymbol{z} \rangle \geq \max_{k\neq i,j} \langle \boldsymbol{\mu}_k, \boldsymbol{z} \rangle\}.$$

For any $\boldsymbol{z} \in \mathcal{V}_i$, the distance from $\boldsymbol{z}$ to the decision boundary between class i and j, $\mathcal{S}_{ij}$, is the length of the projection of $\boldsymbol{z}$ onto the norm vector of $\mathcal{S}_{ij}$:

$$d(\boldsymbol{z}, \mathcal{S}_{ij}) \doteq \frac{\langle \boldsymbol{z}, \boldsymbol{\mu}_i - \boldsymbol{\mu}_j \rangle}{\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|}.$$

For simplicity of notation, we denote the union of decision boundaries as $\mathcal{S} = \cup \mathcal{S}_{ij}$. Additionally, we denote the distance from $\boldsymbol{z}$ to its closest decision boundary as $d(\boldsymbol{z}, \mathcal{S})$.

Following Sun et al. (2022), we assume that OOD features reside outside the dense region of ID feature distribution. We define this dense region as the area within two standard deviations from each class mean:

$$\mathcal{I} \doteq \cup_i \mathcal{I}_i = \cup_i \{\boldsymbol{z} : \|\boldsymbol{z} - \boldsymbol{\mu}_i\| \leq 2\sigma\}.$$

We also assume that ID features are well-separated, so that the dense region of each class is entirely within its decision region, i.e., $\mathcal{I}_i \subset \mathcal{V}_i$.

Main Result Recall from Section 3 that we observe the feature distance to decision boundaries increases as features deviate from the mean of training features $\boldsymbol{\mu}_{train}$. This observation motivates us to compare ID and OOD features at equal deviation levels and design our detection algorithm accordingly. Note that $\boldsymbol{\mu}_{train}$ is an empirical estimation of $\boldsymbol{\mu}$, the mean of ID feature distribution.

To further understand our observation, we present Proposition 5.1, which demonstrates that the feature distance to the decision boundary $d(\mathcal{S}, z)$ increases as z deviates from $\boldsymbol{\mu}$. Additionally, we validate our detection algorithm in Proposition 5.2, showing that, at equal deviation levels, ID features tend to be further from the decision boundary compared to OOD features. We present the complete proofs in Appendix A.

As discussed in Setups, we assume without loss of generality that the features are zero-centered, i.e., $\boldsymbol{\mu} = \mathbf{0}$.

Proposition 5.1. *Consider the set of features of equal distance r to the ID distribution mean $\mathcal{E}_r \doteq \{\boldsymbol{z} : \|\boldsymbol{z} - \boldsymbol{\mu}\| = \|\boldsymbol{z}\| = r\}$. For any $r_0 < r_1$, we have:*

$$\begin{aligned} &\frac{1}{Vol(\mathcal{E}_{r_0})} \int_{z \in \mathcal{E}_{r_0}} d(z, \mathcal{S})\, \mathrm{d(z)} \\ &< \frac{1}{Vol(\mathcal{E}_{r_1})} \int_{z \in \mathcal{E}_{r_1}} d(z, \mathcal{S})\, \mathrm{d(z)}. \end{aligned} \tag{9}$$

Proposition 5.2. *Consider ID and OOD features of equal distance r to the ID distribution mean, where $\sigma < r < 5\sigma$. For ID region, $\mathcal{I} \cap \mathcal{E}_r$, and OOD region, $\mathcal{I}^{\complement} \cap \mathcal{E}_r$, we have*

$$\begin{aligned} &\frac{1}{Vol(\mathcal{I} \cap \mathcal{E}_r)} \int_{z \in \mathcal{I} \cap \mathcal{E}_r} d(z, \mathcal{S})\, \mathrm{d(z)} \\ &> \frac{1}{Vol(\mathcal{I}^{\complement} \cap \mathcal{E}_r)} \int_{z \in \mathcal{I}^{\complement} \cap \mathcal{E}_r} d(z, \mathcal{S})\, \mathrm{d(z)}. \end{aligned} \tag{10}$$

6. Related Work

An extensive body of research work has been focused on developing OOD detection algorithms. And we refer readers to comprehensive literature reviews by Yang et al. (2021b; 2022a;b); Zhang et al. (2023); Bitterwolf et al. (2023). Particularly, one line of work is post-hoc and builds upon pre-trained models. For example, Liang et al. (2018); Hendrycks et al. (2019) and Liu et al. (2020) design OOD score over the output space of a classifier, whereas Lee et al. (2018); Sun et al. (2022); Ndiour et al. (2020) and Liu & Qin (2023) measure OOD-ness using *feature* space information. Moreover, Huang et al. (2021) explore OOD detection from the gradient space. Our work builds on the *feature* space and investigates from the largely under-explored perspective of decision boundaries. Orthogonality, Sun et al. (2021); Djurisic et al. (2022) and Xu et al. (2023) reveal that activation shaping on pre-trained models can enhance the ID/OOD separation and improves the performance of standalone detection scores in general. Our experiments validates that `fDBD` is also compatible with activation shaping methods.

Another line of work explores the regularization of OOD detection in training. For example, DeVries & Taylor (2018) and Hsu et al. (2020) propose OOD-specific architecture whereas Wei et al. (2022); Huang & Li (2021) and Ming et al. (2023) design OOD-specific training loss. In addition, Tack et al. (2020) propose an OOD-specific contrastive learning scheme, while Tao et al. (2023) and Du et al. (2022) explore methods for constructing virtual OOD samples to facilitate OOD-aware training. Recently, Fort et al. (2021) reveal that finetuning a visual transformer with OOD exposure significantly can improve OOD detection performance. Our work does not assume specific training schemes and does not belong to this school of work.

7. Conclusion

In this work, we propose an efficient and effective OOD detector `fDBD` based on the novel perspective of feature distances to decision boundaries. We first introduce a closed-form estimation to measure the feature distance to decision boundaries. Based on our estimation method, we reveal that ID samples tend to reside further away from the decision boundary than OOD samples. Moreover, we find that ID and OOD samples are better separated when compared at equal deviation levels from the mean of training features. By regularizing feature distances to decision boundaries based on feature deviation from the mean, we design a decision boundary-based OOD detector that achieves state-of-the-art effectiveness with minimal latency overhead. We hope our algorithm can inspire future work to explore model uncertainty from the perspective of decision boundaries, both for OOD detection and other research problems such as adversarial robustness and domain generalization.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Bitterwolf, J., Mueller, M., and Hein, M. In or out? fixing imagenet out-of-distribution detection evaluation. In *ICML*, 2023. URL https://proceedings.mlr.press/v202/bitterwolf23a.html.

Carlini, N. and Wagner, D. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pp. 39–57. Ieee, 2017.

Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., and Vedaldi, A. Describing textures in the wild. In *IEEE Conference in Computer Vision and Pattern Recognition*, pp. 3606–3613, 2014.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

DeVries, T. and Taylor, G. W. Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint arXiv:1802.04865*, 2018.

Djurisic, A., Bozanic, N., Ashok, A., and Liu, R. Extremely simple activation shaping for out-of-distribution detection. *arXiv preprint arXiv:2209.09858*, 2022.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Du, X., Wang, Z., Cai, M., and Li, Y. Vos: Learning what you don't know by virtual outlier synthesis. *arXiv preprint arXiv:2202.01197*, 2022.

Fort, S., Ren, J., and Lakshminarayanan, B. Exploring the limits of out-of-distribution detection. *Advances in Neural Information Processing Systems*, 34:7068–7081, 2021.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hendrycks, D. and Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

Hendrycks, D. and Gimpel, K. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

Hendrycks, D., Basart, S., Mazeika, M., Mostajabi, M., Steinhardt, J., and Song, D. Scaling out-of-distribution detection for real-world settings. *arXiv preprint arXiv:1911.11132*, 2019.

Hsu, Y.-C., Shen, Y., Jin, H., and Kira, Z. Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10951–10960, 2020.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

Huang, R. and Li, Y. Mos: Towards scaling out-of-distribution detection for large semantic space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8710–8719, 2021.

Huang, R., Geng, A., and Li, Y. On the importance of gradients for detecting distributional shifts in the wild. *Advances in Neural Information Processing Systems*, 34: 677–689, 2021.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., and Krishnan, D. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020.

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

Lee, K., Lee, K., Lee, H., and Shin, J. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31, 2018.

Liang, S., Li, Y., and Srikant, R. Enhancing the reliability of out-of-distribution image detection in neural networks. In *6th International Conference on Learning Representations, ICLR 2018*, 2018.

Liu, L. and Qin, Y. Detecting out-of-distribution through the lens of neural collapse. *arXiv preprint arXiv:2311.01479*, 2023.

Liu, W., Wang, X., Owens, J., and Li, Y. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems*, 33:21464–21475, 2020.

Loshchilov, I. and Hutter, F. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Ming, Y., Sun, Y., Dia, O., and Li, Y. How to exploit hyperspherical embeddings for out-of-distribution detection? In *The Eleventh International Conference on Learning Representations*, 2023.

Ndiour, I., Ahuja, N., and Tickoo, O. Out-of-distribution detection with subspace techniques and probabilistic modeling of features. *arXiv preprint arXiv:2012.04250*, 2020.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. 2011.

Papernot, N., Faghri, F., Carlini, N., Goodfellow, I., Feinman, R., Kurakin, A., Xie, C., Sharma, Y., Brown, T., Roy, A., Matyasko, A., Behzadan, V., Hambardzumyan, K., Zhang, Z., Juang, Y.-L., Li, Z., Sheatsley, R., Garg, A., Uesato, J., Gierke, W., Dong, Y., Berthelot, D., Hendricks, P., Rauber, J., and Long, R. Technical report on the cleverhans v2.1.0 adversarial examples library. *arXiv preprint arXiv:1610.00768*, 2018.

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

Sastry, C. S. and Oore, S. Detecting out-of-distribution examples with gram matrices. In *International Conference on Machine Learning*, pp. 8491–8501. PMLR, 2020.

Sehwag, V., Chiang, M., and Mittal, P. Ssd: A unified framework for self-supervised outlier detection. In *International Conference on Learning Representations*, 2020.

Sun, Y., Guo, C., and Li, Y. React: Out-of-distribution detection with rectified activations. *Advances in Neural Information Processing Systems*, 34:144–157, 2021.

Sun, Y., Ming, Y., Zhu, X., and Li, Y. Out-of-distribution detection with deep nearest neighbors. *arXiv preprint arXiv:2204.06507*, 2022.

Tack, J., Mo, S., Jeong, J., and Shin, J. Csi: Novelty detection via contrastive learning on distributionally shifted instances. *Advances in neural information processing systems*, 33:11839–11852, 2020.

Tao, L., Du, X., Zhu, X., and Li, Y. Non-parametric outlier synthesis. *arXiv preprint arXiv:2303.02966*, 2023.

Van Horn, G., Mac Aodha, O., Song, Y., Cui, Y., Sun, C., Shepard, A., Adam, H., Perona, P., and Belongie, S. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8769–8778, 2018.

Wang, H., Li, Z., Feng, L., and Zhang, W. Vim: Out-of-distribution with virtual-logit matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4921–4930, 2022.

Wei, H., Xie, R., Cheng, H., Feng, L., An, B., and Li, Y. Mitigating neural network overconfidence with logit normalization. *arXiv preprint arXiv:2205.09310*, 2022.

Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., and Torralba, A. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pp. 3485–3492. IEEE, 2010.

Xu, K., Chen, R., Franchi, G., and Yao, A. Scaling for training time and post-hoc out-of-distribution detection enhancement. In *The Twelfth International Conference on Learning Representations*, 2023.

Xu, P., Ehinger, K. A., Zhang, Y., Finkelstein, A., Kulkarni, S. R., and Xiao, J. Turkergaze: Crowdsourcing saliency with webcam based eye tracking. *arXiv preprint arXiv:1504.06755*, 2015.

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

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

Yang, J., Wang, P., Zou, D., Zhou, Z., Ding, K., Peng, W., Wang, H., Chen, G., Li, B., Sun, Y., Du, X., Zhou, K., Zhang, W., Hendrycks, D., Li, Y., and Liu, Z. Openood: Benchmarking generalized out-of-distribution detection. 2022a.

Yang, J., Zhou, K., and Liu, Z. Full-spectrum out-of-distribution detection. *arXiv preprint arXiv:2204.05306*, 2022b.

Zhang, J., Yang, J., Wang, P., Wang, H., Lin, Y., Zhang, H., Sun, Y., Du, X., Zhou, K., Zhang, W., Li, Y., Liu, Z., Chen, Y., and Li, H. Openood v1.5: Enhanced benchmark for out-of-distribution detection. *arXiv preprint arXiv:2306.09301*, 2023.

Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., and Torralba, A. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017.

A. Proof for Section 5

Under the setups in Section 5, we slice the geometric space into regions $\mathcal{V}_i^j$, $i, j \in \{1, ..., C\}$, defined by

$$\mathcal{V}_i^j \doteq \{\boldsymbol{z} : \langle \boldsymbol{z}, \boldsymbol{\mu}_i \rangle > \langle \boldsymbol{z}, \boldsymbol{\mu}_j \rangle \geq \max_{k \neq i,j} \langle \boldsymbol{z}, \boldsymbol{\mu}_k \rangle\}.$$

Geometrically, $\mathcal{V}_i^j$ represents the region within the decision region $\mathcal{V}_i$ of class i where the second most likely class is j. For any $\boldsymbol{z} \in \mathcal{V}_i^j$, we have $d(\boldsymbol{z}, \mathcal{S}) = d(\boldsymbol{z}, \mathcal{S}_{ij})$. In the following, we establish Proposition 5.1 and Proposition 5.2 in region $\mathcal{V}_i^j$ for any i, j, thereby confirming their validity in the entire region thanks to symmetry.

Proof of Proposition 5.1

Proof. By definition, any $\boldsymbol{z}_0 \in \mathcal{E}_{r_0}$ satisfies $\|\boldsymbol{z}_0\| = r_0$. Scaling $\boldsymbol{z}_0$ by r_1/r_0 yields $\boldsymbol{z}_1 = r_1/r_0 \cdot \boldsymbol{z}_0$. We have $\|\boldsymbol{z}_1\| = r_1/r_0 \cdot \|\boldsymbol{z}\|_0 = r_1$, indicating that $\boldsymbol{z}_1$ is an element of $\mathcal{E}_{r_1}$. Conversely, for any $\boldsymbol{z}_1 \in \mathcal{E}_{r_1}$, we can obtain $\boldsymbol{z}_0 = (r_0/r_1) \cdot \boldsymbol{z}_1 \in \mathcal{E}_{r_0}$. This establishes a one-to-one mapping between elements in $\mathcal{E}_{r_0}$ and $\mathcal{E}_{r_1}$. Considering any pair $(\boldsymbol{z}_0, \boldsymbol{z}_1)$, we have

$$d(\boldsymbol{z}_0, \mathcal{S}) = d(\boldsymbol{z}_0, \mathcal{S}_{ij}) = \frac{\langle \boldsymbol{z}_0, \boldsymbol{\mu}_i - \boldsymbol{\mu}_j \rangle}{\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|} = r_0/r_1 \cdot \frac{\langle \boldsymbol{z}_1, \boldsymbol{\mu}_i - \boldsymbol{\mu}_j \rangle}{\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|} \quad < \frac{\langle \boldsymbol{z}_1, \boldsymbol{\mu}_i - \boldsymbol{\mu}_j \rangle}{\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|} = d(\boldsymbol{z}_1, \mathcal{S}_{ij}) = d(\boldsymbol{z}_1, \mathcal{S}), \tag{11}$$

indicating a consistent relative ordering between elements in $\mathcal{E}_{r_0}$ and $\mathcal{E}_{r_1}$ Therefore, Proposition 5.1, which asserts the ordering of the mean between these two sets, is validated. □

Proof of Proposition 5.2

Proof. Without loss of generality, we assume that $\|\boldsymbol{\mu}_i\| = 1$ for $\forall i \in \mathcal{C}$ and the distance $r = 1$. To parameterize the element $\boldsymbol{z}$ within region $\mathcal{V}_i^j \cap \mathcal{E}_{r=1}$ for given i, j, we consider the geodesic on sphere $\mathcal{E}_{r=1}$ that extends from the class mean $\boldsymbol{\mu}_i$ to element $\boldsymbol{z}$, and further extends to point $\boldsymbol{v} \in \mathcal{S}_{ij} \cap \mathcal{E}_{r=1}$:

$$\gamma_v(t) = \cos(t)\boldsymbol{\mu}_i + \sin(t)\frac{\boldsymbol{v} - \langle \boldsymbol{v}, \boldsymbol{\mu}_i \rangle \boldsymbol{\mu}_i}{\|\boldsymbol{v} - \langle \boldsymbol{v}, \boldsymbol{\mu}_i \rangle \boldsymbol{\mu}_i\|}.$$

For any $\boldsymbol{z} \in \mathcal{V}_i^j \cap \mathcal{E}_{r=1}$ and its corresponding $\boldsymbol{v}$, we have $\boldsymbol{z}$ residing on the geodesic $\gamma_v(t)$ with $t = \arccos \langle \boldsymbol{z}, \boldsymbol{\mu}_i \rangle$.

Geometrically, along a geodesic $\gamma_v(t)$, the parameter t increases as one moves from the ID region $\mathcal{I} \cap \mathcal{E}_{r=1}$ to the OOD region $\mathcal{I}^{\complement} \cap \mathcal{E}_{r=1}$. Moreover, $d(\gamma_v(t), \mathcal{S})$ is equivalent to $d(\gamma_v(t), \mathcal{S}_{ij})$ given that the geodesic resides within $\mathcal{V}_i^j$. Therefore, to show Proposition 5.2 holds for $\forall \boldsymbol{z} \in \mathcal{V}_i^j \cap \mathcal{E}_{r=1}$, it suffices to show that the function $d(\gamma_v(t), \mathcal{S}_{ij})$ decreases with t. Diving into the derivatives of $d(\gamma_v(t), \mathcal{S}_{ij})$ with respect to t, we have:

$$\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\| \frac{d}{dt} d(\gamma_v(t), \mathcal{S}_{ij}) = \langle \gamma_v'(t), \boldsymbol{\mu}_i - \boldsymbol{\mu}_j \rangle = \langle -\sin(t)\boldsymbol{\mu}_i + \cos(t)\frac{\boldsymbol{v} - \langle \boldsymbol{v}, \boldsymbol{\mu}_i \rangle \boldsymbol{\mu}_i}{\|\boldsymbol{v} - \langle \boldsymbol{v}, \boldsymbol{\mu}_i \rangle \boldsymbol{\mu}_i\|}, \boldsymbol{\mu}_i - \boldsymbol{\mu}_j \rangle \tag{12}$$

$$= -\sin(t) + \frac{\sin(t)}{1 - C} + \frac{\cos(t)}{\|\boldsymbol{v} - \langle \boldsymbol{v}, \boldsymbol{\mu}_i \rangle \boldsymbol{\mu}_i\|} \cdot (\langle \boldsymbol{v}, \boldsymbol{\mu}_i \rangle - \langle \boldsymbol{v}, \boldsymbol{\mu}_j \rangle - \langle \boldsymbol{v}, \boldsymbol{\mu}_i \rangle + \frac{\langle \boldsymbol{v}, \boldsymbol{\mu}_i \rangle}{1 - C}) \tag{13}$$

$$= \frac{C}{1 - C}(\sin(t) + \frac{1}{\|\boldsymbol{v} - \langle \boldsymbol{v}, \boldsymbol{\mu}_i \rangle \boldsymbol{\mu}_i\|} \cos(t) \langle \boldsymbol{v}, \boldsymbol{\mu}_j \rangle). \tag{14}$$

We remark that Eqn. 14 remains negative within the feasible range of parameter t, where $\sin(t) > 0$ and $\cos(t) > 0$. This is because the parameter t has its minimum at $\boldsymbol{\mu}_i$ with $t_{min} = 0$ and reaches max at $\boldsymbol{v}$ with $t_{max} = \arccos(\langle \boldsymbol{v}, \boldsymbol{\mu}_i \rangle)$. As $\langle \boldsymbol{v}, \boldsymbol{\mu}_i \rangle > 0$ from the definition of $\mathcal{V}_i^j$, we have $t_{max} < \frac{\pi}{2}$, ensuring that t remains within the interval $t \in (0, \frac{\pi}{2})$. □

B. Theoretical Justification for Performance Enhancement through Regularization

In the following, x denotes the feature distance to the training feature mean, and y denotes the feature distance to decision boundaries. f_{xy} and g_{xy} denote the joint probability density functions of x and y for ID and OOD samples, respectively. The notation in this section may vary from the rest of the paper for clarity and ease of presentation within this context. Please refer to the corresponding sections for consistent notation throughout the paper.

In Section 3.2, we regularize y with respect to x to compare the distance of ID/OOD features to decision boundaries at the same deviation levels from the training feature mean. Eqn. 11 provides intuition on how our regularization effectively

Table 6. **fDBD achieves superior performance with negligible latency overhead on DenseNet.** Evaluated with FPR95, AUROC, and inference latency. ↑ indicates that larger values are better and vice versa. Best performance highlighted in **bold**.

Method	Latency ↓	SVHN		iSUN		Place365		Texture		AVG	
		FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑	FPR95↓	AUROC↑
MSP	0.87	47.24	93.48	42.31	94.52	63.02	88.57	64.15	88.15	54.18	91.18
ODIN	3.04	25.29	94.57	3.98	98.90	52.85	88.55	57.50	82.38	34.91	91.1
Energy	0.90	40.61	93.99	10.07	98.07	39.40	91.64	56.12	86.43	36.55	92.53
ViM	0.95	20.87	96.44	7.73	98.54	56.97	89.09	22.18	95.58	26.94	94.91
MDS	7.55	6.42	98.31	9.78	97.25	85.14	63.15	21.51	92.15	30.71	87.72
KNN	1.86	3.96	99.29	9.54	98.27	39.96	92.24	19.52	96.38	**18.25**	**96.55**
fDBD	0.88	5.89	98.67	5.90	98.75	39.52	91.53	22.75	95.81	18.52	96.19

enables comparison at the same deviation level x, as y scales linearly with x under our modeling. Thus, the regularization effectively conditions y on x.

In Proposition B.1 below, we analytically justify why conditioning enhances ID/OOD separation, thereby explaining the regularization-induced enhancement observed in Section 3.2 and Section 4.5. Specifically, as Figure 3 (Section 3.2) and Table 5 (Section 4.5) show the ID and OOD samples cannot be distinguished by x alone, we consider the case where the marginal distribution of x is the same for ID and OOD, i.e., $f_x = g_x$.

Proposition B.1. *Under Kullback–Leibler (KL) divergence D_{KL}, we have:*

$$D_{KL}(f_y||g_y) \leq D_{KL}(f_{y|x}||g_{y|x}).$$

Here, f_y and g_y denote the marginal distribution of feature distance to decision boundaries for ID and OOD samples respectively, whereas $f_{y|x}$ and $g_{y|x}$ denote the conditional distribution w.r.t. *feature deviation level from the training feature mean for ID and OOD samples respectively.*

Proof. Following the chain rule of KL divergence, we have

$$D_{KL}(f_{xy}||g_{xy}) = D_{KL}(f_x||g_x) + D_{KL}(f_{y|x}||g_{y|x}).$$

Symmetrically, we also have:

$$D_{KL}(f_{xy}||g_{xy}) = D_{KL}(f_y||g_y) + D_{KL}(f_{x|y}||g_{x|y}).$$

Combining both, we have:

$$D_{KL}(f_y||g_y) = D_{KL}(f_x||g_x) + D_{KL}(f_{y|x}||g_{y|x}) - D_{KL}(f_{x|y}||g_{x|y}).$$

Remind that $D_{KL}(f_x||g_x) = 0$, as $f_x = g_x$. Also, $D_{KL}(f_{x|y}||g_{x|y}) \geq 0$ due to the non-negativity of KL divergence. Therefore, we have:

$$D_{KL}(f_y||g_y) \leq D_{KL}(f_{y|x}||g_{y|x}).$$

□

C. Evaluation on DenseNet

We now extend our evaluation to DenseNet (Huang et al., 2017). The CIFAR-10 classifier we evaluated with achieves a classification accuracy of 94.53%. We consider the same OOD test sets as in Section 4.1. The performance shown in Table 6 further indicates the effectiveness and efficiency of our proposed detector across different network architectures.

D. Evaluation under Activation Shaping

In Table 7, we compare the performance of `fDBD` and Energy under activation shaping methods ReAct, ASH, and Scale. For both `fDBD` and Energy, we follow the original paper and set the value of the percentile hyperparameter to 80, 90, 90 for ReAct, ASH, and Scale, respectively. Experiments are on an ImageNet ResNet-50 classifiers following the detailed setups in Section 4.2. Looking into Table 7, we observe that `fDBD` with Scale achieves state-of-art performance on this benchmark, comparable to Energy with Scale.

Table 7. fDBD is competitive compared to Energy under activation shaping algorithms ReAct, ASH, and Scale on ImageNet Benchmark.

Method	iNaturalist		SUN		Places		Texture		Avg	
	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑	FPR95 ↓	AUROC ↑
Energy w/ ReAct	20.38	96.22	24.20	94.20	33.85	91.58	47.30	89.80	31.43	92.95
fDBD w/ ReAct	20.85	96.37	32.37	93.31	41.24	90.78	27.11	94.56	30.39	93.76
Energy w/ ASH	11.49	97.87	27.98	94.02	39.78	90.98	11.93	97.60	22.80	95.12
fDBD w/ ASH	12.89	97.67	30.28	93.66	42.4	90.53	12.18	97.61	24.44	94.87
Energy w/ Scale	9.48	98.17	23.22	95.02	34.50	92.26	12.89	97.37	20.02	95.70
fDBD w/ Scale	10.19	98.07	24.58	94.87	36.12	92.00	12.52	97.48	20.85	95.61

E. Evaluation under Domain Shift

Our `fDBD`, as a detector for semantic shift induced by mismatch in training/test class types, remains effective when ID samples undergo moderate domain shift in real life. In Table 8, we compare fDBD performance with clean and moderately corrupted ID samples on CIFAR-10 benchmarks. Specifically, we consider CIFAR-10-C (Hendrycks & Dietterich, 2019) with severity level 1 & 2. For each severity level, we construct an aggregated dataset by sampling in total 10,000 images from all 4 classes of corruption: Noise, Blur, Weather, and Digital. For the rest of the setups, we follow Section 4.1 and report the average AUROC across OOD datasets. As shown in Table 8, fDBD's performance degrades slightly as the corruption level increases. Nevertheless, fDBD remains highly effective within a moderate range of domain shift.

Table 8. Performance of fDBD with CIFAR-10 / CIFAR-10-C as ID samples on CIFAR-10 Benchmark.

ID Distribution	Avg AUROC
CIFAR-10	94.20
CIFAR-10-C (Severity Level 1)	91.91
CIFAR-10-C (Severity Level 2)	90.86

F. Implementation Details

F.1. CIFAR-10

ResNet-18 w/ Cross Entropy Loss For experiments presented in Figure 1 *Right*, Figure 2, Figure 3, Figure 4 *Left*, Table 8 and part of Table 1, we evaluate on a CIFAR-10 classifier of ResNet-18 backbone trained with cross entropy loss. The classifier is trained for 100 epochs, with a start learning rate 0.1 decaying to 0.01, 0.001, and 0.0001 at epochs 50, 75, and 90 respectively.

ResNet-18 w/ Contrastive Loss For part of Table 1, we experiment with a CIFAR-10 classifier of the ResNet-18 backbone trained with supcon loss. Following Khosla et al. (2020), the model is trained with for 500 epochs with batch size 1024. The temperature is set to 0.1. The cosine learning rate (Loshchilov & Hutter, 2016) starts at 0.5 is used.

DenseNet-101 w/ Cross Entropy Loss For experiments presented in Table 6, we evaluate on a CIFAR-10 classifier of DenseNet-101 backbone. The classifier is trained following the set up in (Huang et al., 2017) with depth $L = 100$ and growth rate $k = 12$.

F.2. ImageNet

ResNet-50 w/ Cross-Entropy Loss For evaluation on ImageNet in Figure 4 *Right*, part of Table 2, Table 4, Table 5, and Table 7 we use the default ResNet-50 model trained with cross-entropy loss provided by Pytorch. See training recipe here: `https://pytorch.org/blog/how-to-train-state-of-the-art-models-using-torchvision-latest-primitives/`.

ResNet-50 w/ Supervised Contrastive Loss For part of Table 2, we experiment with a ImageNet classifier of ResNet-50 backbone trained with supcon loss. Following Khosla et al. (2020), the model is trained with for 700 epochs with batch size 1024. The temperature is set to 0.1. The cosine learning rate (Loshchilov & Hutter, 2016) starts at 0.5 is used.

ImageNet ViT In Table 3, we evaluate on the pytorch implementation of ViT and the default checkpoint, available https://github.com/lukemelas/PyTorch-Pretrained-ViT/tree/master.

G. Baseline Methods

We provide an overview of our baseline methods in this session. We follow our notation in Section 3. In the following, a lower detection score indicates OOD-ness.

MSP Hendrycks & Gimpel (2016) propose to detect OOD based on the maximum softmax probability. Given a test sample $\boldsymbol{x}$, the detection score of MSP can be represented as:

$$\frac{\exp\left(\boldsymbol{w}_{f(\boldsymbol{x})}^T \boldsymbol{z}_{\boldsymbol{x}} + b_{f(\boldsymbol{x})}\right)}{\sum_{c\in\mathcal{C}} \exp\left(\boldsymbol{w}_c^T \boldsymbol{z}_{\boldsymbol{x}} + b_c\right)}, \tag{15}$$

where $\boldsymbol{z}_{\boldsymbol{x}}$ is the penultimate feature space embedding of $\boldsymbol{x}$. Note that calculating the denominator of the softmax score function is an $\Omega(|\mathcal{C}|T(\texttt{exp}))$ operation, where $T(\texttt{exp})$ is the computational complexity for evaluating the exponential function, which is precision related and non-constant. Note that the on-device implementation of exponential functions often requires huge look-up tables, incurring significant delay and storage overhead. Overall, the computational complexity of MSP on top of the inference process is $\Omega(|\mathcal{C}|T(\texttt{exp}))$.

ODIN Liang et al. (2018) propose to amplify ID and OOD separation on top of MSP through temperature scaling and adversarial perturbation. Given a sample $\boldsymbol{x}$, ODIN constructs a noisy sample $\boldsymbol{x'}$ from $\boldsymbol{x}$ following

$$\boldsymbol{x'} = \boldsymbol{x} - \epsilon \text{sign} \nabla_{\boldsymbol{x}} \frac{\exp\left(\boldsymbol{w}_{f(\boldsymbol{x})}^T \boldsymbol{z}_{\boldsymbol{x}} + b_{f(\boldsymbol{x})}\right)}{\sum_{c\in\mathcal{C}} \exp\left(\boldsymbol{w}_c^T \boldsymbol{z}_{\boldsymbol{x}} + b_c\right)}. \tag{16}$$

Denote the penultimate layer feature of the noisy sample $\boldsymbol{x'}$ as $\boldsymbol{h'}$, ODIN assigns OOD score following:

$$\frac{\exp\left((\boldsymbol{w}_c^T \boldsymbol{h'} + b_c)/T\right)}{\sum_{c'\in\mathcal{C}} \exp\left((\boldsymbol{w}_c'^T \boldsymbol{h'} + b_{c'})/T\right)}, \tag{17}$$

where c is the predicted class for the perturbed sample and T is the temperature. ODIN is a hyperparameter-dependent algorithm and requires additional computation and dataset for hyper-parameter tuning. In our implementation, we set the noise magnitude as 0.0014 and the temperature as 1000.

The computational complexity of ODIN is architecture-dependent. This is because the step of constructing the adversarial example requires back-propagation through the NN, whereas the step of evaluating the softmax score from the adversarial example requires an additional forward pass. Both steps require accessing the whole NN, which incurs significantly higher computational cost than our `fDBD` which only requires accessing the penultimate NN layer.

Energy Liu et al. (2020) design an energy-based score function over the logit output. Given a test sample $\boldsymbol{x}$, the energy based detection score can be represented as:

$$-\log \sum_{c\in\mathcal{C}} \exp\left(\boldsymbol{w}_c^T \boldsymbol{z}_{\boldsymbol{x}} + b_c\right), \tag{18}$$

where $\boldsymbol{z}_{\boldsymbol{x}}$ is the penultimate layer embedding of $\boldsymbol{x}$. The computational complexity of Energy on top of the inference process is $\Omega(|\mathcal{C}|T(\texttt{exp}) + T(\texttt{log}))$, whereas $T(\texttt{exp})$ and $T(\texttt{log})$ are the computational complexity functions for evaluating the exponential and logarithm functions respectively. Note that the on-device implementation of exponential functions and the logarithm functions often requires huge look-up tables, incurring significant delay and storage overhead.

ReAct Sun et al. (2021) build upon the energy score proposed by Liu et al. (2020) and regularizes the score by truncating the penultimate layer estimation. We set the truncation threshold at 90 percentile in our experiments.

ASH Djurisic et al. (2022) build upon the energy score proposed by Liu et al. (2020). Prior to computing the Energy score, ASH sorts each feature to find the top-k elements, scales the top-k elements, and sets the rest to zero. We note that in addition to the cost of Energy, ASH introduces a sorting cost of $O(P \log k)$, where P is the penultimate layer dimension.

Scale Xu et al. (2023) build upon the energy score proposed by Liu et al. (2020). Prior to the Energy score, Scale sorts each feature to find the top-k elements. Based on the ratio between the sum of top-k elements and the sum of all elements, Xu et al. (2023) scale all elements in the feature. We note that in addition to the cost of Energy, Scale also introduces a sorting cost of $O(P \log k)$, where P is the penultimate layer dimension.

MDS On the feature space, Lee et al. (2018) model the ID feature distribution as multivariate Gaussian and designs a Mahalanobis distance-based score:

$$\max_c -(\boldsymbol{e_x} - \hat{\boldsymbol{\mu}}_c)^T \hat{\Sigma}^{-1} (\boldsymbol{e_x} - \hat{\boldsymbol{\mu}}_c), \tag{19}$$

where $\boldsymbol{e_x}$ is the feature embedding of $\boldsymbol{x}$ in a specific layer, $\hat{\mu}_c$ is the feature mean for class c estimated on the training set, and $\hat{\Sigma}$ is the covariance matrix estimated over all classes on the training set. Computing Eqn. (19) requires inverting the covariance matrix $\hat{\Sigma}$ prior to inference, which can be computationally expensive in high dimensions. During inference, computing Eqn. (19) for each sample takes $O(|\mathcal{C}|P^2)$, where P is the dimension of the feature space. This indicates that the computational cost of MDS significantly grows for large-scale OOD detection.

On top of the basic score, Lee et al. (2018) also propose two techniques to enhance the OOD detection performance. The first is to inject noise into samples. The second is to learn a logistic regressor to combine scores across layers. We tune the noise magnitude and learn the logistic regressor on an adversarial constructed OOD dataset, which incurs additional computational overhead. The selected noise magnitude in our experiments is 0.005.

CSI Tack et al. (2020) propose an OOD-specific contrastive learning algorithm. In addition, Tack et al. (2020) defines detection functions on top of the learned representation, combining two aspects: (1) the cosine similarity between the test sample embedding to the nearest training sample embedding and (2) the norm of the test sample embedding. As CSI requires specific training, which incurs non-tractible computational costs, we skip the computational complexity analysis for CSI here.

SSD Similar to Lee et al. (2018), Sehwag et al. (2020) design a Mahalanobis-based score under the representation learning scheme. In specific, Sehwag et al. (2020) propose a cluster-conditioned score:

$$\max_m -(\boldsymbol{e_x}/|\boldsymbol{e_x}| - \hat{\mu}_m)^T \hat{\Sigma}_m^{-1} (\boldsymbol{e_x}/|\boldsymbol{e_x}| - \hat{\mu}_m), \tag{20}$$

where $\boldsymbol{e_x}/|\boldsymbol{e_x}|$ is the normalized feature embedding of $\boldsymbol{x}$ and m corresponds to the cluster constructed from the training statistics.

Computing Eqn. (20) requires inverting m number of covariance matrix $\hat{\Sigma}_m$ prior to inference, which can be computationally expensive in high dimension. During inference, computing Eqn. (20) for each sample takes $O(|\mathcal{M}|P^2)$, where $|\mathcal{M}|$ is the number of clusters constructed in the algorithm and P is the dimension of the feature space. This indicates that the computational cost of MDS significantly grows for large-scale OOD detection problems.

KNN Sun et al. (2022) propose to detect OOD based on the k-th nearest neighbor distance between the normalized features of the test sample $\boldsymbol{z_x}/|\boldsymbol{z_x}|$ and the normalized training features on the penultimate space. Sun et al. (2022) also observe that contrastive learning helps improve OOD detection effectiveness.

In terms of computational complexity, normalizing the features is an $O(P)$ operation, where P is the embedding dimension. Computing the Euclidean distance between the normalized test feature and N training features is an $O(NP)$ operation. Additionally, searching for the k_{th} nearest distance out of N computed distances is a $O(N \log(N))$ operation. Therefore, the overall inference complexity of KNN is $O(NP + N \log(N))$. Comparing to our $O(P + |\mathcal{C}|)$ algorithm `fDBD` , KNN exhibits much lower scalability for large-scale OOD detection, especially when the number of training samples N significantly surpasses the number of classes $|\mathcal{C}|$.

ViM Wang et al. (2022) propose to integrate class-specific information into feature space information by adding energy score to the feature norm in the residual space of training feature matrix. The detection score is designed to be:

$$\alpha \sqrt{\boldsymbol{x}^T \boldsymbol{R} \boldsymbol{R} \boldsymbol{x}}, \tag{21}$$

where $\boldsymbol{R} \in R^{P \times (P-D)}$ correspond to the residual after subtracting the $D-$dimensional principle space. In the preparation stage, ViM requires evaluating the residual/null space from the training data, which is computationally expensive given the data volume. During inference, large matrix multiplication is required, resulting in a computational complexity of $O((P-D)^2)$.

H. Quantitative Study of the Proposed Distance Measuring method

In Section 3.1, we propose a closed-form estimation for measuring the feature distance to decision boundaries. To quantitatively understand the effectiveness and efficiency of our proposed method, we compare our method against measuring the distance via iterative optimization. In particular, we use targeted CW $L2$ attacks (Carlini & Wagner, 2017) on feature space which can effectively construct an adversarial example which is classified into the target class from an iterative process. Empirically, CW attack-based estimation and our closed-form estimation differ by $< 1.5\%$. This implies that our closed-form estimation differs from the true value by $< 1.5\%$, since estimation from a CW-attack upper bounds the distance whereas our closed-form estimation lower bounds the distance.

We follow the Pytorch implementation of CW attacks proposed by Papernot et al. (2018) with the default parameters: initial constant 2, learning rate 0.005, max iteration 500, and binary search step 3. In our experiments, CW-attack has a success rate close to 100%. On a Tesla T4 GPU, estimating the distance using CW attack takes 992.2ms per image per class. In contrast, our proposed method incurs negligible overhead in inference, significantly reducing the computational cost of measuring the distance.

I. Ablation Under FPR95

In addition to the AUROC score reported in the main paper, we report our ablation study here under FPR95, the false positive rate of OOD samples when the true positive rate of ID samples is at 95%. In Table 9, we compare the performance of OOD detection using the regularized average distance `regDistDB`, the regularization term $\|z - \mu_{train}\|_2$, as well as the un-regularized average distances `avgDistDB` as detection scores, respectively. Experiments are conducted on a ResNet-50 trained under cross-entropy loss following detailed setups in Section 4.2. The results in FPR95 further validate the effectiveness of regularization in our OOD detector.

Table 9. **Regularization enhances the effectiveness of OOD detection.** FPR95 scores reported on ImageNet Benchmarks (lower is better). `regDistDB` outperforms `avgDistDB`.

	iNaturalist	**SUN**	**Places**	**Texture**
avgDistDB	53.87	63.57	68.65	53.62
$\|z_x - \mu_{train}\|_2$	99.32	97.81	96.94	99.59
regDistDB	40.24	60.60	66.40	37.19

J. Feature Distances to Decision Boundaries

We extensively validate our hypothesis that ID features tend to reside further away from decision boundaries than OOD features in Figure 5, Figure 6, and Figure 7. To observe at a finer level of granularity, we sort per feature the estimated distances to all decision boundaries. On each subplot for a CIFAR-10 classifier, we plot 9 histograms, corresponding to the nearest distances, second nearest distance, and so on, up to the furthest distances. On each subplot for an ImageNet classifier, we sort the distance and plot every 100 ranked distances. We observe that ID features tend to reside further away from the decision boundaries compared to OOD samples across architectures and classification tasks.

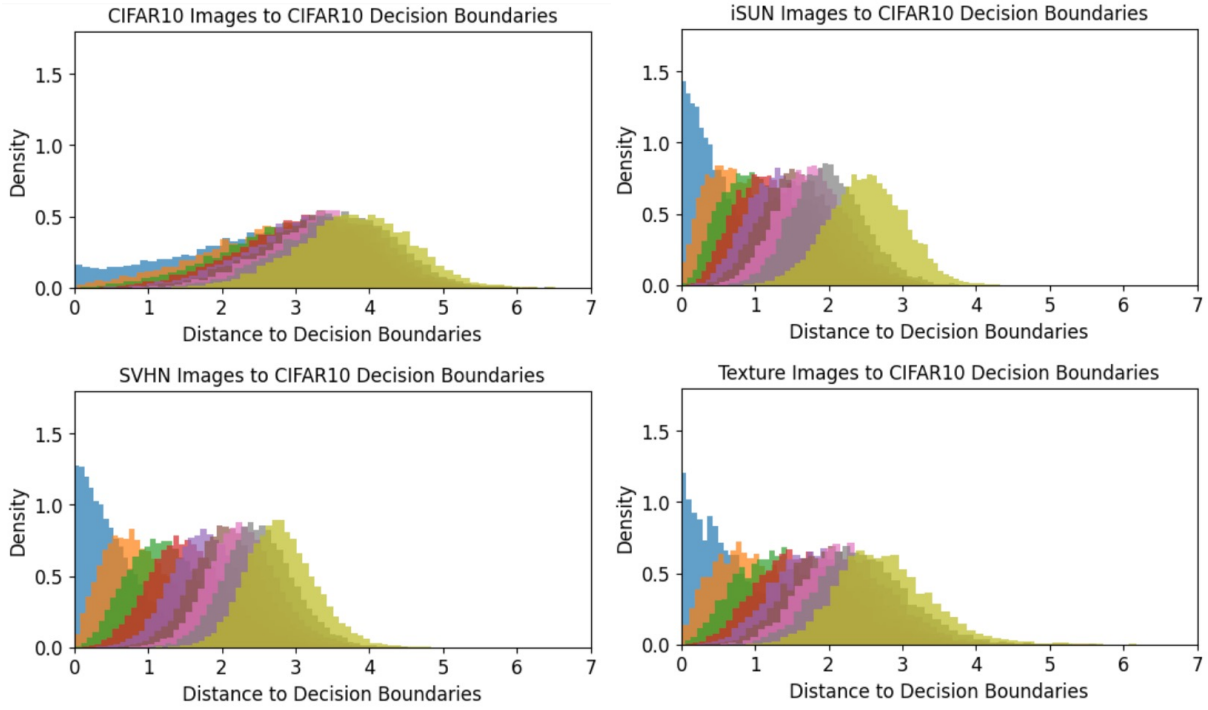

Figure 5. Feature Distances to Decision Boundaries on a **ResNet-18 CIFAR-10** Classifier. ID features tend to be further away from the decision boundaries compared to OOD features.

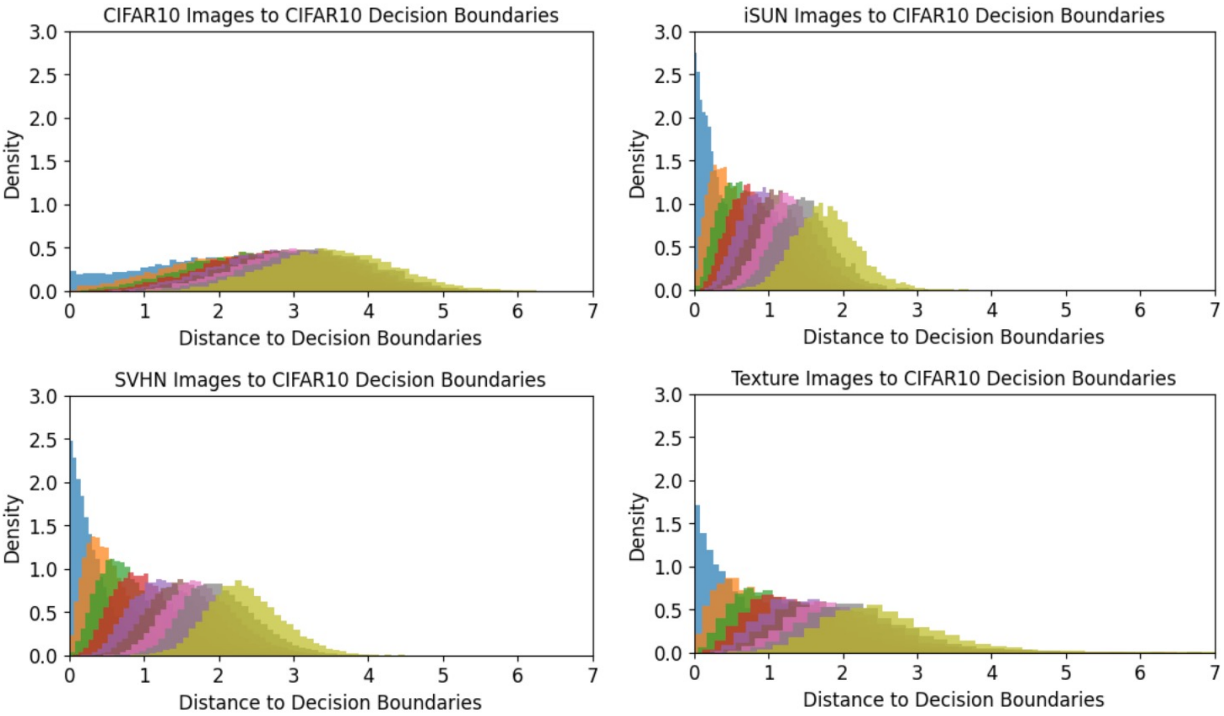

Figure 6. Feature Distances to Decision Boundaries on a **DenseNet CIFAR-10** Classifier. ID features tend to be further away from the decision boundaries compared to OOD features.

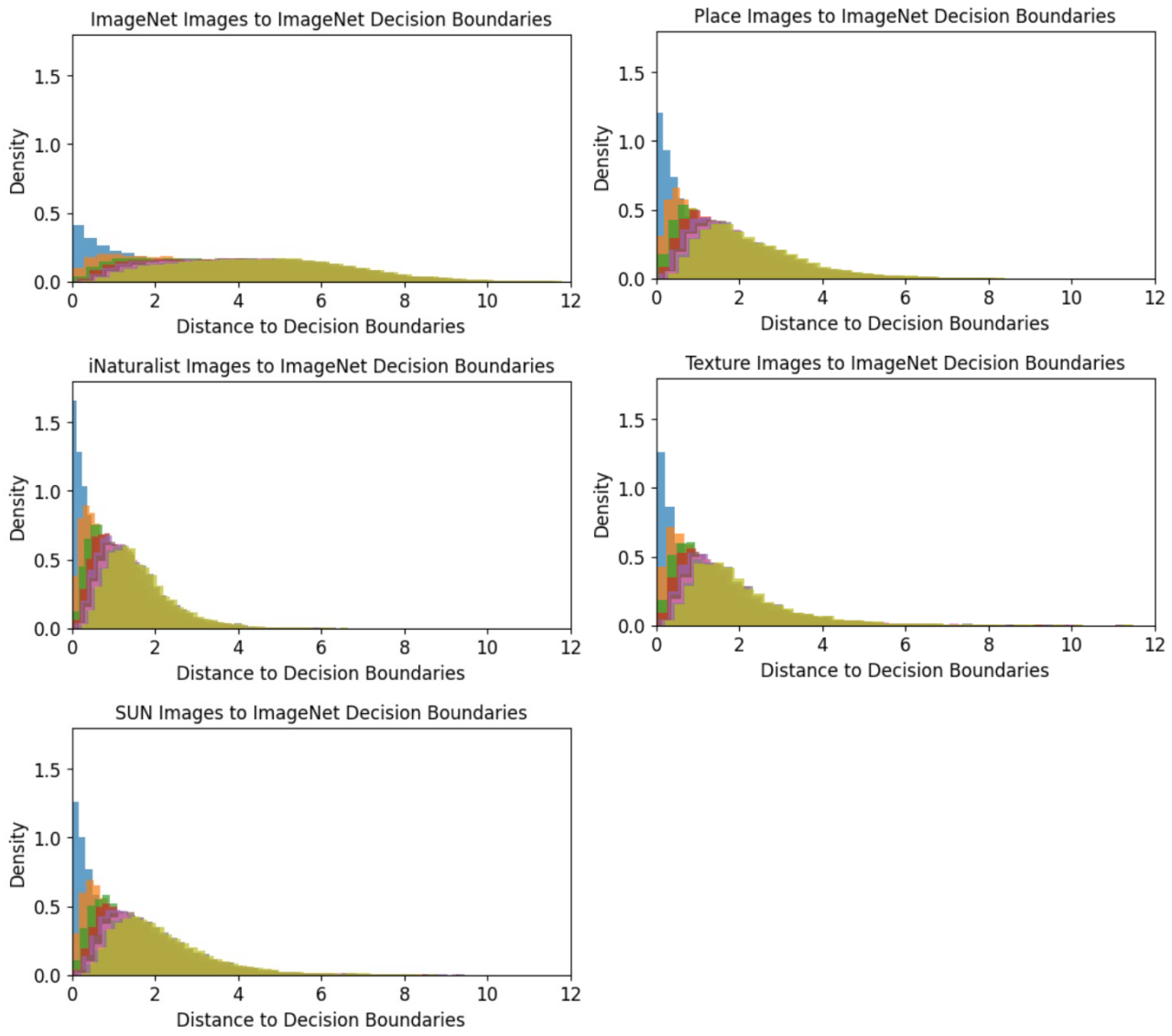

Figure 7. Feature Distances to Decision Boundaries on a **ResNet-50 ImageNet** Classifier. ID features tend to be further away from the decision boundaries compared to OOD features.