SIREN: Shaping Representations for OOD Detection

Anonymous Author(s)
Affiliation
Address
email

Abstract

Out-of-distribution (OOD) detection is indispensable for deploying machine learning models in the wild. Distance-based OOD detection methods are promising, but often suffer from discrepancies between the distributions learned in training vs. the distributional assumptions made in testing. This paper bridges the gap by addressing two key challenges—representation learning and OOD detection—in one coherent framework. Our proposed framework SIREN contributes two novel components: (1) a trainable loss function that shapes the representations into a mixture of von Mises-Fisher (vMF) distributions on the unit hypersphere, and (2) a test-time OOD detection score leveraging the learned vMF distributions. Unlike previous works, the two components in our framework enjoy strong mathematical compatibility with each other, under a unified distributional model. SIREN achieves competitive performance on both the recent detection transformers and CNN-based models, improving the AUROC by over 10% compared to the previous best method on detection transformers.

1 Introduction

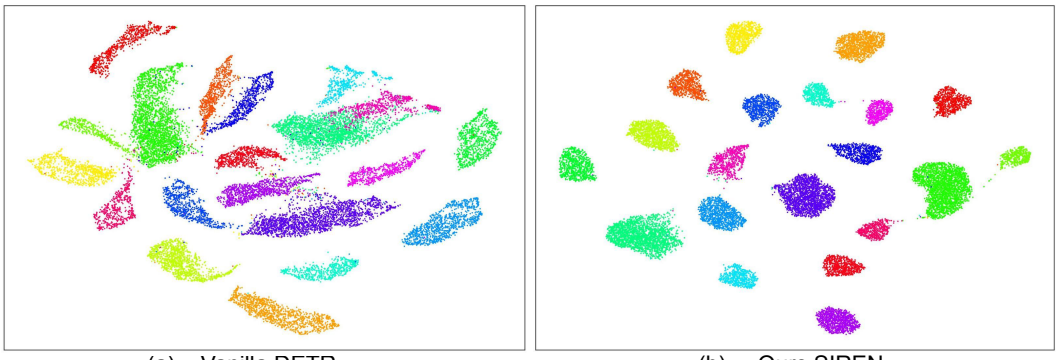
Teaching machine learning models to be aware of out-of-distribution (OOD) data is indispensable for building next-generation AI systems. For decades, machine learning has been operating in the closed-world setting. That is, a model will match an image to one of the given class labels, even if it is irrelevant. Instead, the open-world setting [3] emphasizes that samples from unknown classes can naturally emerge, which should not be blindly predicted into a known class. In safety-critical applications, such as autonomous driving, failing to detect OOD objects on the road can directly lead to disastrous accidents [42]. The situation can be better avoided if the object detection model recognizes the object as unfamiliar and appropriately cautions the human driver to take over [13].

A rich line of OOD detection algorithms have been recently developed, among which the distance-based methods become a promising recipe. For example, Lee *et al.* [33] modeled the feature embedding space as a mixture of multivariate Gaussian distributions and used the maximum Mahalanobis distance [37] to all class centroids for OOD detection. However, this approach makes a strong distributional assumption of the underlying feature space, which may not always hold in testing. To see this, we show that the modern object detection models such as DEFORMABLE-DETR [63] produce irregularly distributed embeddings (Figure 1 (a)), which do not fit the Gaussian distributional assumption. As a result, the OOD detection score relying on such mismatched assumptions can misbehave. This limitation thus motivates our work:

Can we shape the representations for OOD detection under a coherent distributional model?

We propose a novel framework called SIREN, tackling two highly dependent problems—representation learning and OOD detection—in one synergistic framework. That is, we facilitate the *match* between the distributions learned in training and the distributional model used in testing. Concretely, SIREN

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.



(a) Vanilla DETR (b) Ours SIREN

Figure 1: (a) Feature embeddings from the penultimate layer of a vanilla DEFORMABLE-DETR [63] trained on the PASCAL-VOC dataset [14], which display irregular distributions. (b) Feature embeddings shaped by the proposed SIREN, which form compact clusters on the unit hypersphere.

contributes two novel components: **(1)** we introduce a trainable loss function that enables **S**hap**I**ng the **R**epres**EN**tations into a desired parametric form (Section 3.1). In particular, we model the representations by the von Mises-Fisher (vMF) distribution, a probability distribution in directional statistics for hyperspherical data with unit norm. Our loss function encourages the normalized embedding to be aligned with its class prototype and shapes the overall representations into compact clusters for each class. **(2)** Based on the shaped embeddings, we propose a novel test-time OOD detection score by leveraging the vMF distributions (Section 3.2). The parameterization of the vMF distribution is directly obtainable after training—no separate estimation is needed. Different from the Mahalanobis distance, the proposed score in principle suits the learned vMF distributions on the hypersphere. Thus, the two components in our framework operate under a consistent distributional model and enjoy strong mathematical compatibility with each other.

Empirically, SIREN establishes superior performance on transformer-based and CNN-based models. On PASCAL-VOC, SIREN outperforms the latest baseline OW-DETR [17] by a significant margin (20.4% in AUROC). Moreover, our framework is model-agnostic and does not incur change to the existing network architecture. The proposed loss can be flexibly added as a plug-in module on top of modern architectures, as we show in Section 4. Our key **contributions** are summarized as follows:

1. To the best of our knowledge, SIREN pioneers the effort of unifying the distributional model between training and testing for OOD detection. Different from the previous works, both our representation learning and OOD detection operate under a consistent distributional model and enjoy strong mathematical compatibility.
2. SIREN establishes state-of-the-art results on a challenging object-level OOD detection task [13]. Compared to the latest method [17], SIREN improves the OOD detection performance by a considerable margin while preserving the mAP on the ID task. We show that SIREN is effective for both recent transformer-based and classic CNN-based models.
3. We make the first attempt to shape representations via a vMF-based formulation for OOD detection. We conduct in-depth ablations to understand how different factors impact the performance of SIREN (Section 5).

2 Preliminaries: Object-level OOD Detection

We start by introducing the OOD detection problem for object detection in the open-world setting, which has received increasing research attention lately [13, 17]. Our goal is to train object detection networks that can simultaneously: (1) accurately localize and classify objects belonging to known categories, and (2) identify unfamiliar objects outside the training categories.

Compared to image-level OOD detection, object-level OOD detection is more relevant for real-world machine learning systems, yet also more challenging as it requires reasoning OOD uncertainty at the fine-grained object level. Since natural images are composed of numerous objects and components, knowing which regions of an image are anomalous allows for safe handling of unfamiliar objects.

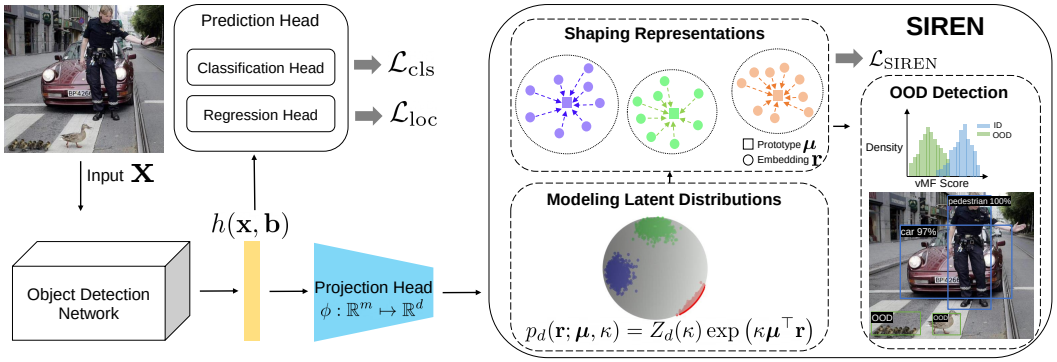


Figure 2: **Overview of the proposed learning framework SIREN.** We introduce a new loss $\mathcal{L}_{\text{SIREN}}$ which shapes the representations on the unit hypersphere into compact class-conditional vMF distributions. The embedding $\mathbf{r} \in \mathbb{R}^d$ has unit norm $\|\mathbf{r}\|^2 = 1$. In testing, we propose a vMF-based uncertainty score for OOD detection. See Section 3 for details.

Notations. We denote the input and label space by $\mathcal{X} = \mathbb{R}^q$ and $\mathcal{Y} = \{1, 2, ..., C\}$, respectively. Let $\mathbf{x} \in \mathcal{X}$ be the input image, $\mathbf{b} \in \mathbb{R}^4$ be the bounding box coordinates associated with objects in the image, and $y \in \mathcal{Y}$ be the semantic label of the object. An object detection model is trained ID data $\mathcal{D}_{\text{tr}}^{\text{in}} = \{(\mathbf{x}_i, \mathbf{b}_i, y_i)\}_{i=1}^{M}$ drawn from an unknown joint distribution $\mathcal{P}$. We use neural networks with parameters θ to model the bounding box regression $p_\theta(\mathbf{b}|\mathbf{x})$ and the classification $p_\theta(y|\mathbf{x}, \mathbf{b})$.

OOD detection for object detection model. The OOD detection can be formulated as a binary classification problem, distinguishing between the in- vs. out-of-distribution objects. Let $P_{\mathcal{X}}$ denote the marginal probability distribution on $\mathcal{X}$. Given a test input $\mathbf{x}' \sim P_{\mathcal{X}}$, as well as an object $\mathbf{b}'$ predicted by the object detector, the goal is to predict $p_\theta(g|\mathbf{x}', \mathbf{b}')$. We use $g = 1$ to indicate a detected object being ID, and $g = 0$ being OOD, with semantics outside the support of $\mathcal{Y}$.

3 Proposed Method

A motivating example. Our framework is motivated by the limitations of existing distance-based approaches, which operate under the assumption that the test OOD samples are relatively far away from the ID data in the latent space. For example, Lee *et al.* [33] modeled the feature embedding space as a mixture of multivariate Gaussian distributions, and used the maximum Mahalanobis distance [37] to all class centroids for OOD detection. However, this approach makes a strong distributional assumption of the underlying feature space, which may not necessarily hold during inference. To see this, we show that modern object detection models such as DDETR [63] produces irregularly distributed features in its penultimate layer (Figure 1 (a)), which do not match the Gaussian distributional assumption. Our learning framework targets this problem.

Overview. Our framework SIREN is illustrated in Figure 2, which trains an object detection model in tandem with a novel representation shaping branch. The object detection network backbone $f : \mathcal{X} \mapsto \mathbb{R}^m$ maps an object to its feature embedding $h(\mathbf{x}, \mathbf{b}) \in \mathbb{R}^m$ (often referred to as the penultimate layer). In addition, we introduce a new MLP projection head $\phi : \mathbb{R}^m \mapsto \mathbb{R}^d$ that maps the $h(\mathbf{x}, \mathbf{b})$ to a lower-dimensional embedding $\mathbf{r} \in \mathbb{R}^d$ $(d < m)$ with unit norm $\|\mathbf{r}\|^2 = 1$. The normalized embeddings are also referred to as *hyperspherical embeddings*, since they are on a unit hypersphere. In designing SIREN, we address two key challenges: **(1)** How to shape the hyperspherical representations into desirable probability distributions during training time (Section 3.1)? **(2)** How to perform test-time OOD detection by leveraging the learned embeddings and their distributions (Section 3.2)? Our method does not incur any change to the object detection network backbone. The proposed regularization can be flexibly used as a plug-in module on top of the modern architectures, as we will show in Section 4.

3.1 SIREN: Shaping Representations

Modeling the latent distributions. We propose to model the latent representations by the von Mises-Fisher (vMF) distribution [38], a probability distribution in directional statistics for spherical

data with unit norm $\|\mathbf{r}\|^2 = 1$. The probability density function for a unit vector $\mathbf{r}$ in $\mathbb{R}^d$ is given as follows:

$$p_d(\mathbf{r}; \boldsymbol{\mu}, \kappa) = Z_d(\kappa) \exp\left(\kappa \boldsymbol{\mu}^\top \mathbf{r}\right), \tag{1}$$

where $\kappa \geq 0$, $\|\boldsymbol{\mu}\|^2 = 1$, and the normalization factor $Z_d(\kappa)$ is defined as:

$$Z_d(\kappa) = \frac{\kappa^{d/2-1}}{(2\pi)^{d/2} I_{d/2-1}(\kappa)}, \tag{2}$$

where I_v is the modified Bessel function of the first kind with order v. $Z_d(\kappa)$ can be calculated in closed form based on κ and the dimensionality d. Importantly, the vMF distribution is characterized by two parameters: the mean vector $\boldsymbol{\mu}$ and concentration κ. Datapoints that are more aligned with the center $\boldsymbol{\mu}$ have a higher probability density, and vice versa. Here κ indicates the tightness of the distribution around the mean direction $\boldsymbol{\mu}$. The larger value of κ, the stronger the distribution is concentrated in the mean direction. In the extreme case of $\kappa = 0$, the sample points are distributed uniformly on the hypersphere.

When considering multiple classes, we model the embedding space as a mixture of class-conditional vMF distributions, one for each class $c \in \{1, 2, ..., C\}$:

$$p_d^c(\mathbf{r}; \boldsymbol{\mu}_c, \kappa_c) = Z_d(\kappa_c) \exp\left(\kappa_c \boldsymbol{\mu}_c^\top \mathbf{r}\right), \tag{3}$$

where κ_c and $\boldsymbol{\mu}_c$ are class-conditional parameters. Under this probability model, an embedding vector $\mathbf{r}$ is assigned to class c with the following normalized probability:

$$p(y = c|\mathbf{r}; \{\kappa_j, \boldsymbol{\mu}_j\}_{j=1}^C) = \frac{Z_d\left(\kappa_c\right) \exp\left(\kappa_c \boldsymbol{\mu}_c^\top \mathbf{r}\right)}{\sum_{j=1}^C Z_d\left(\kappa_j\right) \exp\left(\kappa_j \boldsymbol{\mu}_j^\top \mathbf{r}\right)}. \tag{4}$$

Shaping representations. Our key idea is to design a trainable loss function that enables **S**hap**I**ng the **R**epres**EN**tations into a mixture of vMF distributions, which facilitates test-time OOD detection (Section 3.2). We therefore name our method SIREN. The learned mapping function projects an input to a point in the embedding space, where higher probability is assigned to the correct class in comparison to incorrect classes. To achieve this, we can perform maximum likelihood estimation (MLE) on the training data:

$$\text{argmax}_\theta \prod_{i=1}^M p(y_i|\mathbf{r}_i; \{\kappa_j, \boldsymbol{\mu}_j\}_{j=1}^C), \tag{5}$$

where i is the index of the object embedding and M is the size of the training set. By taking the negative log-likelihood, the objective function is equivalent to minimizing the following loss:

$$\mathcal{L}_{\text{SIREN}} = -\frac{1}{M} \sum_{i=1}^M \log \frac{Z_d\left(\kappa_{y_i}\right) \exp\left(\kappa_{y_i} \boldsymbol{\mu}_{y_i}^\top \mathbf{r}_i\right)}{\sum_{j=1}^C Z_d\left(\kappa_j\right) \exp\left(\kappa_j \boldsymbol{\mu}_j^\top \mathbf{r}_i\right)}, \tag{6}$$

where y_i is the ground truth label for the embedding $\mathbf{r}_i$. In effect, $\mathcal{L}_{\text{SIREN}}$ encourages the object embeddings to be aligned with its class prototype, which shapes the representations such that samples in each class form a compact cluster on the hypersphere; see Figure 1 (b).

Prototype estimation and update. During training, SIREN estimates the class-conditional object prototypes $\boldsymbol{\mu}_c, c \in \{1, 2, ..., C\}$. The conventional approach for estimating the prototypes is to calculate the mean vector of all training samples (or a subset of them) for each class, and update it periodically during training [61]. Despite its simplicity, this method requires alternating training and prototype estimation, which incurs a heavy computational toll and causes undesirable latency. Instead, we update the class-conditional prototypes in an exponential-moving-average (EMA) manner:

$$\boldsymbol{\mu}_c := \text{Normalize}(\alpha \boldsymbol{\mu}_c + (1 - \alpha)\mathbf{r}), \forall c \in \{1, 2, \ldots, C\}, \tag{7}$$

where α is the prototype update factor, and $\mathbf{r}$ denotes the normalized object embeddings from class c. The update can be done efficiently with negligible cost, enabling end-to-end training.

Overall training objective. The overall training objective combines the standard object detection loss, along with our new representation shaping loss:

$$\min_\theta \mathbb{E}_{(\mathbf{x}, \mathbf{b}, y) \sim \mathcal{P}} \left[\mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{loc}}\right] + \beta \cdot \mathcal{L}_{\text{SIREN}}, \tag{8}$$

where β is the weight of the representation shaping loss. $\mathcal{L}_{\text{cls}}$ and $\mathcal{L}_{\text{loc}}$ are losses for classification and bounding box regression, respectively. We provide extensive empirical evidence in Section 4 demonstrating the efficacy of our loss function.

Algorithm 1 SIREN: Shaping Representations for OOD detection

Input: ID training data $\mathcal{D}_{\text{tr}}^{\text{in}} = \{(\mathbf{x}_i, \mathbf{b}_i, y_i)\}_{i=1}^{M}$, randomly initialized object detector and MLP projection head with parameter θ, loss weight β for $\mathcal{L}_{\text{SIREN}}$, and learnable $\{\kappa_c\}_{c=1}^{C}$.
Output: Object detector with parameter θ^* and OOD detector G.
while *train* **do**
  1. Update class-conditional prototypes $\boldsymbol{\mu}_c$ with the hyperspherical embeddings $\mathbf{r}$ by Equation (7).
  2. Calculate the representation shaping loss $\mathcal{L}_{\text{SIREN}}$ by Equation (6).
  3. Update the learnable $\{\kappa_c\}_{c=1}^{C}$ and the network parameters θ using Equation (8).

end
while *eval* **do**
  1. Calculate the OOD score by Equation (9).
  2. Perform OOD detection by Equation (10).

end

Remark 1. *To the best of our knowledge, our work makes the first attempt exploring vMF-based learning and inference for OOD detection. To highlight the novelty of the loss itself: (1) We introduce a novel* ***learnable*** $\{\kappa_j\}_{j=1}^{C}$ *for each ID class in Equation* (6)*, whereas previous works have mostly used fixed values. Our loss allows concentration parameter κ to adaptively and flexibly capture the class-conditional feature statistics during training. This is desirable since each ID class may have its own concentration in the hypersphere. We will later show that such learnable $\{\kappa_j\}_{j=1}^{C}$ enables better OOD detection performance (Section 5). (2) We introduce the prototype updates using EMA, whereas previous works require expensive alternating training and prototype estimation.*

3.2 Test-time OOD Detection

During inference, we propose a new test-time OOD detection score based on the learned class-conditional vMF distributions, parameterized by $\{\widehat{\kappa}_c, \boldsymbol{\mu}_c\}_{c=1}^{C}$. Here $\widehat{\kappa}_c$ denotes the learned concentration parameter for class c, which captures the concentration of representations for class c. For a test-time object $(\mathbf{x}', \mathbf{b}')$, we use the largest estimated class-conditional likelihood as the OOD score:

$$S\left(\mathbf{x}', \mathbf{b}'\right) = \max_{c} Z_d(\widehat{\kappa}_c) \exp\left(\widehat{\kappa}_c \boldsymbol{\mu}_c^{\top} \mathbf{r}'\right), \tag{9}$$

where $\mathbf{r}' = \phi(h(\mathbf{x}', \mathbf{b}'))$ is the normalized embedding from the MLP projection head. Our OOD detection score thus in principle suits our learned embeddings and vMF distributions.

For OOD detection, one can use the level-set to distinguish between ID and OOD objects:

$$G\left(\mathbf{x}', \mathbf{b}'\right) = \begin{cases} 1 & \text{if } S\left(\mathbf{x}', \mathbf{b}'\right) \geq \gamma \\ 0 & \text{if } S\left(\mathbf{x}', \mathbf{b}'\right) < \gamma \end{cases} \tag{10}$$

The rationale is that ID object will be matched with one of the known classes with high probability, and vice versa for OOD. The threshold γ can be chosen so that a high fraction of ID data (e.g., 95%) is correctly classified. For objects classified as ID, one can obtain the bounding box and class prediction using the prediction head as usual. The complete algorithm is summarized in Algorithm 1.

Remark 2. *Different from Mahalanobis distance [33], our proposed test-time OOD detection score operates under the same distributional model as the training process, and hence enjoys strong mathematical compatibility. Computationally, we can directly use the parameters of vMF distributions (such as κ) learned from training. In contrast, the Mahalanobis distance requires a separate test-time estimation of feature statistics—which involves an expensive and numerically unstable step of calculating the covariance matrix. The distinction of the uncertainty surface calculated by both our vMF score and the Mahalanobis distance is qualitatively demonstrated in Figure 3. The data points are sampled from a mixture of three class-conditional vMF distributions (see details in Appendix I).*

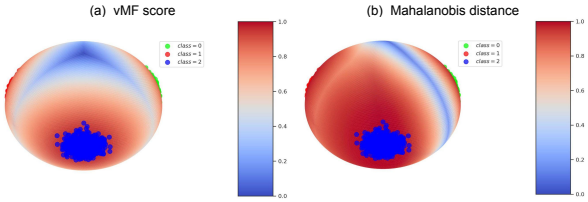


Figure 3: The uncertainty surface calculated using our vMF score (a) and the Mahalanobis distance (b). We showcase one class for visual clarity.

In-distribution dataset	Method	FPR95 ↓	AUROC ↑	mAP (ID)↑
		OOD: MS-COCO / OpenImages		
PASCAL-VOC	MaxLogit [21]	96.61 / 97.11	46.60 / 43.98	60.6
	MSP [22]	85.96 / 86.75	64.55 / 61.07	60.6
	ODIN [34]	85.29 / 89.46	65.13 / 61.39	60.6
	Mahalanobis [33]	97.39 / 97.88	50.28 / 49.08	60.6
	Gram matrices [47]	94.16 / 95.29	43.97 / 38.81	60.6
	Generalized ODIN [24]	92.08 / 92.83	57.34 / 55.74	60.4
	CSI [53]	84.00 / 79.16	55.07 / 51.37	59.5
	VOS [13]	97.46 / 97.07	54.40 / 52.77	60.3
	OW-DETR [17]	93.09 / 93.82	55.70 / 57.80	58.3
	SIREN (ours)	**75.49**±0.8 / 78.36±1.0	76.10±0.1 / 71.05±0.1	60.8±0.1
	SIREN-DETReg (ours)	75.80±0.6 / **78.33**±0.9	**77.50**±0.5 / **71.55**±0.6	**62.8**±0.3
BDD-100K	MaxLogit [21]	90.90 / 88.62	67.43 / 70.57	31.3
	MSP [22]	78.48 / 78.24	73.34 / 72.48	31.3
	ODIN [34]	79.00 / 79.66	74.51 / 74.49	31.3
	Mahalanobis [33]	70.86 / 71.43	76.83 / 77.98	31.3
	Gram matrices [47]	73.81 / 71.56	60.13 / 57.14	31.3
	Generalized ODIN [24]	72.70 / 69.12	75.37 / 78.95	31.2
	CSI [53]	70.27 / 71.30	77.93 / 76.42	29.9
	VOS [13]	76.44 / 72.58	77.33 / 76.62	31.0
	OW-DETR [17]	80.78 / 77.37	70.29 / 73.78	28.1
	SIREN (ours)	67.54±1.3 / 66.31±0.9	**80.06**±0.5 / **79.77**±1.2	31.3±0.0
	SIREN-DETReg (ours)	**66.90**±0.7 / **66.10**±0.2	78.99±1.1 / 79.01±1.8	**32.5**±0.1

Table 1: **Main results.** Comparison with competitive out-of-distribution detection methods. All baseline methods are based on the same model backbone DDETR. ↑ indicates larger values are better and ↓ indicates smaller values are better. All values are percentages. **Bold** numbers are superior results. We report standard deviations estimated across 3 runs. DETReg denotes the model SIREN pretrained with a more powerful self-supervised approach [44] for the DDETR.

4 Experiments

In this section, we validate the effectiveness of SIREN on flagship object detection models, including the latest transformer-based (Section 4.1) and traditional CNN-based models (Section 4.2).

Datasets. Following [13], we use PASCAL-VOC [14] and Berkeley DeepDrive (BDD100K) [59] datasets as the ID training data. For both tasks, we evaluate on two OOD datasets that contain a subset of images from: MS-COCO [35] and OPENIMAGES (validation set) [30]. Extensive details on the datasets are in Appendix B and C.

Metrics. For evaluating the OOD detection performance, we report: (1) the false positive rate (FPR95) of OOD samples when the true positive rate of ID samples is at 95%; (2) the area under the receiver operating characteristic curve (AUROC). For evaluating the object detection performance on the ID task, we report the common metric mAP.

4.1 Evaluation on Transformer-based Model

Experimental details. Following [17], we adopt a very recent DEFORMABLE-DETR (DDETR) architecture [63]. DDETR introduces multi-scale deformable attention modules in the transformer encoder and decoder layers of DETR [4], and provides better convergence and lower complexity. The multi-scale feature maps in DDETR are extracted from a ResNet-50 [20] pretrained on ImageNet in a self-supervised fashion, *i.e.*, DINO [5]. For the projection head, we use a two-layer MLP with a ReLU nonlinearity, with dimensionality $256 \rightarrow d \rightarrow d$. The dimension d of the unit hypersphere is 16 for PASCAL-VOC and 64 for BDD100K. The default weight β for the SIREN is 1.5 and the prototype update factor α is 0.95. We initialize the learnable κ to be 10 for all classes. Ablations on the hyperparameters are provided in Section 5 and Appendix E. The other hyperparameters are the same as the default ones in DDETR [63].

SIREN achieves superior performance. In Table 1, we compare SIREN with competitive OOD detection methods in literature. For a fair comparison, all the methods only use ID data for training. SIREN outperforms competitive baselines, including MaxLogit [21], MSP [22], ODIN [34], Mahalanobis distance [33], Generalized ODIN [24], CSI [53], and Gram matrices [47]. These approaches rely on a classification model, and can be naturally extended to the object detection model due to the existence of a classification head. Implementation details and the training time for all baseline methods are reported in Appendix F and G. We highlight a few interesting observations:

- The comparison between **SIREN vs. Mahalanobis** highlights precisely the benefits of our embedding shaping loss $\mathcal{L}_{\text{SIREN}}$. As shown in Figure 1, the vanilla DETR model produces ill-conditioned embeddings that do not conform to multivariate Gaussian distributions, rendering Mahalanobis approach ineffective (with AUROC around 50%–which is random guessing). In contrast, SIREN improves the OOD detection performance (AUROC) by **25.82**% on PASCAL-VOC (MS-COCO as OOD). Different from the Mahalanobis distance, our test-time scoring function naturally suits the learned hyperspherical embeddings with vMF distributions.
- SIREN outperforms the latest methods OW-DETR [17] and VOS [13], which are designed specifically for object detection models and serve as strong baselines for us. Compared with OW-DETR, SIREN substantially improves the AUROC by **20.4**% on PASCAL-VOC (MS-COCO as OOD). OW-DETR uses the unmatched object queries with high confidence as the unknowns, and trains a binary classifier to separate ID and unknown objects. However, the unmatched object queries might be distributionally too close to the ID classes and thus displays limited improvement for OOD detection. In addition, VOS synthesizes virtual outliers from the class-conditional Gaussian distributions of the penultimate layer but fails to perform well due to the ill-conditioned embedding distribution in DDETR (non-Gaussian).
- Lastly, SIREN's performance may be further improved by initializing the model using DETReg [2], which introduces a more powerful self-supervised pretraining method. For example, on PASCAL-VOC, SIREN-DETReg improves the AUROC by 1.4% (MS-COCO as the OOD), compared to the DINO initialization.

4.2 Evaluation on CNN-based Models

Going beyond detection transformers, we show that SIREN is also suitable and effective on CNN-based models. Table 2 showcases the performance on two tasks: a Faster R-CNN-based object detector [46] and image classification networks. For Faster R-CNN, we train on PASCAL-VOC dataset and evaluate on both MS-COCO and OPENIMAGES datasets.

Method	FPR95 ↓	AUROC ↑
	ResNet / DenseNet	
Vanilla cls. network	70.71 / 40.91	79.23 / 88.32
SIREN (ours)	**64.56 / 40.67**	**84.86 / 92.37**
	COCO / OpenImages	
Vanilla Faster R-CNN	96.46 / 96.27	59.25 / 57.42
SIREN (ours)	**65.45 / 67.33**	**84.60 / 81.51**

Table 2: OOD detection results of SIREN and comparison with Mahalanobis distance on the penultimate layer. Two architectures are considered: Faster R-CNN and classification networks.

For classification networks, we use CIFAR-10 [29] as the ID training data, with standard train/val splits. We train on ResNet-18 [20] and DenseNet-101 [25], where we substitute the object detection loss in Equation (8) with the cross-entropy loss. We evaluate on six OOD datasets: TEXTURES [7], SVHN [41], PLACES365 [62], LSUN-C [60], LSUN-RESIZE [60], and ISUN [57]. The comparisons are shown in Table 2, with results averaged over six test datasets. SIREN consistently improves OOD detection performance on all three architectures.

5 Ablations and Discussions

In this section, we provide ablation results on how different factors impact the performance of SIREN. For consistency, we present the analyses below based on the DDETR model.

Ablations on learnable vs. fixed κ. In this ablation, we show that our approach using learnable concentration parameters $\{\kappa_c\}_{c=1}^{C}$ is better than using the fixed ones. Our method with learnable κ is desirable, since each ID class may have its own concentration in the hypersphere. With fixed κ, the loss function can be simplified as follows:

$$\mathcal{L} = -\frac{1}{M}\sum_{i=1}^{M}\log\frac{\exp\left(\kappa\boldsymbol{\mu}_{y_i}^{\top}\mathbf{r}_i\right)}{\sum_{j=1}^{C}\exp\left(\kappa\boldsymbol{\mu}_{j}^{\top}\mathbf{r}_i\right)}. \tag{11}$$

Empirically, we indeed observe that employing learnable $\{\kappa_c\}_{c=1}^{C}$ achieves better OOD detection performance, with AUROC 76.10% on PASCAL-VOC model (MS-COCO as OOD). In Figure 4 (a), we show SIREN's performance when trained under different fixed κ values. The best model under the fixed $\kappa = 10$ achieves an AUROC of 74.54%. Too small of κ value (*e.g.*, $\kappa = 1$) leads to almost uniform distributions and is therefore not desirable.

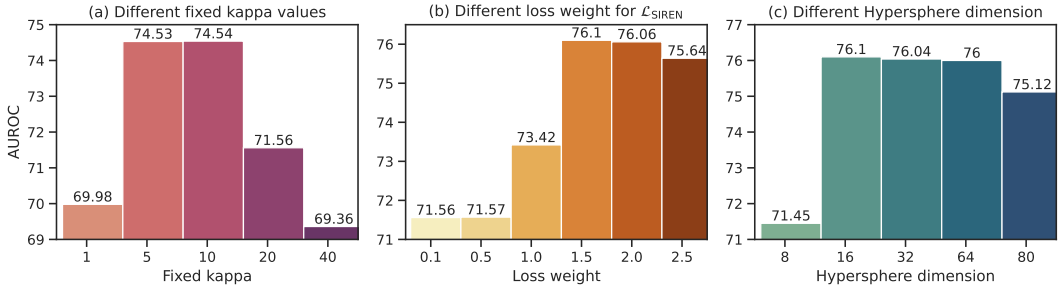


Figure 4: (a) Ablation study on using different fixed values of concentration parameter κ in Equation (11). (b) Ablation on different weights β for $\mathcal{L}_{\text{SIREN}}$ in Equation (8). (c) Ablation on the dimension of the hypersphere embeddings $\mathbf{r}$. Numbers are AUROC. The OOD dataset is MS-COCO and the ID training dataset is PASCAL-VOC.

Ablations on the vMF uncertainty score. We perform ablation on two variants of the vMF-based OOD detection score (*c.f.* Equation (9)): using the learned $\{\kappa_c\}_{c=1}^{C}$ vs. approximately estimate the κ parameters directly from the converged embeddings. In literature, there are several established methods for approximating κ [49, 38]. Denote $\bar{\mathbf{r}}_c$ the average of object embeddings for class c, the simplest approximate solution is given by:

$$\widehat{\kappa_c} = \frac{\|\bar{\mathbf{r}}_c\|(d - \|\bar{\mathbf{r}}_c\|^2)}{1 - \|\bar{\mathbf{r}}_c\|^2}, \tag{12}$$

with proof given in Appendix D. We show in Table 3 *(left)* that using learned $\{\kappa_c\}_{c=1}^{C}$ avoids the imprecision in approximating $\widehat{\kappa_c}$ directly from embeddings. This affirms the importance of employing learnable concentration parameters in our SIREN loss.

Ablations on the SIREN loss weight β. Figure 4 (b) reports the OOD detection results as we vary the weight β for the representation shaping loss $\mathcal{L}_{\text{SIREN}}$. The model is evaluated on both MS-COCO and OPENIMAGES datasets as OOD. Overall a mild weight works well. Across all the β values considered, SIREN consistently outperforms the baseline OOD detection methods in Table 1. One can use our SIREN loss as an easy plug-in module, with minimal hyperparameter tuning.

Ablations on the dimension d of hypersphere. SIREN projects the object feature embeddings into a lower-dimensional hypersphere in $\mathbb{R}^d$, which allows tractable vMF estimation. Figure 4 (c) shows the effect the embedding dimension d on OOD detection performance. We find that a lower dimension between 16 and 64 achieves favorable and stable performance. In the extreme case with dimension $d = 8$, the model suffers from significant information loss and degrades the performance. On the other hand, too large of d causes training instability, which is not desirable either.

Ablations on the projection head. We ablate on the nonlinearity in the projection head by comparing with SIREN trained with a single linear layer in Table 3 (*right*). The result demonstrates using a nonlinear mapping for projection helps to get a more meaningful and expressive hypersphere, which is able to improve OOD detection by 4.37% in terms of AUROC (MS-COCO as OOD).

	FPR95 ↓	AUROC ↑
	COCO / OpenImages as OOD	
Approximated κ [49]	78.60 / 78.42	73.03 / 70.27
Learned κ (ours)	**75.49 / 78.36**	**76.10 / 71.05**

	FPR95 ↓	AUROC ↑
	COCO / OpenImages as OOD	
w/o Nonlinearity	82.85 / 83.69	71.73 / 66.82
Ours	**75.49 / 78.36**	**76.10 / 71.05**

Table 3: **Left:** Ablation on κ. **Right:** Ablation on the projection head. The ID dataset is PASCAL-VOC.

6 Qualitative analysis

In Figure 5, we visualize the predictions on several OOD images, using object detection models trained without SIREN (top) and with SIREN (bottom), respectively. The in-distribution data is BDD100K. SIREN better identifies OOD objects (in green) compared to a vanilla object detector DDETR, reducing false positives. Moreover, the confidence score of the false-positive objects of SIREN is lower than that of the vanilla model (see the train/bicycle in the 3rd/5th column).

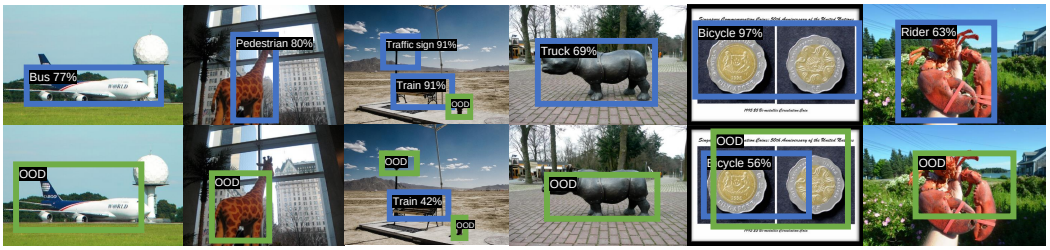


Figure 5: Visualization of detected objects on the OOD images (from MS-COCO and OPENIMAGES) by vanilla DDETR (*top*) and SIREN (*bottom*). The ID is BDD100K dataset. **Blue**: OOD objects classified as the ID classes. **Green**: OOD objects detected by DDETR and SIREN, which reduce false positives.

7 Related work

OOD detection for classification can be broadly categorized into post hoc and regularization-based approaches. In [3], the OpenMax score is proposed for OOD detection based on the extreme value theory (EVT). Subsequent works [22] made a simple baseline using maximum softmax probability. Several improvements have been proposed, such as deep ensemble [31], ODIN [34], Mahalanobis distance [33, 45], Gram matrices [47], energy score [36] and ReAct score [52]. While previous post hoc works often considered the OOD scoring function alone, our work proposes a unified framework with both representation learning (at training) and OOD detection (at testing).

Along the line of regularization-based approaches, most prior works focused on regularizing the model's *output*. For example, the model is regularized to produce lower confidence [23, 32] or higher energy [13, 36] on the outlier data. In contrast, SIREN regularizes the model by shaping the *representations* during training. Tack *et al.* [53] employed self-supervised learning to learn desirable representations for novelty detection, but they used complex score ensembles during inference, which is sub-optimal (see Table 1). In SIREN, both our representation learning and OOD score are based on a unified distributional model. Sun *et al.* [51] shaped the latent representations as multivariate Gaussians but they trained a variational autoencoder and utilized the reconstruction error score for open-set recognition. In contrast, SIREN directly shapes the embeddings of a *discrminative-based model*, and does not require training a generative model.

OOD detection for object detection is a rising topic with very few existing works. For Faster R-CNN, Du *et al.* [13] proposed to synthesize virtual outliers in the feature space for model regularization, which we compare in Section 4. [12] explored unknown-aware object detection by leveraging videos in the wild, whereas we focus on settings with still images only. In DETR, [17] adopted unmatched object queries that are with high confidence as unknowns, which did not focus on regularizing the model for desirable representations. Several works [9, 18, 11, 39, 40] used approximate Bayesian methods, such as MC-Dropout [15] for OOD detection. They require multiple inference passes to generate the uncertainty score, which are computationally expensive on larger datasets and models.

vMF distribution in ML has been adopted for supervised classification [28, 48], face verification [19], generative modeling [8], few-shot learning [43], semantic segmentation [26] and clustering [1, 16], etc. Some works exploited vMF distribution for anomaly detection, they employed generative models and used the vMF distribution as the prior for zero-shot learning [6] and document analysis [64]. Both the problem settings and approaches are orthogonal to SIREN. To the best of our knowledge, our work makes the first attempt to employ vMF-based learning and inference with learnable parameters for OOD detection. Metric learning has been explored for both classification [10, 54, 61, 58] and object detection [27, 50, 55, 56]. Unlike ours, they did not focus on the OOD detection problem.

8 Conclusion

In this paper, we propose a novel framework SIREN, which tackles representation learning and OOD detection in one coherent framework. SIREN mitigates the key shortcoming of the previous distance-based OOD detection approach, and unifies the distributional model between training and testing for OOD detection. To the best of our knowledge, SIREN makes the first attempt to employ vMF-based learning and inference for OOD detection. SIREN establishes competitive performance on challenging object-level OOD detection tasks, evaluated broadly under both the recent detection transformers and CNN-based models. Our in-depth ablations provide further insights on the efficacy of SIREN. We hope our work inspires future research on OOD detection with representation shaping.

References

[1] Arindam Banerjee, Inderjit S. Dhillon, Joydeep Ghosh, and Suvrit Sra. Clustering on the unit hypersphere using von mises-fisher distributions. *J. Mach. Learn. Res.*, 6:1345–1382, 2005.

[2] Amir Bar, Xin Wang, Vadim Kantorov, Colorado J. Reed, Roei Herzig, Gal Chechik, Anna Rohrbach, Trevor Darrell, and Amir Globerson. Detreg: Unsupervised pretraining with region priors for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2022*, 2022.

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

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *16th European Conference on Computer Vision, ECCV 2020*, volume 12346, pages 213–229, 2020.

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *IEEE/CVF International Conference on Computer Vision, ICCV 2021*, pages 9630–9640, 2021.

[6] Xingyu Chen, Xuguang Lan, Fuchun Sun, and Nanning Zheng. A boundary based out-of-distribution classifier for generalized zero-shot learning. In *16th European Conference on Computer Vision, ECCV 2020*, volume 12369, pages 572–588, 2020.

[7] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014*, pages 3606–3613, 2014.

[8] Tim R. Davidson, Luca Falorsi, Nicola De Cao, Thomas Kipf, and Jakub M. Tomczak. Hyperspherical variational auto-encoders. In Amir Globerson and Ricardo Silva, editors, *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI 2018*, pages 856–865, 2018.

[9] Kumari Deepshikha, Sai Harsha Yelleni, P. K. Srijith, and C. Krishna Mohan. Monte carlo dropblock for modelling uncertainty in object detection. *CoRR*, abs/2108.03614, 2021.

[10] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019*, pages 4690–4699, 2019.

[11] Akshay Raj Dhamija, Manuel Günther, Jonathan Ventura, and Terrance E. Boult. The overlooked elephant of object detection: Open set. In *IEEE Winter Conference on Applications of Computer Vision, WACV 2020*, pages 1010–1019, 2020.

[12] Xuefeng Du, Xin Wang, Gabriel Gozum, and Yixuan Li. Unknown-aware object detection: Learning what you don't know from videos in the wild. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[13] Xuefeng Du, Zhaoning Wang, Mu Cai, and Yixuan Li. Vos: Learning what you don't know by virtual outlier synthesis. *Proceedings of the International Conference on Learning Representations*, 2022.

[14] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

[15] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059, 2016.

[16] Siddharth Gopal and Yiming Yang. Von mises-fisher clustering models. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014*, volume 32, pages 154–162, 2014.

[17] Akshita Gupta, Sanath Narayan, KJ Joseph, Salman Khan, Fahad Shahbaz Khan, and Mubarak Shah. Ow-detr: Open-world detection transformer. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2022*, 2022.

[18] David Hall, Feras Dayoub, John Skinner, Haoyang Zhang, Dimity Miller, Peter Corke, Gustavo Carneiro, Anelia Angelova, and Niko Sünderhauf. Probabilistic object detection: Definition and evaluation. In *IEEE Winter Conference on Applications of Computer Vision, WACV 2020*, pages 1020–1029, 2020.

[19] Md. Abul Hasnat, Julien Bohné, Jonathan Milgram, Stéphane Gentric, and Liming Chen. von mises-fisher mixture model-based deep learning: Application to face verification. *CoRR*, abs/1706.04264, 2017.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, pages 770–778, 2016.

[21] Dan Hendrycks, Steven Basart, Mantas Mazeika, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. A benchmark for anomaly segmentation. *CoRR*, abs/1911.11132, 2019.

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

[23] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations*, 2019.

[24] Yen-Chang Hsu, Yilin Shen, Hongxia Jin, and Zsolt Kira. Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10951–10960, 2020.

[25] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[26] Jyh-Jing Hwang, Stella X. Yu, Jianbo Shi, Maxwell D. Collins, Tien-Ju Yang, Xiao Zhang, and Liang-Chieh Chen. Segsort: Segmentation by discriminative sorting of segments. In *IEEE/CVF International Conference on Computer Vision, ICCV 2019,*, pages 7333–7343, 2019.

[27] K. J. Joseph, Salman Khan, Fahad Shahbaz Khan, and Vineeth N. Balasubramanian. Towards open world object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2021*, 2021.

[28] Takumi Kobayashi. T-vmf similarity for regularizing intra-class feature distribution. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021*, pages 6616–6625, 2021.

[29] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical Report, 2009.

[30] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4. *International Journal of Computer Vision*, pages 1–26, 2020.

[31] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pages 6402–6413, 2017.

[32] Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *International Conference on Learning Representations*, 2018.

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

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

[35] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755, 2014.

[36] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems*, 2020.

[37] Prasanta Chandra Mahalanobis. On the generalized distance in statistics. National Institute of Science of India, 1936.

[38] Kanti V Mardia, Peter E Jupp, and KV Mardia. *Directional statistics*, volume 2. Wiley Online Library, 2000.

[39] Dimity Miller, Feras Dayoub, Michael Milford, and Niko Sünderhauf. Evaluating merging strategies for sampling-based uncertainty techniques in object detection. In *International Conference on Robotics and Automation, ICRA 2019*, pages 2348–2354, 2019.

[40] Dimity Miller, Lachlan Nicholson, Feras Dayoub, and Niko Sünderhauf. Dropout sampling for robust object detection in open-set conditions. In *2018 IEEE International Conference on Robotics and Automation, ICRA 2018*, pages 1–7, 2018.

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

[42] Julia Nitsch, Masha Itkina, Ransalu Senanayake, Juan I. Nieto, Max Schmidt, Roland Siegwart, Mykel J. Kochenderfer, and Cesar Cadena. Out-of-distribution detection for automotive perception. In *24th IEEE International Intelligent Transportation Systems Conference, ITSC 2021*, pages 2938–2943, 2021.

[43] Junyoung Park, Subin Yi, Yongseok Choi, Dong-Yeon Cho, and Jiwon Kim. Discriminative few-shot learning based on directional statistics. *CoRR*, abs/1906.01819, 2019.

[44] Ilija Radosavovic, Raj Prateek Kosaraju, Ross B. Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020*, pages 10425–10433, 2020.

[45] Jie Ren, Stanislav Fort, Jeremiah Liu, Abhijit Guha Roy, Shreyas Padhy, and Balaji Lakshminarayanan. A simple fix to mahalanobis distance for improving near-ood detection. *CoRR*, abs/2106.09022, 2021.

[46] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015.

[47] Chandramouli Shama Sastry and Sageev Oore. Detecting out-of-distribution examples with gram matrices. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020*, pages 8491–8501, 2020.

[48] Tyler R. Scott, Andrew C. Gallagher, and Michael C. Mozer. von mises-fisher loss: An exploration of embedding geometries for supervised learning. In *IEEE/CVF International Conference on Computer Vision, ICCV 2021*, pages 10592–10602. IEEE, 2021.

[49] Suvrit Sra. A short note on parameter approximation for von mises-fisher distributions: and a fast implementation of $I_s(x)$. *Comput. Stat.*, 27(1):177–190, 2012.

[50] Bo Sun, Banghuai Li, Shengcai Cai, Ye Yuan, and Chi Zhang. FSCE: few-shot object detection via contrastive proposal encoding. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021*, pages 7352–7362, 2021.

[51] Xin Sun, Zhenning Yang, Chi Zhang, Keck Voon Ling, and Guohao Peng. Conditional gaussian distribution learning for open set recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020*, pages 13477–13486, 2020.

[52] Yiyou Sun, Chuan Guo, and Yixuan Li. React: Out-of-distribution detection with rectified activations. In *Advances in Neural Information Processing Systems*, 2021.

[53] Jihoon Tack, Sangwoo Mo, Jongheon Jeong, and Jinwoo Shin. Csi: Novelty detection via contrastive learning on distributionally shifted instances. In *Advances in Neural Information Processing Systems*, 2020.

[54] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *14th European Conference on Computer Vision, ECCV 2016*, volume 9911, pages 499–515, 2016.

[55] Aming Wu, Yahong Han, Linchao Zhu, and Yi Yang. Universal-prototype enhancing for few-shot object detection. In *IEEE/CVF International Conference on Computer Vision, ICCV 2021*, pages 9547–9556, 2021.

[56] Yongyu Xiong, Peipei Yang, and Cheng-Lin Liu. One-stage open set object detection with prototype learning. In *28th International Conference on Neural Information Processing, ICONIP 2021*, volume 13108, pages 279–291, 2021.

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

[58] Hong-Ming Yang, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. Robust classification with convolutional prototype learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018*, pages 3474–3482, 2018.

[59] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. BDD100K: A diverse driving dataset for heterogeneous multitask learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020*, pages 2633–2642, 2020.

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

[61] Xuefei Zhe, Shifeng Chen, and Hong Yan. Directional statistics-based deep metric learning for image classification and retrieval. *Pattern Recognit.*, 93:113–123, 2019.

[62] Bolei Zhou, Àgata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions Pattern Analysis and Machine Intelligence.*, 40(6):1452–1464, 2018.

[63] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: deformable transformers for end-to-end object detection. In *9th International Conference on Learning Representations, ICLR 2021*, 2021.

[64] Honglei Zhuang, Chi Wang, Fangbo Tao, Lance M. Kaplan, and Jiawei Han. Identifying semantically deviating outlier documents. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017*, pages 2748–2757, 2017.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section A in the appendix.
   (d) Did you describe the limitations of your work? [N/A]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A] We are very much in support of open-source. Code will be released for reproducible research.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4 and Appendix B and E.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See main results in Table 1.
   (d) Did you include the amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix J
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Section 4
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

SIREN: Shaping Representations for OOD Detection (Appendix)

A Broad Impact

Our project aims to improve the reliability and safety of modern machine learning models. Our study can lead to direct benefits and societal impacts, particularly for safety-critical applications such as autonomous driving. Our study does not involve any human subjects or violation of legal compliance. We do not anticipate any potentially harmful consequences to our work. Through our study and releasing our code, we hope to raise stronger research and societal awareness towards the problem of out-of-distribution detection in real-world settings.

B Experimental Details

Following [13], we summarize the OOD detection evaluation task in Table 4. The OOD test dataset is selected from MS-COCO and OPENIMAGES dataset, which contains disjoint labels from the respective ID dataset. Our SIREN is trained for a total of 50 epochs on PASCAL-VOC, and trained for 30 epochs on BDD100K using ADAM optimizer. The initial learning rate is 2e-4 and decays at epoch 40 and 24 for PASCAL-VOC and BDD100K dataset, respectively. We set the number of the object queries as 300, the batch size as 8, and the weight β in Equation (8) as 1.5. See *detailed ablations on the hyperparameters on a validation OOD dataset in Appendix E*.

	Task 1	**Task 2**
ID train dataset	VOC train	BDD train
ID val dataset	VOC val	BDD val
OOD dataset	COCO & OPENIMAGES val	COCO and OPENIMAGES val
#ID train images	16,551	69,853
#ID val images	4,952	10,000
#OOD images for COCO	930	1,880
#OOD images for OPENIMAGES	1,761	1,761

Table 4: OOD detection evaluation tasks.

C In-distribution Classes

PASCAL-VOC consists of the following ID labels: Person, Car, Bicycle, Boat, Bus, Motorbike, Train, Airplane, Chair, Bottle, Dining Table, Potted Plant, TV, Sofa, Bird, Cat, Cow, Dog, Horse, Sheep.

BDD100K consists of ID labels: Pedestrian, Rider, Car, Truck, Bus, Train, Motorcycle, Bicycle, Traffic light, Traffic sign.

D Proof for Estimating $\widehat{\kappa}$

We provide the mathematical proof for the estimation of κ in Equation (12). Concretely, we frame the estimation problem as deriving maximum likelihood estimators (MLEs) for the vMF density function, given the training data $\{\mathbf{r}_i\}_{i=1}^{M}$. Specifically, the maximum likelihood learning can be written as the following optimization problem:

$$\begin{aligned} &\max_{\boldsymbol{\mu},\kappa} \mathcal{L}_{\text{MLE}}(\boldsymbol{\mu},\kappa) := \sum_{i=1}^{M} \log p\left(\mathbf{r}_i\right) = M\kappa\boldsymbol{\mu}^\top\overline{\mathbf{r}} + M\log Z_d(\kappa) \\ &\text{s.t. } \|\boldsymbol{\mu}\| = 1 \text{ and } \kappa \geq 0, \end{aligned} \tag{13}$$

where $\overline{\mathbf{r}} = \frac{1}{M}\sum_{i=1}^{M}\mathbf{r}_i$ and $\mathcal{L}_{\text{MLE}}$ is the optimization objective of the maximum likelihood estimation for the distributional parameters $\boldsymbol{\mu}, \kappa$. In order to minimize this objective, we take the derivatives of

the objective *w.r.t* the parameters $\boldsymbol{\mu}, \kappa$ and set them to 0. Then, the optimal parameters $\boldsymbol{\mu}^*, \widehat{\kappa}$ should satisfy the following two conditions:

$$\boldsymbol{\mu}^* = \frac{\bar{\mathbf{r}}}{\|\bar{\mathbf{r}}\|}, \quad \frac{Z'_d(\widehat{\kappa})}{Z_d(\widehat{\kappa})} = -\|\bar{\mathbf{r}}\|, \tag{14}$$

where $Z'_d(\widehat{\kappa}) = \frac{dZ_d(\widehat{\kappa})}{d\widehat{\kappa}}$.

For the second condition in Equation (14), we let $\xi = (2\pi)^{d/2}$ and $s = d/2 - 1$ for notation simplicity. Put them into the Equation (2), we get the following:

$$Z_d(\widehat{\kappa}) = \frac{\widehat{\kappa}^s}{\xi \cdot I_s(\widehat{\kappa})}, \quad Z'_d(\widehat{\kappa}) = \frac{1}{\xi} \cdot \frac{s\widehat{\kappa}^{s-1} I_s(\widehat{\kappa}) - \widehat{\kappa}^s I'_s(\widehat{\kappa})}{I_s(\widehat{\kappa})^2}. \tag{15}$$

Divide $Z_d(\widehat{\kappa})$ by $Z'_d(\widehat{\kappa})$, we get:

$$\frac{Z'_d(\widehat{\kappa})}{Z_d(\widehat{\kappa})} = \frac{s}{\widehat{\kappa}} - \frac{I'_s(\widehat{\kappa})}{I_s(\widehat{\kappa})}. \tag{16}$$

Note that the Bessel function holds a recursive property, which is given as follows:

$$\frac{I'_s(\widehat{\kappa})}{I_s(\widehat{\kappa})} = \frac{s}{\widehat{\kappa}} + \frac{I'_{s+1}(\widehat{\kappa})}{I_s(\widehat{\kappa})}. \tag{17}$$

Substitute $\frac{I'_s(\widehat{\kappa})}{I_s(\widehat{\kappa})}$ in Equation (16) with the formula in Equation (17) and integrate it with the second condition of Equation (14), we get:

$$\frac{I_{d/2}(\widehat{\kappa})}{I_{d/2-1}(\widehat{\kappa})} = \|\bar{\mathbf{r}}\|. \tag{18}$$

Since there is no known closed-form solution to the Bessel ratio inversion problem as shown in the formula above, we adopt the approximation schemes based on the continued fraction form of the Bessel ratio function [1], namely:

$$R := \frac{I_{d/2}(\widehat{\kappa})}{I_{d/2-1}(\widehat{\kappa})} = \frac{1}{\frac{d}{\widehat{\kappa}} + \frac{1}{\frac{d+2}{\widehat{\kappa}} + \cdots}} \approx \frac{1}{\frac{d}{\widehat{\kappa}} + R}. \tag{19}$$

Replace R with $\|\bar{\mathbf{r}}\|$, we get $\widehat{\kappa} \approx \frac{d \cdot \|\bar{\mathbf{r}}\|}{1 - \|\bar{\mathbf{r}}\|^2}$. Following [1], we further add a correction term $-\|\bar{\mathbf{r}}\|^3$ to the numerator and we get the approximated estimation calculated as:

$$\widehat{\kappa} = \frac{\|\bar{\mathbf{r}}\|(d - \|\bar{\mathbf{r}}\|^2)}{1 - \|\bar{\mathbf{r}}\|^2}. \tag{20}$$

E Hyperparameter Analysis

Below we perform sensitivity analysis for each important hyperparameter. Our sensitivity analysis uses the speckle-noised PASCAL-VOC validation dataset as OOD data, which is different from the actual OOD test datasets in use. We use SIREN pretrained with DINO [5] as the object detection backbone, trained on in-distribution dataset PASCAL-VOC.

Effect of the hypersphere dimension d. SIREN projects the object feature embeddings into a lower-dimensional hypersphere in $\mathbb{R}^d$, which allows tractable vMF estimation. A reasonable choice of the hyperspherical dimension d is able to preserve the sufficient information for OOD detection while avoiding distributional parameter estimation in the high-dimension space. As shown in Table 5, using a dimension d between 16 and 64 yields a desirable and stable performance on the OOD validation data while properly maintaining the ID performance (mAP). We set $d = 16$ for PASCAL-VOC and 64 for BDD100K in Table 1.

Effect of the SIREN loss weight β. In Table 6, we show the sensitivity of the OOD detection performance of our SIREN *w.r.t* the weight β of the representation shaping loss. Overall, we find that $\beta = 1.5$ achieves the best OOD detection and ID performance.

d	mAP↑	FPR95 ↓	AUROC↑
8	60.4	85.50	59.64
16	**60.8**	**82.08**	**68.42**
32	58.1	83.17	67.04
64	58.7	83.04	66.68
80	58.2	88.09	67.41

Table 5: Ablation study on the dimension of the hypersphere d.

β	mAP↑	FPR95 ↓	AUROC↑
0.1	60.2	85.16	66.58
0.5	59.2	88.37	61.25
1.0	59.8	86.77	64.59
1.5	**60.8**	**82.08**	**68.42**
2.0	58.9	83.73	67.74
2.5	56.0	84.76	65.66

Table 6: Ablation study on the loss weight β for $\mathcal{L}_{\text{SIREN}}$.

F Baselines

To evaluate the baselines, we follow the original methods in MaxLogit [21], MSP [22], ODIN [34], Generalized ODIN [24] and CSI [53] and apply them accordingly on the classification branch of the object detectors. The Mahalanobis distance [33] and gram matrices [47] are calculated based on the penultimate-layer features of the decoder in DDETR. For ODIN [34], the temperature is set to be $T = 1000$ for BDD100K and $T = 0.1$ for PASCAL-VOC. For CSI [53], we use the rotation degree prediction (0°, 90°, 180°, 270°) as the self-supervised task. We set the temperature in the contrastive loss to 0.5. We use the penultimate-layer features of the decoder (with the dimension to be 256) to perform contrastive learning. The weights of the losses that are used for classifying shifted instances and instance discrimination are both set to 1 following the original paper [53]. For Generalized ODIN [24], we replace and train the final classification head of the object detector by the most effective Deconf-C head shown in the original paper. For OW-DETR [17], we follow the original paper and utilize the logit (after sigmoid) of the additional unknown class for OOD detection. For VOS [13], we use the same hyperparameters as those in the original paper and synthesize virtual outliers in the penultimate layer of the object detector. Then we regard the virtual outliers as the negative samples during the object classification.

G Comparison of Training Time

Method	Training time (h)
ID: PASCAL-VOC / BDD100K	
MaxLogit [21]	9.7 / 27
MSP [22]	9.7 / 27
ODIN [34]	9.7 / 27
Mahalanobis [33]	9.7 / 27
Gram matrices [47]	9.7 / 27
Generalized ODIN [24]	10.0 / 27.9
CSI [53]	17.1 / 47.9
VOS [13]	11.4 / 32.7
OW-DETR [17]	23.7 / 59.3
SIREN (ours)	10.1 / 27.7

Table 7: Comparison of the training time for different baselines in Table 1 of the main paper.

We provide the comparison of the training time for various baselines in Table 7 on our reported hardware (Section J in the Appendix). As the table shows, the training of our SIREN incurs minimal computational overhead compared to the vanilla DDETR. In contrast, other baselines such as OW-DETR can be more than 2 times slower than SIREN.

H More Visualizations

In Figure 6, we show that SIREN can improve the embedding quality in the penultimate layer as a result of shaping the hyperspherical embeddings. Quantitatively, we compare the performance of using Mahalanobis distance [33] for OOD detection, on a vanilla DDETR vs. DDETR trained with SIREN. In both cases, the Mahalanobis distance is derived from the same penultimate layer. On PASCAL-VOC, SIREN improves the AUROC from 50.28% to 56.87% (MS-COCO as OOD), and from 49.08% to 52.66% (OPENIMAGES as OOD). Nonetheless, the performance of Mahalanobis is still inferior to ours—using the vMF score on the hyperspherical embeddings directly.

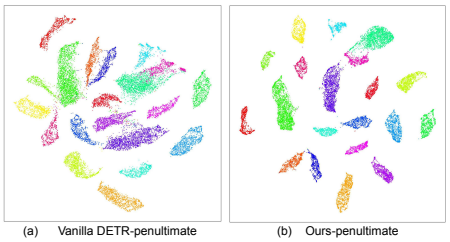


Figure 6: Comparison of the penultimate-layer feature embeddings for the vanilla DDETR (a) and SIREN (b).

I Details of Toy Example

For the Figure 3 in the main paper, we generate the toy data in the unit hypersphere by sampling from three vMF distributions in the 3D space. We adopt a concentration parameter κ of 100 for all the three distributions. The centroid vectors are set to $[0, 0, 1]$, $[\frac{\sqrt{3}}{2}, 0, -\frac{1}{2}]$ and $[-\frac{\sqrt{3}}{2}, 0, -\frac{1}{2}]$, respectively. The uncertainty surface is obtained by calculating the uncertainty score of 200^2 points in the surface of the 3D ball.

J Software and Hardware

We run all experiments with Python 3.8.5 and PyTorch 1.7.0, using 8 NVIDIA GeForce RTX 2080Ti GPUs.