Enhancing Out-of-Distribution Detection through Generating Distance-Based Virtual Outliers

Juyoung Choi[1] and Kibok Lee[1]

[1] Department of Statistics and Data Science, Yonsei University, Seoul, South Korea, {juchoi, kibok}@ yonsei.ac.kr

Abstract

Detecting out-of-distribution (OOD) data is essential for the safe deployment of machine learning models. However, acquiring outliers from the real world for training poses significant challenges, as it is impossible to account for all potential outliers. We propose Distance-based Virtual Outlier generation (DiVO), a method that generates virtual outliers in the feature space by leveraging the distances between in-distribution (ID) samples within the same class. We employ supervised contrastive training to encourage learning an isotropic feature distribution, ensuring that the generated outliers are positioned just beyond the decision boundaries between ID and OOD. Unlike previous methods, our approach does not rely on any distributional assumptions for generating virtual outliers. We validate the effectiveness of DiVO through extensive benchmarks and experimental settings, demonstrating its superiority in OOD detection.

Keywords— *Out-of-distribution detection, Supervised contrastive training, Virtual outlier generation*

I. Introduction

Recent studies have shown that machine learning models exhibit excellent performance under the assumption that the distribution of training data, known as in-distribution (ID), is equal to the distribution of test data. However, in real-world scenarios, it is highly likely to encounter samples from different distributions, referred to as out-of-distribution (OOD) data. To address this, OOD detection methods have been proposed for classifiers that can identify OOD data and output *unknown* results.

The major challenge of OOD detection is that the model produces overconfident predictions for OOD samples [15]. In the case of autonomous driving, it is important to produce reliable results to avoid mis-predictions in safety-critical situations. One of the approaches among existing methods is outlier exposure (OE) [7]. This approach modifies the training process to include a loss term that trains the classifier to output a uniform distribution for outlier inputs. However, the performance of this method depends on the selection of auxiliary datasets, and the auxiliary data should not overlap with the ID data, which requires a high computational cost to prepare. There are methods for generating outliers from ID data that do not require additional outlier data, but these methods require a strong assumption for distribution of features in the feature space which do not always hold [3]. This leads to the question: *Can we use the distance to generate outliers close to the boundaries of the ID samples without any distribution assumptions?*

In this work, we propose a method coined DiVO (Distance-based Virtual Outliers generation) that generates virtual outliers in the feature space by leveraging the distances between in-distribution samples within the same class. To generate outliers, we start by estimating the class-wise mean vectors and the empirical covariance matrix in the feature space. From the estimated class mean, we identify the point within the class with the largest L2 distance, compute the Mahalanobis distance to this point and use it to generate outlier vectors. Here, supervised contrastive learning [8] is used because distance can be utilized effectively only when the features of in-distribution samples are densely positioned for each class in the space. We ensure that contrastive learning produces isotropic representations which are effective in solving the anisotropy of embeddings [4]. During training, we apply the SupCon loss to the ID data to prevent the features from being concentrated on one side of the embedding space. Additionally, a regularization loss is applied to the generated outliers to encourage the model to produce high OOD scores for these outliers while maintaining low scores for the ID data.

Our training procedure, unlike methods that directly utilize outliers, does not require OOD fine-tuning and is effective. Moreover, we demonstrate that the synergy between supervised contrastive learning and proposed outlier generation enhances OOD detection performance through extensive experiments. In Section ii., we provide background on existing methods. Section iii. introduces our proposed method, and Section iv. presents the experimental results. Finally, we conclude in Section v..

기존 머신러닝 모델들은 학습 데이터와 같이 분포의 테스트 데이터에서는 좋은 성능을 보이지만,
실제 환경에서 마주칠 수 있는 분포외 데이터를 탐지하는 것이 어려움

DiVo

- 특징 공간에서 거리 기반으로 가상 아웃라이어 생성
- 분포 가정 없이 아웃라이어를 생성
- 같은 클래스 = 다른클레스 가중

+ 마할라노비스 거리 —> 클래스별 공분산

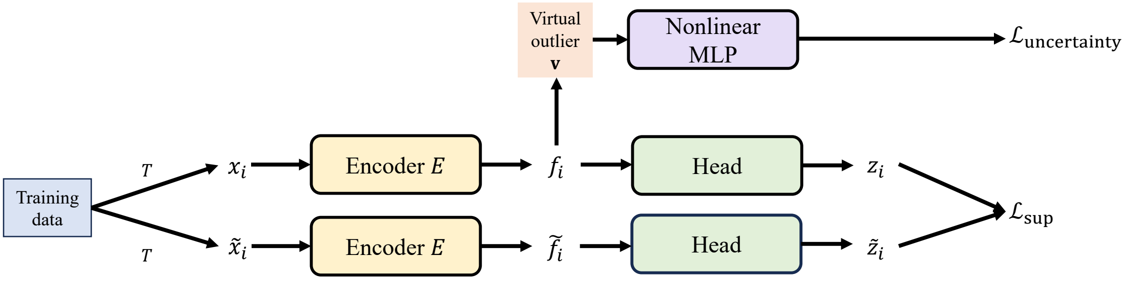


Fig. 1. The overall framework of DiVO. We consider features extracted from an encoder using the training set as input. Virtual outliers generated in the feature space are passed through a nonlinear MLP function where uncertainty loss is applied, while ID features are projected through a head and subjected to supervised contrastive learning loss. The final loss is a weighted sum of these two losses.

II. Related Work

Solutions for OOD detection can generally be divided into two broad approaches: post-hoc solutions and training strategies.

A. Post-hoc OOD Detection Approaches

Post-hoc methods propose scoring functions for OOD detection based on the output of pretrained models. The most widely used scoring function among them is the maximum softmax probability (MSP), which detects by thresholding the highest value among class-wise softmax probabilities [6]. Additionally, methods such as ODIN [11], Mahalanobis distance-based score [10], and energy score [12] exist. Similarly, there are studies focusing on analyzing the activation patterns of models based on the changes in output depending on whether the input is ID or OOD. Particularly, it has been shown that rectifying the activation of the penultimate layer output of trained models aids in distinguishing between ID and OOD [18]. Furthermore, a contribution matrix is defined using the output of the penultimate layer to measure important weights, and only necessary weights are utilized in the fully connected layer [19]. For detection, important layers are selected via Shapley value-based pruning to mitigate the influence of noisy output [1].

Recently, a simple method is proposed that utilizes generalized entropy to capture small deviations better than traditional Shannon entropy, along with reflecting them at scoring, and can be applied to large-scale datasets [13].

B. Training Strategies in OOD Detection Task

One approach in training-based methods involves utilizing OOD data during training. By preparing auxiliary OOD data, these methods employ supervision techniques to train the model to have a uniform distribution for OOD inputs [7]. Additionally, there are methods that fine-tune the model using energy scores, assigning low energy values to ID inputs and high energy values to OOD inputs [12].

In this context, a method has been proposed to utilize virtual outliers instead of using OOD data directly. This can be achieved by employing hard augmentation in contrastive learning [21], assuming that ID features follow a class-conditional multivariate Gaussian distribution and sampling outliers with low likelihood values from this distribution [3]. Alternatively, Mixup [25] techniques can be applied at the feature level [17].

Another approach involves contrastive learning during model training. KNN [20] utilizes a non-parametric k-th nearest neighbors-based score as a post-hoc scoring function. A variant incorporating contrastive learning during model training is denoted as KNN+ [20]. Similarly, SSD [16] trains the model using self-supervised training loss and subsequently employs Mahalanobis distance for detection. SSD+ [16] denotes the version that utilizes supervised contrastive learning loss, leveraging access to labeled data.

III. Methodology

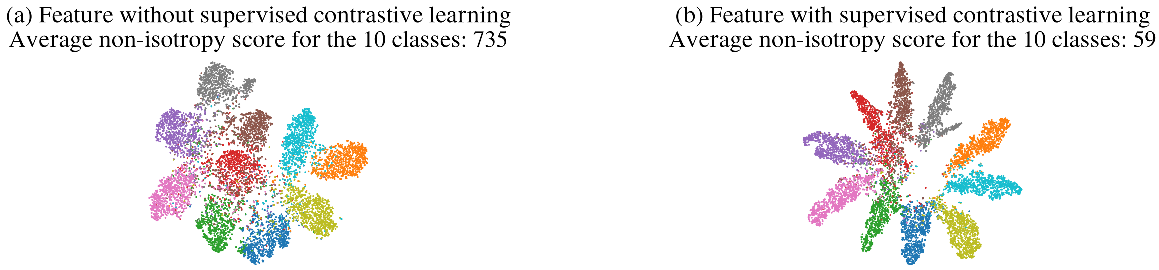
We consider the input data $\mathbf{x} = \{x_1, x_2, ..., x_N\}$ and labels $\mathbf{y} = \{y_1, y_2, ..., y_N\}$. The N features $\mathbf{f} = \{f_1, f_2, ..., f_N\}$ are extracted by an encoder E for input $(\mathbf{x}, \mathbf{y})$ within a batch, $\mathbf{f} = E(\mathbf{x}, \mathbf{y})$. The projection head P maps representation $\mathbf{f}$ to a head feature $\mathbf{z} = \{z_1, z_2, ..., z_N\}$.

Fig. 1 illustrates our framework. DiVO does not require additional OOD data and generates virtual outliers solely from ID training data, thus requiring an outlier generation method, and loss terms to be applied to ID samples and generated outliers. We aim to explain these in Section A. and Section B., respectively.

A. Generating Virtual Outliers

Our method generates outliers using features extracted from an encoder, such as those obtained from a relatively easy to optimize encoder, in lower dimensions than pixel space [3].

We estimate the empirical mean and covariance matrix



(a) Feature without supervised contrastive learning
Average non-isotropy score for the 10 classes: 735

(b) Feature with supervised contrastive learning
Average non-isotropy score for the 10 classes: 59

Fig. 2. t-SNE 2D projection of penultimate features of WideResNet40-2 trained with (a) cross-entropy loss and (b) supervised contrastive (SupCon) loss. Each class is represented in a different color. The average non-isotropy score is calculated by averaging the scores of each class. The high non-isotropy score indicates that the feature is isotropic, which means that features within the same class are more closely grouped together. Comparing the use of SupCon loss, the score is 12 times higher with SunCop loss than without. When using SupCon loss, features are positioned without overlapping. We use SupCon loss to ensure class features are clustered together.

for each class of training samples: 클래스 별 평균 벡터 추정

$$\hat{\mu}_k = \frac{1}{N_k} \sum_{c:y_c=k} f_c, \tag{1}$$

$$\hat{\Sigma} = \frac{1}{N} \sum_k \sum_{c:y_c=k} (f_c - \hat{\mu}_k)(f_c - \hat{\mu}_k)^T, \tag{2}$$

where N_k is the number of samples in class $k \in \{1, 2, \ldots, K\}$. $\hat{\mu}_k$ is the mean vector for class k and $\hat{\Sigma}$ is the tied covariance matrix.

To generate outliers that exist outside the class boundary, we compute the the L2 distance between the mean vector of the k^{th} class and each sample of the k^{th} class, we find the furthest feature index i, $f_i = \underset{f}{\text{argmax}} ||\hat{\mu}_k - f||_2$. Then, in the feature space, we find the mean vector of the j^{th} class, which has the closest L2 distance to the k^{th} class, $\hat{\mu}_j = \underset{\hat{\mu}}{\text{argmin}} ||\hat{\mu}_k - \hat{\mu}||_2$. It is to generate outliers that exist on the boundary of the k^{th} class but do not belong to other classes. We propose generating outliers as follows:

가상

$$\mathbf{v} = \rho \times \frac{\hat{\mu}_k - \hat{\mu}}{||\hat{\mu}_k - \hat{\mu}||_2}, \tag{3}$$

where $\rho = \sqrt{(\hat{\mu}_j - \hat{\mu}_k)^T \Sigma^{-1} (\hat{\mu}_j - \hat{\mu}_k)}$ is the Mahalanobis distance between $\hat{\mu}_k$ and f_i. Since the distribution of features in the feature space is unknown, we measure the Mahalanobis distance to incorporate the covariance matrix into the distance calculation. By multiplying the unit vector between the mean vectors by the Mahalanobis distance, we generate outliers that are close to and also outside the class boundary. These are non-trivial for the model, thus aiding in training the model to perform the OOD detection task [3].

B. Training Objective

Now, we introduce the training objective utilizing the generated outliers. Since we leverage the distances in the feature space, we should ensure the isotropy of the embeddings. To achieve this, we use contrastive learning, employing a supervised contrastive loss term to repel only samples form different classes.

Given N data pairs $\{x_o, y_o\}_{o=1,\ldots,N}$ and N augmented data pairs $\{\tilde{x}_t, \tilde{y}_t\}_{t=1,\ldots,N}$ within a multiviewed batch, let $i \in I \equiv \{1, 2, \ldots, 2N\}$ be an index of an arbitrary augmented sample. As [8] denoted, the supervised contrastive loss term:

$$\mathcal{L}_{\text{sup}} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(z_i \cdot z_p / \tau)}{\sum_{a \in A(i)} \exp(z_i \cdot z_a / \tau)} \tag{4}$$

where $A(i) \equiv I - \{i\}$, $P(i) \equiv \{p \in A(i) : \tilde{y}_p = \tilde{y}_i\}$ and $|P(i)|$ is its cardinality.

By the definition of isotropy, a distribution is isotropic if its variance is uniformly distributed across all dimensions. The covariance matrix of an isotropic distribution is proportional to identity matrix. We compute the Frobenius norm between normalized empirical covariance matrix and identity matrix to ensure the isotropy:

$$||\hat{\Sigma} - \mathbf{I}_d|| = \sqrt{\sum_{i=1}^{d} \sum_{j=1}^{d} a_{ij}^2}, \tag{5}$$

where the empirical covariance matrix $\hat{\Sigma} \in \mathbb{R}^{d \times d}$.

Fig. 2 compares the case trained using the cross-entropy (CE) loss term (a) with the case trained using the supervised contrastive (SupCon) loss term (b). Features are extracted from the penultimate layer of pretrained WideResNet40-2 [24] model using two learning strategies. The dataset used for plotting t-SNE 2D projection is the CIFAR-10. Here, the higher the non-isotropy score, the more isotropic the class-specific distribution. Fig. 2 visualizes contrastive learning gathering samples from the same class intensively, demonstrating that leveraging distance is effective, as evidenced by the non-isotropy score, Equation (5).

To effectively utilize the generated outliers, we apply a loss term based on energy, which is an effective measurement for OOD detection. As mentioned in [3], the loss term

라벨
클레스 , 샘플

Algorithm 1 DiVO training algorithm

```
Require: ID data-label pairs $O = \{(x_1, y_1), \cdots, (x_N, y_N)\}$, augmented ID data-label pairs $A = \{(\tilde{x}_1, \tilde{y}_1), \cdots, (\tilde{x}_N, \tilde{y}_N)\}$, $C_i$ is the $i^{\text{th}}$ class, $i = 1, 2, \cdots, K$, $N_i$ = the number of samples for the $i^{\text{th}}$ class, $N$ = the total number of samples, $N = \sum_{i=1}^{k} n_i$, weight $\beta$.
  while train do
    for $i = 1, 2, \cdots, K$ do
      Estimate the empirical class mean and covariance matrix using the Equation (1) and (2)
    end for
    Find the sample which is from $C_i$ and has the maximum L2 distance from class mean
    for $i = 1, 2, \cdots, K$ do
      Calculate the Mahalanobis distance $\rho$ between $\mu_{C_i}$ and the sample
      Find the other class which is has the minimum L2 distance from class mean
      Generate $\mathbf{v}$ using the Equation (3)
    end for
  end while
```

serving as an uncertainty regularization assigns low energy values to ID data and high energy values to outliers, and is defined as follows:

$$\mathscr{L}_{\text{uncertainty}} = \mathbb{E}_{\mathbf{v}}\left[-\log \frac{1}{1+\exp^{-\phi(E(\mathbf{v};f))}}\right] + \mathbb{E}_{\mathbf{x}}\left[-\log \frac{\exp^{-\phi(E(\mathbf{x};f))}}{1+\exp^{-\phi(E(\mathbf{x};f))}}\right] \tag{6}$$

where ϕ is a nonlinear MLP function and $E(\mathbf{x}; f) = -\log \sum_{k=1}^{K} e^{f_k(\mathbf{x})}$ is the free energy function defined in [12]. Previous research using energy for uncertainty regularization was carried out based on the hinge loss function and required adjustment for hyperparameters [12], but this loss has the advantage of being used simply without hyperparameters.

The total loss to be applied to both ID samples and virtual outliers is as follows:

$$\mathscr{L} = \mathscr{L}_{\text{sup}} + \beta \cdot \mathscr{L}_{\text{uncertainty}}, \tag{7}$$

where β is a hyperparameter, which is the weight of uncertainty loss. Algorithm 1 shows the DiVO training algorithm.

C. *Inference Time in OOD Detection*

At inference time of OOD detection, it is determined whether the input image $\mathbf{x}$ is ID or OOD. The detector D classifies based on the OOD score:

$$D(\mathbf{x}) = \begin{cases} 1, & \text{if } g(\mathbf{x}) \geq \delta, \\ 0, & \text{if } g(\mathbf{x}) < \delta. \end{cases} \tag{8}$$

The threshold δ is chosen so that the true positive rate, which is the fraction of ID images classified as ID, is 95%. g refers to the scoring function, and we used the function that calculates the OOD uncertainty score with energy proposed in [3].

IV. Experiments

A. *Experimental Settings*

The experiments were conducted with CIFAR-10 and CIFAR-100 datasets [9] and ResNet18 [5] as a backbone. We train the model for DiVO based on original settings in [8], a cosine annealing scheduler, and an SGD optimizer. TwoCropTransform is applied to ID data as the augmentation to generate views in contrastive learning, as used in the original setup. Models are trained with SupCon loss which has the temperature $\tau = 0.1$. Batch size is 512 for CIFAR-10 and 256 for CIFAR-100. The dimension of the penultimate layer output for the outlier generation with features is 512 and the dimension of the projection layer is 128. The training is conducted for 500 epochs and 50 epochs for the linear evaluation. We apply warm-up epochs of $\{0, 100, 200, 300, 400, 500\}$, during which the model was trained only with ID samples without exposing it to virtual outliers. We use an energy scoring function for evaluation. Further results, employing WideResNet with 40 layers and a widening factor of 2 as the encoder, are provided.

The evaluation is conducted with Textures [2], SVHN [14], Places365 [26], LSUN-C [23], LSUN-resize [23], and iSUN [22].

We use evaluation metrics commonly employed in OOD detection: FPR95, the false positive rate of OOD samples when the true positive rate is set to 95%, and AUROC, the area under the receiver operating characteristic curve.

We refer to our method in two ways: DiVO$_{\text{CE}}$ and DiVO. DiVO$_{\text{CE}}$ is a version that optimizes the model using CE loss and applies regularization loss to both the features of the outliers generated from the penultimate layer and ID data. DiVO, on the other hand, optimizes the model using SupCon loss instead of CE loss, while also generating an outlier. In other words, the difference between the two terms is whether or not contrastive learning is used.

We compare MSP [6], ODIN [11], Energy [12], Mahalanobis [10], VOS [3], KNN [20] and DiVO$_{\text{CE}}$, which are methods that do not use contrastive learning. Simultaneously, CSI [21], SSD+ [16], and KNN+ [20], which are methods that use contrastive learning, are compared with DiVO.

B. *Results*

In the experiments, we compare the performance of various OOD detection methods. Table 1 and 2 show the OOD detection performance on the CIFAR-10 and CIFAR-100 datasets. The values shown in the tables are

Method	SVHN		LSUN		iSUN		Textures		Places365		Average	
	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑
MSP	59.66	91.25	45.21	93.80	54.57	92.12	66.45	88.50	62.46	88.64	57.67	90.86
ODIN	53.78	91.30	10.93	97.93	28.44	95.51	55.59	89.47	43.40	90.98	38.43	93.04
Energy	54.41	91.22	10.19	98.05	27.52	95.59	55.23	89.37	42.77	91.02	38.02	93.05
Mahalanobis	9.24	97.80	67.73	73.61	6.02	98.63	23.21	92.91	83.50	69.56	37.94	86.50
VOS	18.94	96.44	22.84	95.42	29.63	94.31	46.25	89.82	41.31	90.20	31.79	93.24
KNN	27.97	95.48	18.50	96.84	24.68	95.52	26.74	94.96	47.84	89.93	29.15	94.55
DiVO$_{\text{CE}}$ (ours)	9.31	98.28	15.56	97.41	29.62	95.39	35.22	93.69	40.36	92.03	**26.01**	**95.36**
CSI	37.38	94.69	5.88	98.86	10.36	98.01	28.85	94.87	38.31	93.04	24.16	95.89
SSD+	1.51	99.68	6.09	98.48	33.60	95.16	12.98	97.70	28.41	94.72	16.52	97.15
KNN+	2.42	99.52	1.78	99.48	20.06	96.74	8.09	98.56	23.02	95.36	11.07	97.93
DiVO (ours)	4.95	98.93	0.98	99.47	6.18	98.49	10.68	97.74	17.71	96.46	**8.10**	**98.22**

Table 1. Comparison with state-of-the-art methods on CIFAR-10. All values are percentages. SVHN, LSUN, iSUN, Textures, and Places365 are OOD datasets used in test-time, and Average refers to the mean performance across these five OOD datasets. ↑ indicates that a higher value is better, whereas ↓ signifies that a lower value is better. **Bold** text denotes superior performance, and underlined text indicates second-best performance. The line beneath DiVO$_{\text{CE}}$ separates methods that do not use contrastive learning (above) from those that do (below).

Method	SVHN		LSUN		iSUN		Textures		Places365		Average	
	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑
MSP	78.89	79.80	83.47	75.28	84.61	76.51	86.51	72.53	84.38	74.21	83.57	75.67
ODIN	70.16	84.88	76.36	80.10	79.54	79.16	85.28	75.23	82.16	75.19	78.70	78.91
Energy	66.91	85.25	84.49	59.77	66.52	84.49	79.01	79.96	81.41	76.37	75.67	77.17
Mahalanobis	87.09	80.62	84.15	79.43	83.18	78.83	61.72	84.87	84.63	73.89	80.15	79.53
VOS	61.96	87.68	78.42	79.14	69.05	85.99	82.56	78.03	85.35	72.42	75.47	80.65
KNN	60.97	84.20	71.40	78.85	71.87	81.90	70.30	81.32	78.95	76.89	**70.70**	80.63
DiVO$_{\text{CE}}$ (ours)	79.75	78.52	52.15	91.51	83.59	78.35	79.75	78.52	90.30	72.92	76.21	**81.18**
CSI	44.53	92.65	75.58	83.78	76.62	84.98	61.61	86.47	79.08	76.27	67.48	84.83
SSD+	16.66	96.96	44.65	91.98	77.05	83.88	44.21	90.98	74.48	79.47	**51.41**	**88.65**
KNN+	37.26	93.12	57.97	85.63	71.58	82.48	49.60	89.10	75.53	78.44	58.39	85.75
DiVO (ours)	67.45	87.34	21.61	96.4	63.09	88.32	74.48	81.21	74.54	79.78	60.23	86.61

Table 2. Comparison with state-of-the-art methods on CIFAR-100. All values are percentages.

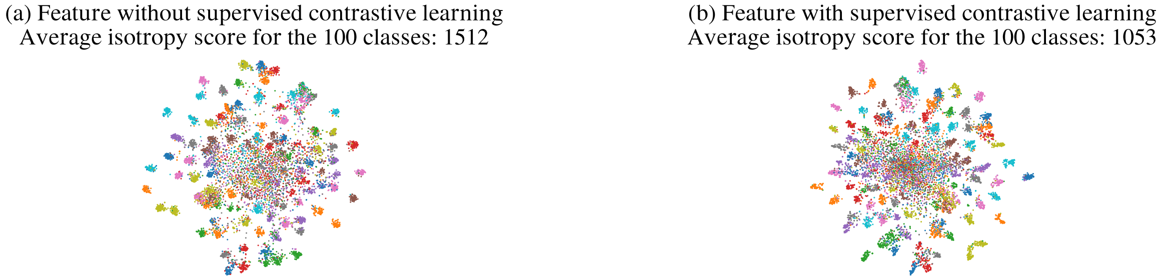
the FPR and AUROC for each OOD test dataset, with the last column showing the average values across the five datasets. For easy comparison, a dividing line is drawn in the center of tables. The top section contains methods that do not use contrastive learning, while the bottom section contains methods that use. MSP, ODIN, Energy, Mahalanobis, KNN, and our DiVO$_{\text{CE}}$ are methods that use CE loss in training. All methods used for comparison are trained for 500 epochs using the ResNet18 backbone with CIFAR benchmarks. Values in bold represent the best performance among the comparison groups, and underlined values indicate the second-best performance. In Table 1, DiVO$_{\text{CE}}$ and DiVO show the best performance. Both methods outperform state-of-the-art approaches by reducing FPR rate by 3. This demonstrates the benefit of leveraging distance-based virtual outliers both with and without contrastive learning. In particular, we focus on the comparison with VOS, a method of sampling virtual outliers with low likelihood for the estimated distribution of the ID data. Since both VOS and DiVO$_{\text{CE}}$ use CE loss-based virtual outliers in training, the superior performance of DiVO$_{\text{CE}}$ indicates that the outliers generated by DiVO$_{\text{CE}}$ are more beneficial for training representation. Fig. 3 shows the reason for the CIFAR-100 performance shown in Table 2. As the number of classes increases, the ability to ensure isotropy through contrastive learning inevitably decreases, which can be seen through 2D projection and the non-isotropy score.

Method	FPR↓	AUROC↑
$\mathcal{L}_{\text{regularization}}$	26.01	95.36
$\mathcal{L}_{\text{sup}}$	10.17	98.13
$\mathcal{L}_{\text{sup}} + \beta\mathcal{L}_{\text{regularization}}$	8.10	98.22

Table 3. Ablation on components of loss function. Values in table are the mean performance across five OOD datasets: SVHN, LSUN, iSUN, Textures, and Places365.

C. *Ablation Study*

Effect of components of loss function. Table 3 reports the results of various changes during training. 1) Without contrastive learning: It is a method of training for the warm-up epoch by applying CE loss to ID data, and training by weighted summing the regularization loss for the subsequently generated outlier. 2) Without outlier generation: It is a method of training for the entire epoch by applying SupCon loss to ID data. At this time, no virtual outlier is generated. 3) Contrastive learning + outlier generation: supervised contrastive learning and outlier generation after warm-up epoch are performed



(a) Feature without supervised contrastive learning
Average isotropy score for the 100 classes: 1512

(b) Feature with supervised contrastive learning
Average isotropy score for the 100 classes: 1053

Fig. 3. t-SNE 2D projection of penultimate features of ResNet18 trained with (a) cross-entropy loss and (b) supervised contrastive loss. Each class is represented in a different color. Comparing the use of SupCon loss, the score is 1.5 times higher with SunCop loss than without. In both cases, the projections of features overlap between classes. This implies that the effect of SupCon loss is minimal on the CIFAR-100 benchmark compared to CIFAR-10.

Distance Metric	**FPR↓**	**AUROC↑**
L1 distance	14.69	97.17
L2 distance	11.71	97.91
Mahalanobis distance	8.10	98.22

Table 4. Ablation on distance metrics of computing ρ. Values in table are the mean performance across five OOD datasets: SVHN, LSUN, iSUN, Textures, and Places365.

	FPR↓	**AUROC↑**
$\beta = 0.05$	10.05	98.09
$\beta = 0.1$	8.10	98.22
$\beta = 0.2$	10.58	98.04
$\beta = 0.3$	10.24	98.07
$\beta = 0.4$	11.96	97.81
$\beta = 0.5$	14.64	97.30

Table 5. Ablation on distance metrics of computing β. Values in table are the mean performance across five OOD datasets: SVHN, LSUN, iSUN, Textures, and Places365.

together in the proposed method. It was experimentally confirmed that distance-based outlier has synergy with contrastive learning.

Effect of distance metric. In Section 3, we defined ρ as the distance between the class-wise sample mean and the feature with the largest L2 distance from the mean within the class, as a parameter for generating virtual outliers in Equation (3). Table 4 presents the results of the ablation experiment on the measuring ρ. We consider L1 distance, L2 distance, and Mahalanobis distance for distance calculation. Experimental results show that employing the Mahalanobis distance improve the FPR rate by 4 and 1 compared to L1 and L2 distance, respectively. Therefore, we chose the Mahalanobis distance as the final method for distance calculation. As indicated by the definition of the Mahalanobis distance, it incorporates the sample covariance matrix into distance computation. This plays a crucial role in our virtual outlier generation process, where no assumptions about the distribution are needed.

	FPR↓	**AUROC↑**
MSP	31.53	95.75
Energy	8.10	98.22

Table 6. Ablation on scoring function g. Values in table are the mean performance across five OOD datasets: SVHN, LSUN, iSUN, Textures, and Places365.

Effect of weight of uncertainty loss. We defined the total loss in Equation (7) as the weighted sum of the SupCon loss and the uncertainty loss. To assess the impact of the uncertainty loss on the total loss, we vary the weight parameter *beta* of the loss over a candidate set of values: $\{0.05, 0.01, 0.2, 0.3, 0.4, 0.5\}$, and examined its effect on performance. It is observed that setting β to excessively high or low values does not consistently yield optimal results. Based on the results in Table 5, we set $\beta = 0.1$ across all experiments.

Effect of OOD scoring function g. In OOD detection at inference time, OOD scoring functions such as MSP and energy functions are commonly applied to distinguish between ID and OOD inputs using the function g defined in Equation (8). Table 6 compares their performance. Particularly noteworthy is the significant performance gap observed between MSP and the average values in Table 1, underscoring the effectiveness of SupCon and the proposed outlier generation in this paper. This reaffirms that the uncertainty loss used during training plays a crucial role in reducing energy values, thereby influencing the scoring function performance.

Effect of training with only ID data during warm-up epochs. Fig. 4 is the result of performing with different numbers of epochs in which the model is trained only with a SupCon loss term before starting the outlier generation. We conclude that it is better to generate outliers after training with ID data for a sufficient number of epochs than

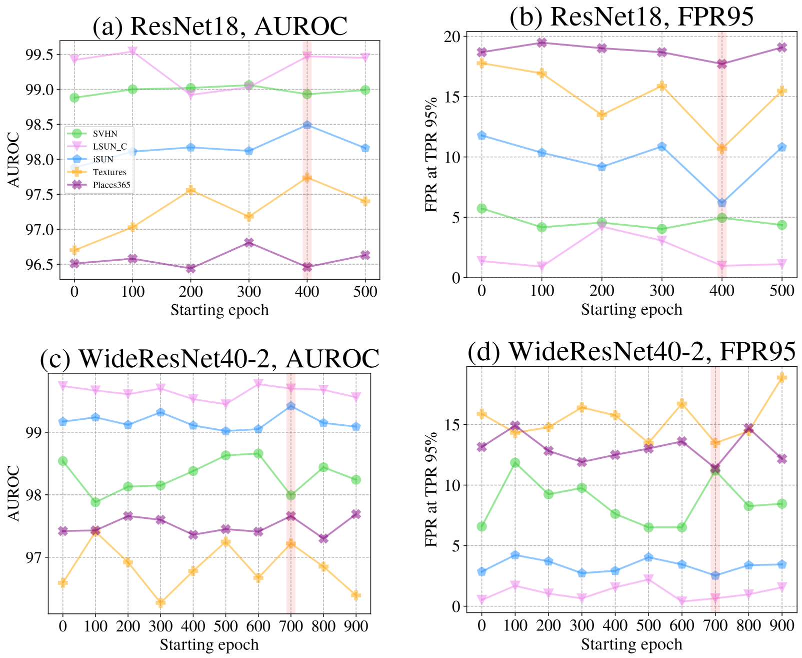


Fig. 4. Effects of training with only ID data during warm-up epochs in (a)(b) ResNet18. (c)(d) WideResNet40-2. This indicates that it is effective to perform outlier generation after sufficiently learning the ID features with ID data alone for 400 and 700 epochs out of a total of 500 and 1000 epochs, respectively.

to generate them form the beginning of training and train with them simultaneously. We use the WideResNet40-2 architecture additionally to show the best performance at different epochs values for each architecture. The model with the WideResNet40-2 architecture is trained for 1000 epochs. A candidate group of warm-up epoch is used as $\{0, 100, 200, 300, 400, 500, 600, 700, 800, 900\}$. As shown in Fig. 4, we achieve the best performance at 400 and 700 epochs for ResNet18 and WideResNet40-2, respectively; we apply the optimal warm-up strategy throughout all experiments.

V. Conclusion

In this paper, we propose DiVO that provides a method for generating a distance-based outlier and a framework for using it for training. The outlier generation takes place on the feature space, which uses the output of the penultimate layer of the model. Here, an outlier is generated by multiplying the unit vector between the nearby classes using the mean for each class and the Mahalanobis distance between the points. Since the outlier generated based on this distance exists outside the class boundary, we apply the SupCon loss to the id data to help with training.

In the experiment part, it was confirmed that DiVO showed superior performance in most benchmarks. In addition, we show that the epoch that starts to generate the outlier can affect the results, and the ablation study proves how much each loss term affects the results by varying the components of our proposed loss term. However, as the number of classes increases, the robustness decreases compared to the existing methodology, which leaves it as a future task to be solved before applying to the large-scale dataset.

References

[1] Yong Hyun Ahn, Gyeong-Moon Park, and Seong Tae Kim. Line: Out-of-distribution detection by leveraging important neurons. In *CVPR*, 2023.

[2] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *CVPR*, 2014.

[3] Xuefeng Du, Zhaoning Wang, Mu Cai, and Yixuan Li. VOS: learning what you don't know by virtual outlier synthesis. In *ICLR*, 2022.

[4] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. In *EMNLP*, 2021.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

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

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

[8] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *NeurIPS*, 2020.

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

[10] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *NeurIPS*, 2018.

[11] Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *ICLR*, 2018.

[12] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. In *NeurIPS*, 2020.

[13] Xixi Liu, Yaroslava Lochman, and Christopher Zach. GEN: pushing the limits of softmax-based out-of-distribution detection. In *CVPR*, 2023.

[14] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NeurIPS workshop*, 2011.

[15] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *CVPR*, 2015.

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

[17] Soroush Seifi, Daniel Olmeda Reino, Nikolay Chumerin, and Rahaf Aljundi. OOD aware supervised contrastive learning. In *WACV*, 2024.

[18] Yiyou Sun, Chuan Guo, and Yixuan Li. React: Out-of-distribution detection with rectified activations. In *NeurIPS*, 2021.

[19] Yiyou Sun and Yixuan Li. DICE: leveraging sparsification for out-of-distribution detection. In *ECCV*, 2022.

[20] Yiyou Sun, Yifei Ming, Xiaojin Zhu, and Yixuan Li. Out-of-distribution detection with deep nearest neighbors. In *ICML*, 2022.

[21] Jihoon Tack, Sangwoo Mo, Jongheon Jeong, and Jinwoo Shin. CSI: novelty detection via contrastive learning on distributionally shifted instances. In *NeurIPS*, 2020.

[22] 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.

[23] 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.

[24] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[25] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.

[26] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2017.

Summary of this paper

A. Problem Setup

In previous machine learning approaches, it achieved good performance on test datasets with the same distribution as the training dataset, known as in-distribution (ID). However, in the real world settings, there is a risk of encountering data not seen during training, so it is necessary to have a classifier that can classify samples not learned during training as ”unknown”. This is the main challenge of the out-of-distribution (OOD) detection task. To address this issue, one of the existing OOD detection approaches proposes using an auxiliary outlier dataset, different from the ID dataset, and exposing the model to it during training. Subsequently, due to the cost associated with preparing an outlier dataset, methods have been proposed to virtually generate outliers from the ID dataset.

B. Novelty

We generate virtual outliers using features extracted from ID samples in the feature space. Existing outlier generation methods often require hard assumptions about the distribution in the feature space or involve stochastic sampling to identify samples that are difficult to classify as ID based on likelihood. In our approach, coined as Distance-based Virtual Outliers generation (DiVO), we generate outliers based on distances without estimating distributions. Specifically, we use Mahalanobis distance, which incorporates class-specific covariances of features, to measure this distance. This represents a novel distance-based outlier generation technique that is beneficial for training models to distinguish OOD samples without relying on distribution assumptions.

C. Algorithms

The algorithm for generating virtual outliers is executed after training the model using only ID data. During this process, supervised contrastive learning is utilized to minimize distances between embeddings of the same class and maximize distances between embeddings of different classes. Once the ID information is sufficiently learned, virtual outliers are generated in the feature space by calculating the unit vector from the mean of the i-th class to the nearest mean of a different class and multiplying it by the largest Mahalanobis distance among the features of the i-th class from their mean. The generated outliers are assigned high energy values, while ID samples are assigned low energy values using an uncertainty loss.

D. Experiments

To demonstrate the effectiveness of DiVO, we compare it with existing state-of-the-art OOD detection methods. We use the CIFAR-10 and CIFAR-100 datasets for training, employing the same ResNet18 backbone across all experiments. For evaluation, we use five commonly used OOD datasets: SVHN, LSUN, iSUN, Textures, and Places365. The detection performance on these five OOD datasets is evaluated using the AUROC and FPR95 metrics, showcasing DiVO’s robustness. We verify the synergy of applying supervised contrastive loss with ID-only data and subsequent data generation through experiments. Various ablation studies are conducted to adjust parameters and identify the optimal performance settings.