How to Exploit Hyperspherical Embeddings for Out-of-Distribution Detection?

Yifei Ming[1], **Yiyou Sun**[1], **Ousmane Dia**[2], **Yixuan Li**[1]
Department of Computer Sciences, University of Wisconsin-Madison[1]
Meta[2]
{alvinming,sunyiyou,sharonli}@cs.wisc.edu, ousamdia@meta.com

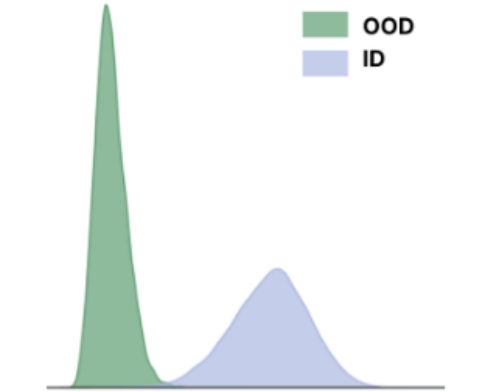
ICLR 2023

The task of OOD Detection

Based on the model f

Test input $x \longrightarrow G_\lambda(\mathbf{x}; f) = \begin{cases} \text{in} & S(\mathbf{x}; f) \geq \lambda \\ \text{out} & S(\mathbf{x}; f) < \lambda \end{cases}$

S: Scoring function



OOD score distribution

Mathematically, let $\mathcal{D}_{\text{test}}^{\text{ood}}$ denote an OOD test set where the label space $\mathcal{Y}^{\text{ood}} \cap \mathcal{Y}^{\text{in}} = \emptyset$. The decision can be made via a level set estimation:

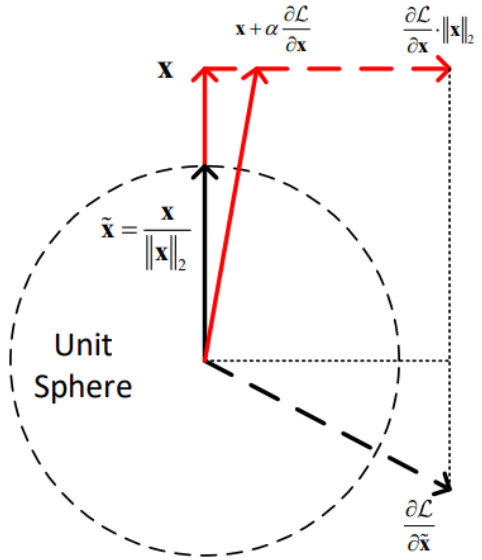
$$G_\lambda(x) = \mathbb{1}\{S(x) \geq \lambda\},$$

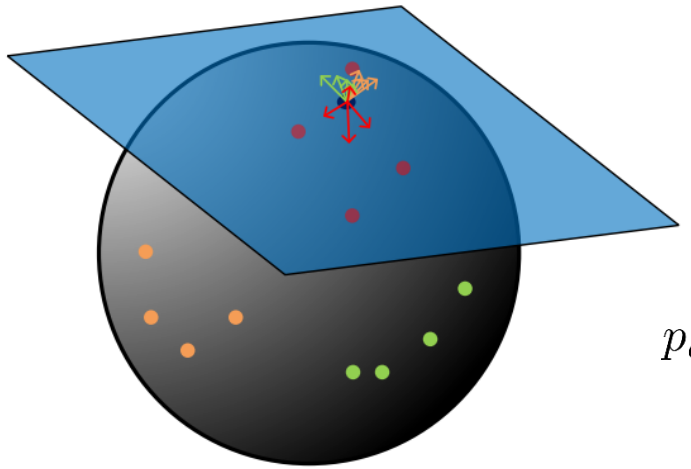
where samples with higher scores $S(x)$ are classified as ID and vice versa. The threshold λ is typically chosen so that a high fraction of ID data (*e.g.* 95%) is correctly classified.

- Using the model's confidence score for OOD detection, which can be abnormally high on OOD samples.

- Distance-based methods leverage feature embeddings extracted from a model, and operate under the assumption that the test OOD samples are relatively far away from the clusters of ID data.

Hyperspherical embeddings





Naturally modeled by the von Mises-Fisher (vMF) distribution:

$$p_d\left(\mathbf{z};\boldsymbol{\mu}_c,\kappa\right)=\frac{\kappa^{\frac{p}{2}-1}}{(2\pi)^{\frac{p}{2}}I_{\frac{p}{2}-1}(\kappa)}e^{\kappa\vec{\mu}^T\vec{z}}=Z_d(\kappa)\exp\left(\kappa\boldsymbol{\mu}_c^\top\mathbf{z}\right)$$

A hypersphere is a topological space that is homeomorphic to a standard n-sphere, which is the set of points in $(n+1)$-dimensional Euclidean space that are located at a constant distance from the center. When the sphere has a unit radius, it is called the unit hypersphere. Formally, an n-dimensional unit-hypersphere

$$S^n := \{z \in \mathbb{R}^{n+1} | \|z\|_2 = 1\}$$

Geometrically, hyperspherical embeddings lie on the surface of a hypersphere.

Previous work

- Arguably, the efficacy of distance-based approaches can depend largely on the quality of feature embeddings.
  - Recent works including SSD+ and KNN+ directly employ off-the-shelf contrastive losses for OOD detection.
  - They use the SupCon for learning the embeddings, which are then used for OOD detection with either parametric Mahalanobis distance or non-parametric KNN distance.

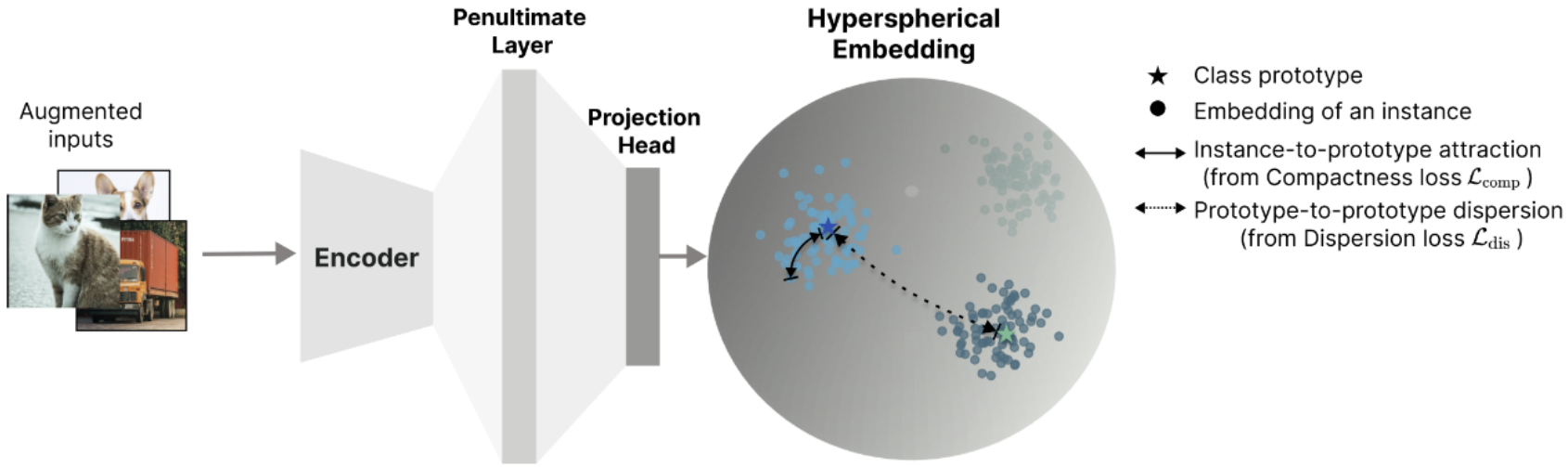
  (Example) When trained on CIFAR-10 using SupCon loss, the average angular distance between ID and OOD data is only 29.86 degrees in the embedding space, which is too small for effective ID-OOD separation.

- Two properties
  - Each sample has a higher probability assigned to the correct class in comparison to incorrect classes
  - Different classes are far apart from each other.

Introduces two loss ➔ *Dispersion Loss & Compactness Loss*

Framework Overview

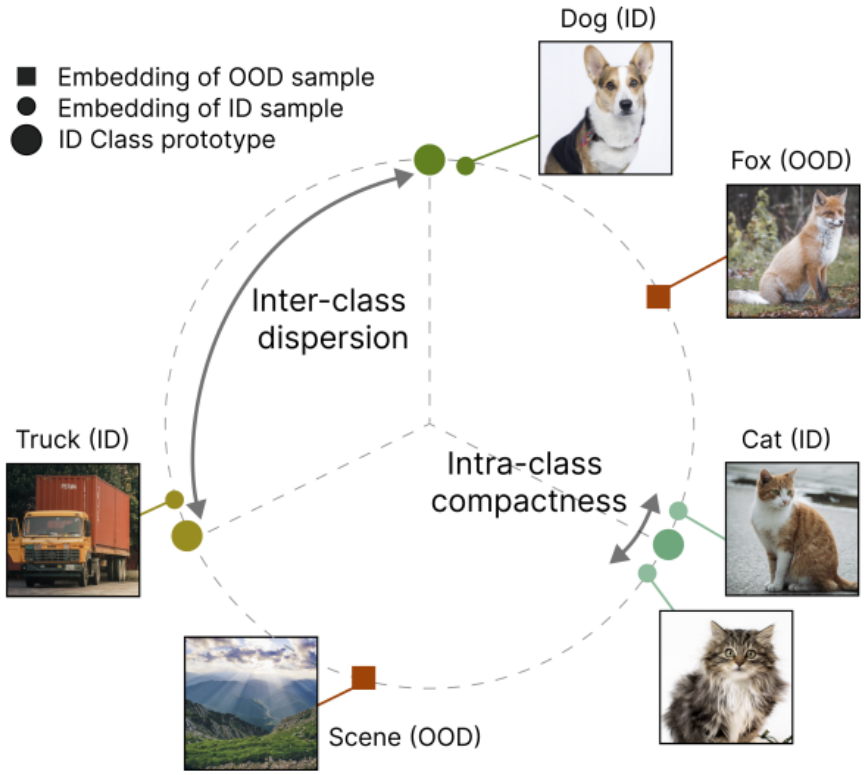
Compactness and Dispersion Regularized (CIDER)



➢A deep neural network encoder $f : \mathcal{X} \mapsto \mathbb{R}^e$ that maps the augmented input to a high dimensional feature

➢A projection head: $h : \mathbb{R}^e \mapsto \mathbb{R}^d$ maps high-dim feature to a lower-dim feature

Desirable Hyperspherical Embeddings in the Open World



ID and OOD Samples on the hypersphere

Preliminaries

➢Multi-class classification

$\mathcal{X}$ denotes the input space

$\mathcal{Y}^{\text{in}} = \{1, 2, ..., C\}$ denotes the ID labels.

The training set $\mathcal{D}_{\text{tr}}^{\text{in}} = \{(x_i, y_i)\}_{i=1}^{N}$ is drawn *i.i.d.* from $P_{\mathcal{X}\mathcal{Y}^{\text{in}}}$.

$P_{\mathcal{X}}$ denote the marginal distribution over $\mathcal{X}$, which is called the in-distribution (ID).

Modeling Hyperspherical Embeddings

➢Intra-class compactness

- Augmented input $\tilde{x} \rightarrow$ high-dim feature $f(\tilde{x}) \rightarrow$ low-dim feature $\tilde{z} = h(f(\tilde{x}))$
- Hyperspherical (l_2-normalized) feature $z = \tilde{z}/\|\tilde{z}\|_2$
- **von-Mises-Fisher (vMF) distribution**

$$p_d\left(\mathbf{z}; \boldsymbol{\mu}_c, \kappa\right) = \frac{\kappa^{\frac{p}{2}-1}}{(2\pi)^{\frac{p}{2}} I_{\frac{p}{2}-1}(\kappa)} e^{\kappa \vec{\mu}^T \vec{z}} = Z_d(\kappa) \exp\left(\kappa \boldsymbol{\mu}_c^\top \mathbf{z}\right) \quad \mathbb{P}\left(y = c \mid \mathbf{z}; \{\kappa, \boldsymbol{\mu}_j\}_{j=1}^C\right) = \frac{Z_d(\kappa) \exp\left(\kappa \boldsymbol{\mu}_c^\top \mathbf{z}\right)}{\sum_{j=1}^C Z_d(\kappa) \exp\left(\kappa \boldsymbol{\mu}_j^\top \mathbf{z}\right)}$$

$$= \frac{\exp\left(\boldsymbol{\mu}_c^\top \mathbf{z}/\tau\right)}{\sum_{j=1}^C \exp\left(\boldsymbol{\mu}_j^\top \mathbf{z}/\tau\right)},$$

- μ_c is the class prototype (mean) of class c
- κ indicates the spread of the distribution around mean direction

$$\text{argmax}_\theta \prod_{i=1}^N p\left(y_i \mid \mathbf{z}_i; \{\kappa_j, \boldsymbol{\mu}_j\}_{j=1}^C\right) \xrightarrow{\textbf{Maximum likehood}} \mathcal{L}_{\text{comp}} = -\frac{1}{N} \sum_{i=1}^N \log \frac{\exp\left(\mathbf{z}_i^\top \boldsymbol{\mu}_{c(i)}/\tau\right)}{\sum_{j=1}^C \exp\left(\mathbf{z}_i^\top \boldsymbol{\mu}_j/\tau\right)}$$

Modeling Hyperspherical Embeddings

➢Inter-class dispersion

- Optimize large angular distances among different class prototypes

$$\mathcal{L}_{\text{dis}} = \frac{1}{C}\sum_{i=1}^{C}\log\frac{1}{C-1}\sum_{j=1}^{C}\mathbb{1}\{j\neq i\}e^{\boldsymbol{\mu}_i^\top\boldsymbol{\mu}_j/\tau}.$$

- **C**ompactness and **D**ispersion **R**egularized Learning (CIDER)

$$\mathcal{L}_{\text{CIDER}} = \mathcal{L}_{\text{dis}} + \lambda_c\mathcal{L}_{\text{comp}}$$

➢Update of class prototypes

- Class prototypes are initialized as the classwise mean μ_c using the training set

- Updated via exponential-moving-average (EMA):

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

Pseudo-code for CIDER

Algorithm 1: Pseudo-code of CIDER.

Input: Training dataset $\mathcal{D}$, neural network encoder f, projection head h, classifier g, class prototypes $\boldsymbol{\mu}_j$ $(1 \leq j \leq C)$, weights of loss terms λ_d, and λ_c, temperature τ

for $epoch = 1, 2, \ldots,$ **do**

  for $iter = 1, 2, \ldots,$ **do**

    sample a mini-batch $B = \{\mathbf{x}_i, y_i\}_{i=1}^{b}$

    obtain augmented batch $\tilde{B} = \{\tilde{\mathbf{x}}_i, \tilde{y}_i\}_{i=1}^{2b}$ by applying two random augmentations to $\mathbf{x}_i \in B$ $\forall i \in \{1, 2, \ldots, b\}$

    for $\tilde{\mathbf{x}}_i \in \tilde{B}$ **do**

      // obtain normalized embedding

      $\tilde{\mathbf{z}}_i = h(f(\tilde{\mathbf{x}}_i)), \mathbf{z}_i = \tilde{\mathbf{z}}_i / \|\tilde{\mathbf{z}}_i\|_2$

      // update class-prototypes

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

    // calculate compactness loss

    $\mathcal{L}_{\text{comp}} = -\sum_{i=1}^{b} \log \frac{\exp\left(\mathbf{z}_i^\top \boldsymbol{\mu}_{c(i)}/\tau\right)}{\sum_{j=1}^{C} \exp\left(\mathbf{z}_i^\top \boldsymbol{\mu}_j/\tau\right)}$

    // calculate dispersion loss

    $\mathcal{L}_{\text{dis}} = \frac{1}{C} \sum_{i=1}^{C} \log \frac{1}{C-1} \sum_{j=1}^{C} \mathbb{1}\{j \neq i\} e^{\boldsymbol{\mu}_i^\top \boldsymbol{\mu}_j/\tau}$

    // calculate overall loss

    $\mathcal{L} = \mathcal{L}_{\text{dis}} + \lambda_c \mathcal{L}_{\text{comp}}$

    // update the network weights

    update the weights in the encoder and the projection head

Experiment Setup

In-distribution datasets

1) **CIFAR-10 [Krizhevsky 2012]**
A dataset with 60,000 images composed of ***five tasks*** from ***ten animal and vehicle classes***.

2) **CIFAR-100 [Krizhevsky 2012]**
A dataset with 60,000 images composed of ***20 tasks*** from ***100 generic object classes***.

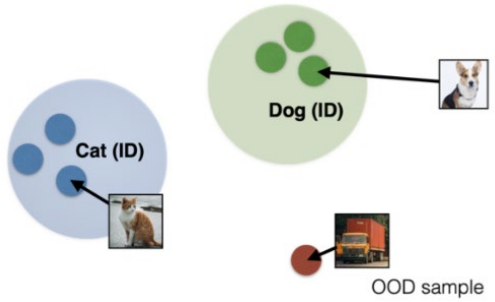
OOD test datasets

3) **SVHN[Netzer 2011]、Places365[Zhou 2017)]、Textures[Cimpoi 2014], LSUN [Yu 2015], and iSUN [Xu 2015].**

Test Time OOD Score

1) **Maha score [1] (based on the Mahalanobis distance)**

2) **KNN score [2] (based on the Euclidean distance to the K-th nearest neighbor)**



Evaluation metrics.

1) **FPR95 (the false positive rate of OOD samples when the true positive rate of ID samples is at 95%)**

2) **AUROC (the area under the receiver operating characteristic curve)**

3) **ID ACC (ID classification accuracy)**

[1] Lee et al., A Simple Unified Framework for Detecting OOD Saamples and Adversarial Attacks, NeurIPS 2018
[2] Sun et al., Out-of-Distribution Detection with Deep Nearest Neighbors, ICML 2022

Experiment

CIDER on Small-scale Datasets

- CIFAR-100 (ID): methods with contrastive losses are trained for 500 epoch
- Backbone: ResNet-34

Method	OOD Dataset SVHN FPR↓	SVHN AUROC↑	Places365 FPR↓	Places365 AUROC↑	LSUN FPR↓	LSUN AUROC↑	iSUN FPR↓	iSUN AUROC↑	Texture FPR↓	Texture AUROC↑	Average FPR↓	Average AUROC↑
Without Contrastive Learning												
MSP	78.89	79.80	84.38	74.21	83.47	75.28	84.61	74.51	86.51	72.53	83.12	75.27
ODIN	70.16	84.88	82.16	75.19	76.36	80.10	79.54	79.16	85.28	75.23	78.70	79.11
Mahalanobis	87.09	80.62	84.63	73.89	84.15	79.43	83.18	78.83	61.72	84.87	80.15	79.53
Energy	66.91	85.25	81.41	76.37	59.77	86.69	66.52	84.49	79.01	79.96	70.72	82.55
GODIN	74.64	84.03	89.13	68.96	93.33	67.22	94.25	65.26	86.52	69.39	87.57	70.97
LogitNorm	59.60	90.74	80.25	78.58	81.07	82.99	84.19	80.77	86.64	75.60	78.35	81.74
With Contrastive Learning												
ProxyAnchor	87.21	82.43	70.10	79.84	37.19	91.68	70.01	84.96	65.64	84.99	66.03	84.78
CE + SimCLR	24.82	94.45	86.63	71.48	56.40	89.00	66.52	83.82	63.74	82.01	59.62	84.15
CSI	44.53	92.65	79.08	76.27	75.58	83.78	76.62	84.98	61.61	86.47	67.48	84.83
SSD+	31.19	94.19	77.74	79.90	79.39	85.18	80.85	84.08	66.63	86.18	67.16	85.90
KNN+	39.23	92.78	80.74	77.58	48.99	89.30	74.99	82.69	57.15	88.35	60.22	86.14
CIDER	23.09	95.16	79.63	73.43	16.16	96.33	71.68	82.98	43.87	90.42	**46.89**	**87.67**

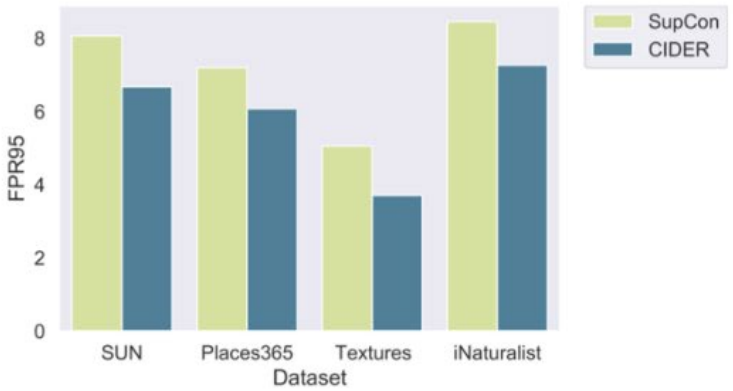
ID classification accuracy on CIFAR-100 (%)

Method	ID ACC
w.o. contrastive loss	
MSP	74.59
ODIN	74.59
GODIN	74.92
Energy	74.59
Mahalanobis	74.59
w. contrastive loss	
CE + SimCLR	73.54
SSD+	75.11
ProxyAnchor	74.21
KNN+	75.11
CIDER	75.35

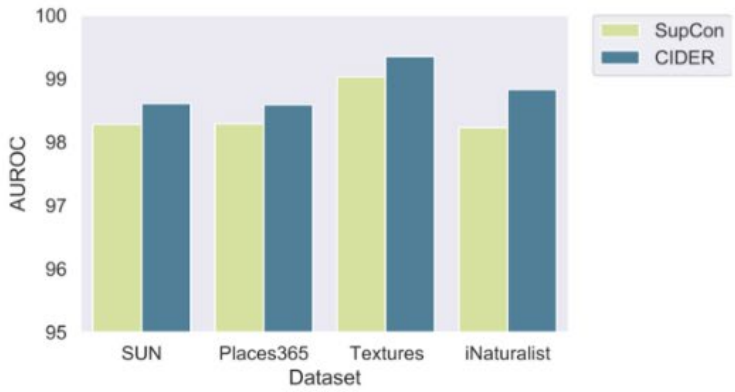
Experiment

CIDER on Large-scale Datasets

- ImageNet-100 (ID): finetune for 10 epoch



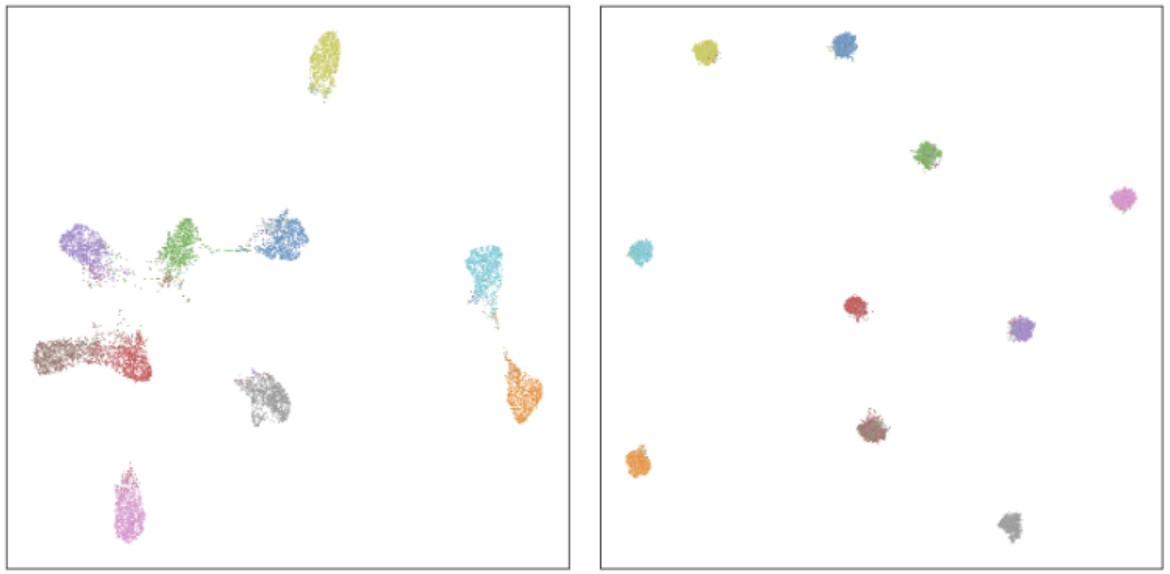
(a) FPR95 for different OOD test sets



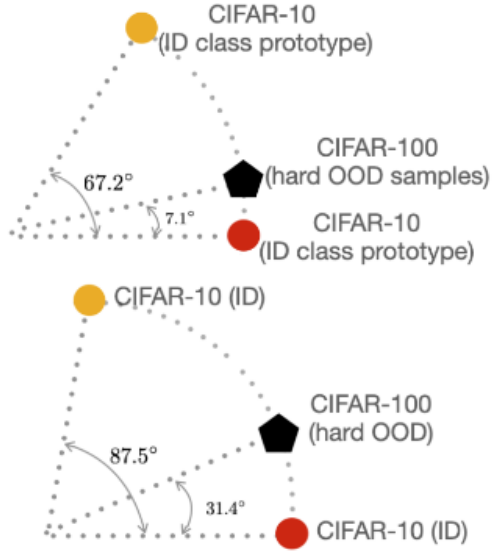
(b) AUROC for different OOD test sets

CIDER learns distinguishable representations.

Visualization of learned features by UMAP On CIFAR-10 (ID)



(a) ID embeddings of CE (left) and CIDER (right)



(b) ID & OOD of CE (top) and CIDER (down)

Figure 3: (a): UMAP (McInnes et al., 2018) visualization of the features when the model is trained with CE vs. CIDER for CIFAR-10 (ID). (b): CIDER makes OOD samples more separable from ID compared to CE (*c.f.* Table 4).

CIDER improves inter-class dispersion and intra-class compactness.

$$\text{Compactness}(\mathcal{D}_{\text{tr}}^{\text{in}}, \boldsymbol{\mu}) = \frac{1}{C}\sum_{j=1}^{C}\frac{1}{n}\sum_{i=1}^{n} z_i^\top \boldsymbol{\mu}_j \mathbb{1}\{y_i = j\},$$

$$\text{Dispersion}(\boldsymbol{\mu}) = \frac{1}{C}\sum_{i=1}^{C}\frac{1}{C-1}\sum_{j=1}^{C} \boldsymbol{\mu}_i^\top \boldsymbol{\mu}_j \mathbb{1}\{j \neq i\}.$$

$$\uparrow \text{Separability} = \frac{1}{|\mathcal{D}_{\text{test}}^{\text{ood}}|}\sum_{x\in\mathcal{D}_{\text{test}}^{\text{ood}}} \max_{j\in[C]} z_x^\top \boldsymbol{\mu}_j - \frac{1}{|\mathcal{D}_{\text{test}}^{\text{in}}|}\sum_{x'\in\mathcal{D}_{\text{test}}^{\text{in}}} \max_{j\in[C]} z_{x'}^\top \boldsymbol{\mu}_j, .$$

Table 4: Compactness and dispersion of CIFAR-10 feature embedding, along with the separability *w.r.t.* each OOD test set. We convert cosine similarity to angular degrees for better readability.

Training Loss	**Dispersion (ID)**↑	**Compactness (ID)**↓	**ID-OOD Separability**↑ (in degree)					
	(in degree)	(in degree)	CIFAR-100	LSUN	iSUN	Texture	SVHN	**AVG**
Cross-Entropy	67.17	24.53	7.11	14.57	13.70	13.76	11.08	12.04
SSD+ (SupCon loss)	75.50	22.08	23.90	28.55	25.70	33.45	37.70	29.86
CIDER (ours)	**87.53**	**21.35**	**31.41**	**48.37**	**41.54**	**39.60**	**51.65**	**42.51**

Experiment

Ablation study

- CIFAR-100 (ID): methods with contrastive losses are trained for 500 epoch
- Backbone: ResNet-34

Table 3: Ablation study on loss component. Results (in AUROC) are based on CIFAR-100 trained with ResNet-34. Training with only $\mathcal{L}_{\text{comp}}$ suffices for ID classification. Inter-class dispersion induced by $\mathcal{L}_{\text{dis}}$ is key to OOD detection.

Loss Components		AUROC↑						ID ACC↑
$\mathcal{L}_{\text{comp}}$	$\mathcal{L}_{\text{dis}}$	Places365	LSUN	iSUN	Texture	SVHN	**AVG**	
✓		79.63	85.75	84.45	87.21	91.33	85.67	75.19
	✓	54.76	69.81	54.99	44.26	46.48	54.06	2.03
✓	✓	**73.43**	**96.33**	**82.98**	**90.42**	**95.16**	**87.67**	**75.35**

Thanks