Out-of-distribution Detection with Deep Nearest Neighbors

Yiyou Sun
UW-Madison

Yifei Ming
UW-Madison

Xiaojin (Jerry) Zhu
UW-Madison

Yixuan Li
UW-Madison

Closed-world:
Training and testing distributions **match**

Open-world:
Training and testing distributions **differ**

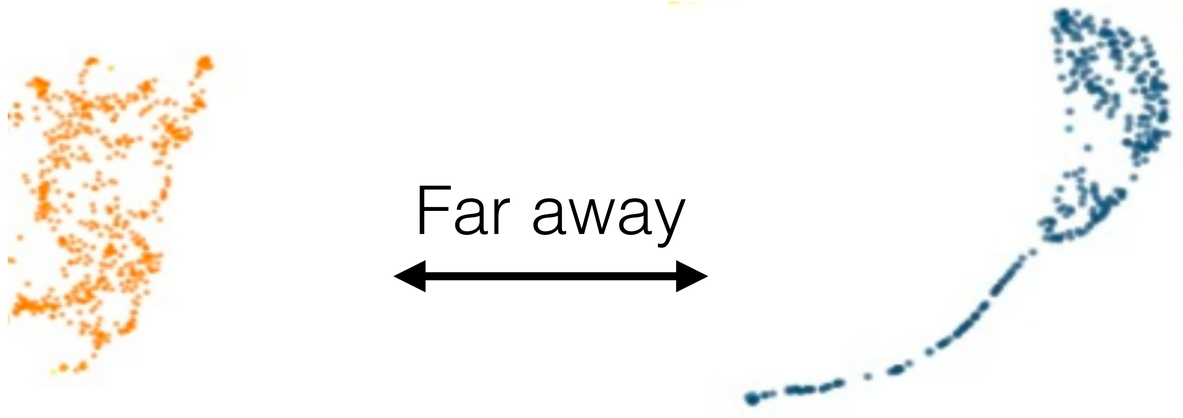
CNN

Prior literature on distance-based OOD detection

Mahalanobis
[Lee et al. NeurIPS 2018]

SSD
[Sehwag et al. ICLR 2021]

These methods assume samples in each class is **Gaussian** distributed



NOT
Gaussian Distributed

Out-of-distribution (OOD)
Embeddings

In-Distribution (ID)
Embeddings

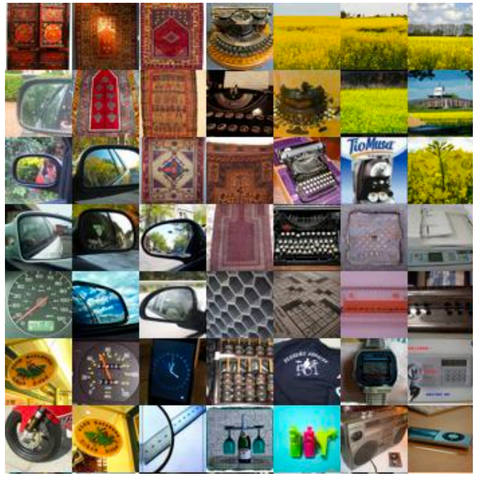
Our proposal:
K-nearest neighbor distance for OOD Detection

Leverage the **non-parametric** approach brings the following advantage:

1. Distributional assumption free
2. OOD-agnostic
3. Easy-to-use
4. Model-agnostic

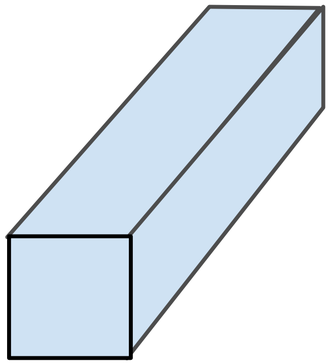
Methodology

Set up



ID: ImageNet

x



Convolutional Layers

Feature Encoder

$\phi(\mathbf{x})$

Normalization

Embedding

$\mathbf{z} = \phi(\mathbf{x})/\|\phi(\mathbf{x})\|_2$

OOD Detection with k-NN Distance

- 1. Given a test sample $\mathbf{x}^*$, we calculate feature vector $\mathbf{z}^* = \phi(\mathbf{x}^*) / \|\phi(\mathbf{x}^*)\|_2$
- 2. Calculate distance to the k-th nearest neighbor (k-NN): $\boldsymbol{r}_k(\mathbf{z}^*)$
- 3. The decision function is given by $G_\lambda(\mathbf{x}^*; \lambda) = \begin{cases} \text{ID} & -r_k(\mathbf{z}^*) \geq \lambda \\ \text{OOD} & -r_k(\mathbf{z}^*) < \lambda \end{cases}$

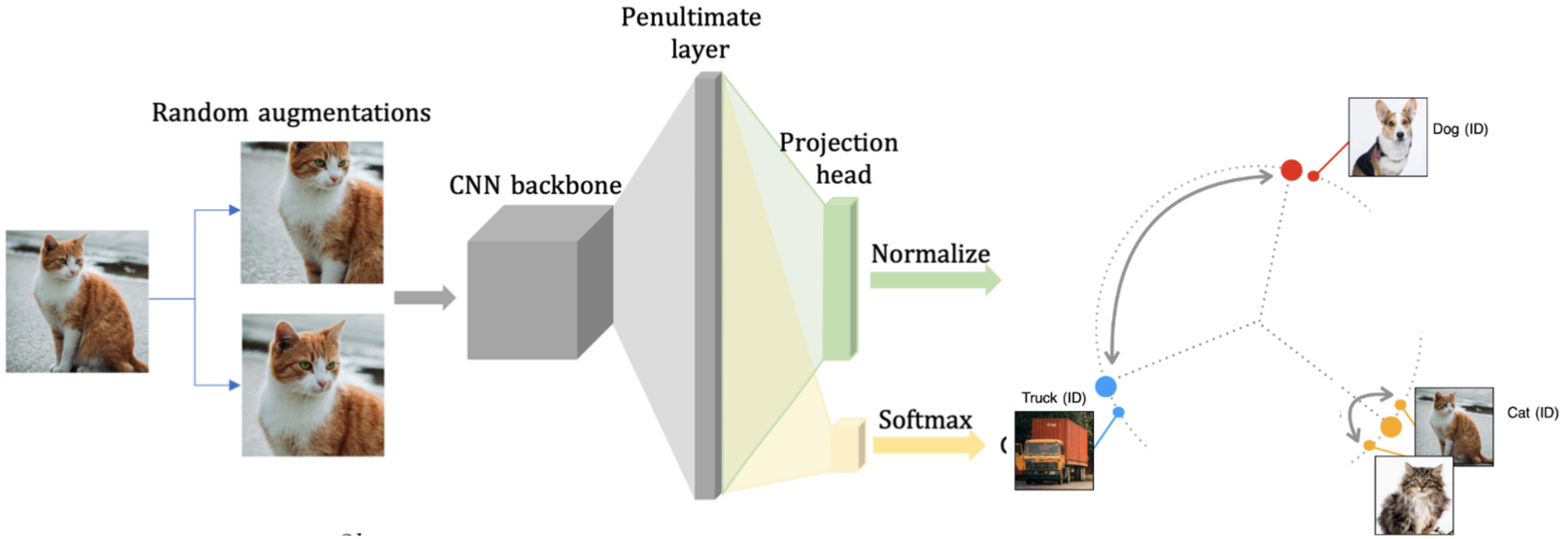
Experiments

Experiments Setting: Training Loss

(1) KNN:
Cross-entropy Loss

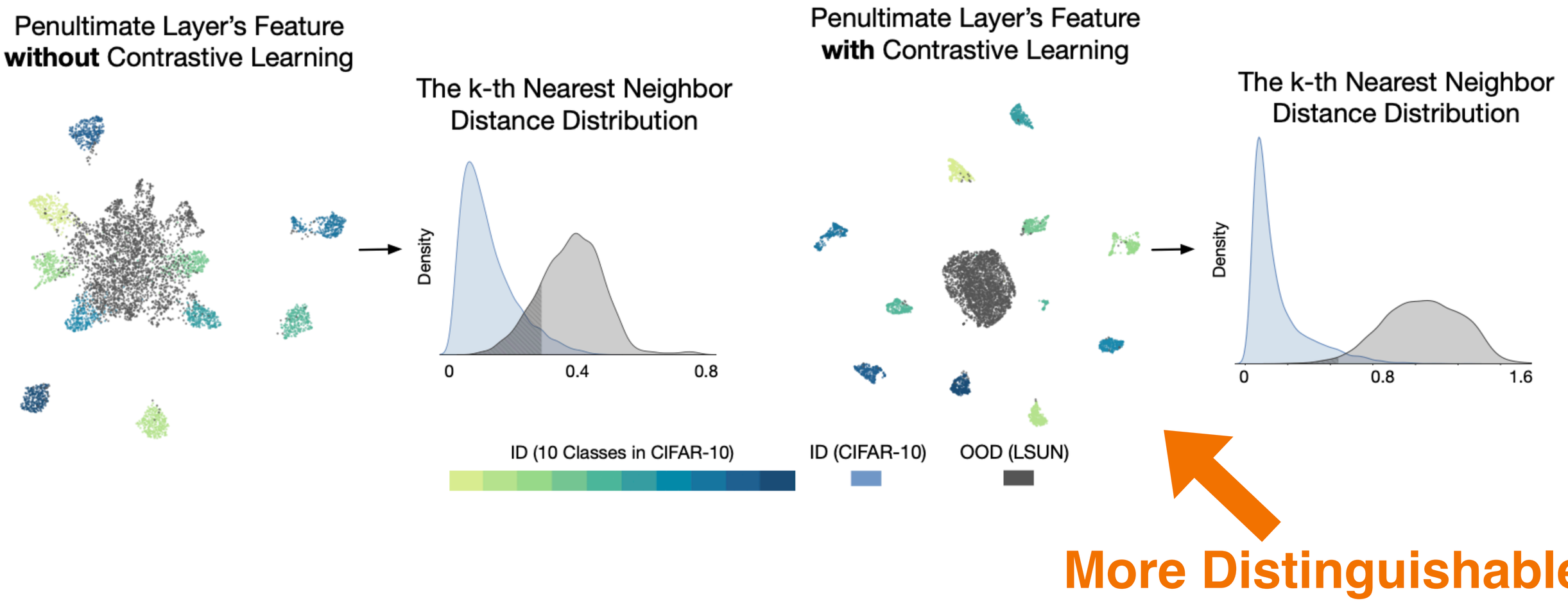
$$\mathcal{L}_{\text{ce}} = -\sum_{i=1}^{C} y_i \log(p_i), \text{for } C \text{ classes}$$

(2) KNN+:
Supervised Contrastive Loss
[Khosla, et al. 2020]



$$\mathcal{L}_{\text{sup}} = \sum_{i=1}^{2b} \frac{-1}{|P(i)|} \sum_{j \in P(i)} \log \frac{\exp(\mathbf{z}_i \cdot \mathbf{z}_j / \tau)}{\sum_{t=1, t \neq i}^{2b} \exp(\mathbf{z}_i \cdot \mathbf{z}_t / \tau)}$$

Contrastively Learned Representation Helps



More Distinguishable

K-NN Distance Achieves Superior Performance

Table 1. **Results on CIFAR-10.** Comparison with competitive OOD detection methods. All methods are based on a discriminative model trained on ID data only, without using outlier data. ↑ indicates larger values are better and vice versa.

Method	OOD Dataset										Average		ID ACC
	SVHN		LSUN		iSUN		Texture		Places365				
	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	FPR↓	AUROC↑	
Without Contrastive Learning													
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	94.21
ODIN	20.93	95.55	7.26	98.53	33.17	94.65	56.40	86.21	63.04	86.57	36.16	92.30	94.21
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	94.21
GODIN	15.51	96.60	4.90	99.07	34.03	94.94	46.91	89.69	62.63	87.31	32.80	93.52	93.96
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	94.21
KNN (ours)	24.53	95.96	25.29	95.69	25.55	95.26	27.57	94.71	50.90	89.14	30.77	94.15	94.21
With Contrastive Learning													
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	94.38
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	**95.07**
KNN+ (ours)	2.42	99.52	1.78	99.48	20.06	96.74	8.09	98.56	23.02	95.36	**11.07**	**97.93**	**95.07**

Additional Analysis

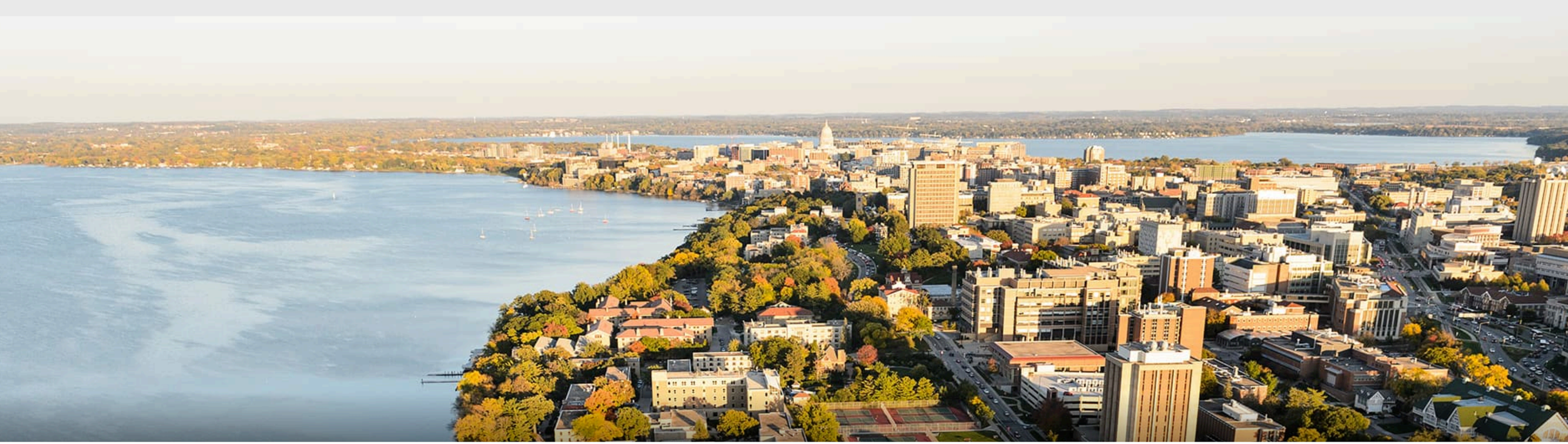
1. Ablation on k and random sampling ratio

2. Nearest Neighbor approach is competitive on hard OOD datasets

3. Comparison with other non-parametric methods

4. Theoretical justification of using the Nearest Neighbor approach for OOD detection

(See more details in the paper)



Thank you!

https://github.com/deeplearning-wisc/knn-ood