

A EXPERIMENTAL DETAILS

Software and Hardware. We conduct our experiments NVIDIA RTX A6000 GPUs (48GB VRAM). We use Ubuntu 22.04.2 LTS as the operating system and install the NVIDIA CUDA Toolkit version 11.6 and cuDNN 8.9. All experiments are implemented in Python 3.8.1 using the PyTorch 1.12.1 framework.

Training cross-entropy models. For methods using cross-entropy loss, such as MSP (Hendrycks & Gimpel, 2017), ODIN (Liang et al., 2018), Mahalanobis (Lee et al., 2018b), Energy (Liu et al., 2020), ViM (Wang et al., 2022), ReAct (Sun et al., 2021), DICE (Sun & Li, 2022), SHE (Zhang et al., 2023), and FeatureNorm (Yu et al., 2023b), we adopt the same training scheme as in (Ming et al., 2023) for CIFAR-10 and CIFAR-100. Specifically, the models are trained using stochastic gradient descent with momentum 0.9 and weight decay 10^{-4} . The initial learning rate is 0.1 and decays by a factor of 10 at epochs 100, 150, and 180. We train the models for 200 epochs on both CIFAR-10 and CIFAR-100. For the ImageNet benchmark, we adopt PyTorch’s pre-trained ResNet-50 model with ImageNet-1k weights.

Training contrastive models. For methods using SupCon loss (Khosla et al., 2020), such as KNN+ (Khosla et al., 2020) and SSD+ (Schwag et al., 2021), we adopt the same training scheme as in (Ming et al., 2023). For CIFAR-10 and CIFAR-100 models, we use stochastic gradient descent with a momentum of 0.9 and a weight decay of 10^{-4} . The initial learning rate is 0.5 and follows a cosine annealing schedule. We train the models for 500 epochs. The training-time temperature τ is set to be 0.1. For the ImageNet model, we use the checkpoint provided in (Sun et al., 2022b). For methods using CIDER, the models are trained using stochastic gradient descent with momentum 0.9 and weight decay 10^{-4} . The initial learning rate is 0.5 and follows a cosine annealing schedule. We use a batch size of 512 and train the model for 500 epochs. The training-time temperature τ is set to be 0.1. We adopt the exponential-moving-average (EMA) for the prototype update (Li et al., 2020), with a momentum of 0.99 for CIFAR-10 and 0.5 for CIFAR-100. Training details of ImageNet-1k are included in Section 4.3.

Evaluation configurations. We outline the configurations for methods that require appropriate hyperparameter selection, which has already been extensively studied in the literature.

- MSP uses the maximum softmax probability obtained from the logits. The method does not require any specific configuration.
- ODIN uses temperature scaling to calibrate the softmax score. We set the temperature T to 1,000.
- Mahalanobis utilizes class conditional Gaussian distributions based on the low- and upper-level features obtained from a model. These distributions are then used to calculate the Mahalanobis distance.
- Energy derives the energy score, which includes the temperature parameter. We set the temperature to the default value of $T = 1$.
- ViM generates an additional logit from the residual of the feature against the principal space. We set the dimension of principal space to $D = 256$ for ResNet-18 and ResNet-34 and $D = 1024$ for ResNet-50.
- ReAct improves the energy score by rectifying activations at an upper limit, which is set based on the p -th percentile of the activations estimated on the in-distribution (ID) data. We set p to the default value of $p = 90$.
- DICE utilizes logit sparsification to enhance the vanilla energy score, which is set based on the p -th percentile of the unit contributions estimated on the ID data. We set the sparsity parameter $p = 0.7$.
- SHE stores the mean direction of the penultimate layer features from correctly classified training samples. During inference, the Hopfield energy score is calculated as the dot product between the sample embedding and the class mean of the predicted class.
- FeatureNorm utilizes the norm of the feature map obtained from the chosen block in the neural network. We utilize the ImageNet-1k results as represented in Yu et al. (2023b),

and for CIFAR-10 and CIFAR-100, we run the experiments using block 4.1 as the selected block.

- `SSD+` applies the Mahalanobis score on the penultimate layer features obtained from a model pre-trained with the `SupCon` loss.
- `KNN+` and `CIDER` utilize the KNN score, which requires selecting a number of nearest neighbors k . Following the settings in [Ming et al. \(2023\)](#), we set k to 100, 300, and 1,000 for CIFAR-10, CIFAR-100, and the ImageNet benchmark, respectively.

B VALIDATION METHOD FOR SELECTING TEST-TIME TEMPERATURE

To select the test-time temperature for our proposed hyperspherical energy score, we follow the validation method outlined in [Hendrycks et al. \(2019\)](#). We generate a validation distribution by corrupting in-distribution data with speckle noise, creating speckle-noised anomalies that simulate out-of-distribution data. After that, we compute the performance of hyperspherical energy at different test-time temperatures on the validation set and select the one that achieves the highest AUROC.

C ID CLASSIFICATION ACCURACY

[Table 5](#) presents the in-distribution classification accuracy for each training dataset. We evaluate the classification accuracy of methods that involve learning hyperspherical representations, such as `KNN+`, `SSD+`, `CIDER`, and `Hyperspherical Energy`, by performing linear probes on normalized features, following the approach in [Khosla et al. \(2020\)](#). Our method shows competitive ID classification accuracy compared to the other existing methods, indicating it does not compromise the model’s capability to distinguish samples between in-distribution classes.

Table 5: ID classification accuracy of each method on CIFAR-10, CIFAR-100, and ImageNet (%).

Method	ID classification accuracy \uparrow		
	CIFAR-10	CIFAR-100	ImageNet
Methods using cross-entropy loss			
MSP	94.21	75.03	76.15
ODIN	94.21	75.03	76.15
Mahalanobis	94.21	75.03	76.15
Energy	94.21	75.03	76.15
ViM	94.21	75.03	76.15
ReAct	93.95	74.43	74.89
DICE	93.92	74.38	73.72
SHE	94.21	75.03	76.15
Methods using hyperspherical representations			
SSD+	94.75	75.42	79.10
KNN+	94.75	75.42	79.10
CIDER	94.62	74.28	76.55
Hyperspherical energy	94.62	74.28	76.55

D RESULTS ON CIFAR-10 BENCHMARK

In this section, we present additional results and analysis on the CIFAR-10 benchmark, using the experimental settings described in [Section A](#). As shown in [Table 6](#), `Hyperspherical Energy` displays competitive performance compared to existing state-of-the-art methods. In particular, our method achieves FPR95 of 14.16%, which is similar to `CIDER`, which achieves 13.85%.

E VISUALIZATION ANALYSIS FOR LARGE-SCALE DATASET

[Figure 4](#) presents the UMAP visualization of the learned embeddings derived from a subset of ImageNet-1k classes and larger-scale out-of-distribution (OOD) datasets. The class prototypes are

Table 6: OOD detection performance for CIFAR-10 (ID) with ResNet-18. Hyperspherical energy achieves competitive performance with state-of-the-art methods.

Method	SVHN		Places365		OOD Dataset LSUN		iSUN		Texture		Average	
	FPR ↓	AUROC ↑	FPR ↓	AUROC ↑	FPR ↓	AUROC ↑	FPR ↓	AUROC ↑	FPR ↓	AUROC ↑	FPR ↓	AUROC ↑
Methods using cross-entropy loss												
MSP	59.81	91.25	62.57	88.69	45.43	93.80	55.20	92.03	66.61	88.50	57.92	90.85
ODIN	53.81	91.29	44.31	91.04	10.96	97.93	28.36	95.49	55.32	89.40	38.55	93.03
Mahalanobis	9.24	97.80	83.94	70.04	67.73	73.61	5.57	98.71	23.09	92.92	37.91	86.61
Energy	54.43	91.22	43.85	91.08	10.21	98.05	27.53	95.57	54.98	89.38	38.20	93.06
ViM	26.34	95.23	44.84	91.24	15.65	97.33	30.57	95.10	25.69	95.01	28.62	94.78
ReAct	48.21	92.20	48.11	90.97	23.03	95.96	22.02	96.38	48.90	91.19	38.05	93.34
DICE	65.34	89.66	50.44	89.81	3.95	99.21	34.98	94.87	59.22	88.50	42.79	92.41
SHE	64.29	88.31	70.13	80.70	8.00	98.56	55.27	90.78	58.10	87.99	51.16	89.27
FeatureNorm	8.79	98.27	76.75	79.84	0.16	99.92	37.67	94.17	29.96	94.08	30.67	93.26
Methods using hyperspherical representations												
SSD+	1.52	99.68	28.56	94.74	6.13	98.48	33.69	95.15	13.05	97.70	16.59	97.15
KNN+	2.52	99.51	22.96	95.40	1.72	99.52	19.96	96.73	8.05	98.57	11.04	97.95
CIDER	3.46	99.37	31.57	94.50	2.59	99.37	15.97	97.28	15.66	97.58	13.85	97.62
Hyperspherical energy	3.89	99.28	32.59	94.14	3.05	99.29	16.02	97.20	15.27	97.64	14.16	97.51

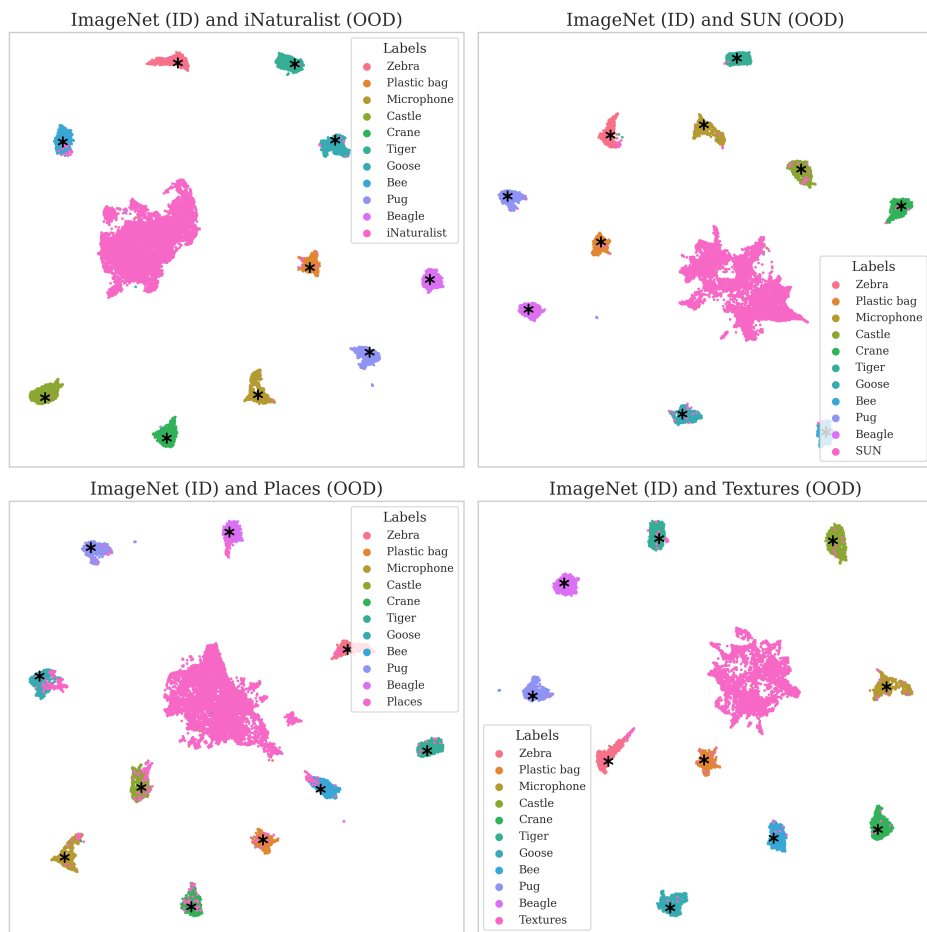


Figure 4: UMAP visualization of a subset of ImageNet classes and OOD datasets.

designated by a star symbol *, while the OOD embeddings are distinguished by pink color. These visualizations indeed reveal compact representations, where each sample appears to be effectively drawn in towards its corresponding class prototype. A notable separation between in-distribution (ID) and OOD classes is also evident, suggesting that the OOD samples exhibit a high hyperspherical energy score.

F ID AND OOD DENSITY PLOTS

Figure 5 showcases the density plots of the hyperspherical score for In-Distribution (ID) data, using CIFAR-100, and Out-Of-Distribution (OOD) data, comprising SVHN, Places365, LSUN, iSUN, and Textures.

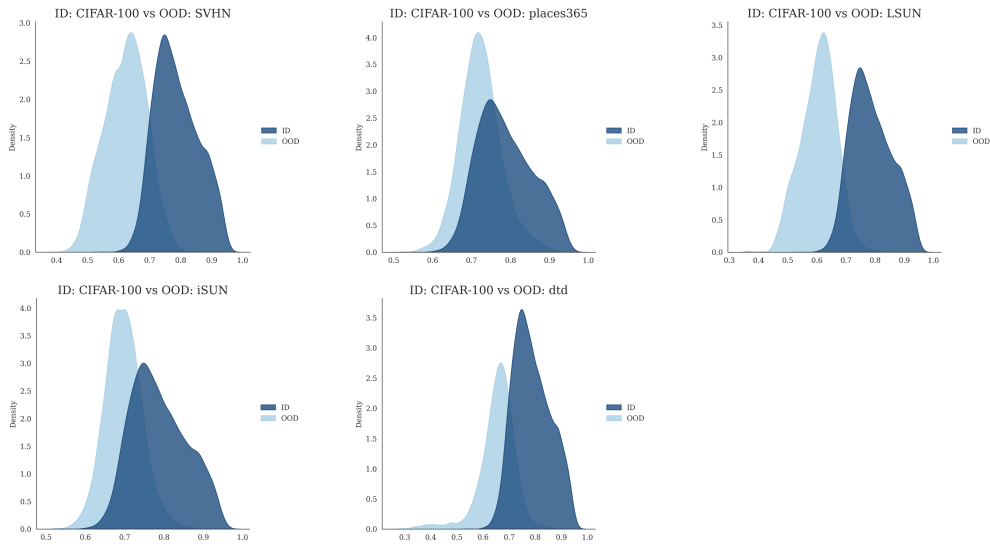


Figure 5: Density plots illustrating the distribution of hyperspherical scores.