Supplementary materials for ENCORE : A Neural Collapse Perspective on Out-of-Distribution Detection

1. Proof of Proposition 1

Proposition 1 *Let cosine logits be of the form $z_c = \kappa \cos(\theta_c)$, as defined in Eqn. 20. Assume: i) the weights $\{w_c\}_{c=1}^{C}$ and class means $\boldsymbol{\mu}_c$ form a unit-norm Equiangular Tight Frame (ETF), with pairwise inner product $\rho < 1$ ii) an In-Distribution (ID) input has angle $\theta \approx 0$ with its correct class weight w_y, so $z_y = \kappa \cos(\theta)$, and for all $c \neq y$, $z_c = \kappa\rho$ iii) an Out-of-Distribution (OOD) input has approximately equal angle $\Phi \approx 90°$ with all class weights, so $z_c = \kappa \cos(\Phi) \approx \kappa\epsilon$. Then the energy gap between ID and OOD satisfies:*

$$\begin{aligned}\Delta E &= E_{\text{ood}}(\epsilon) - E_{\text{id}}(\theta)\\ &\le -\log C + \kappa\left(1 - \frac{\theta^2}{2} - \epsilon\right)\end{aligned}$$

Proof We start by expanding the energy score for the ID input:

$$\begin{aligned}E_{\text{id}}(\theta) &= -\log\left(e^{\kappa\cos(\theta)} + (C-1)e^{\kappa\rho}\right)\\ &= -\log\left(e^{\kappa(1-\frac{\theta^2}{2}+o(\theta^2))} + (C-1)e^{\kappa\rho}\right)\\ &\ge -\log\left(e^{\kappa}e^{-\frac{\kappa\theta^2}{2}} + (C-1)e^{\kappa\rho}\right)\\ &\ge -\kappa\left(1 - \frac{\theta^2}{2}\right)\end{aligned}$$

Here, we take the Taylor series expansion of $cos(\theta)$ in the second line as $\theta \approx 0$. From Equation (13), we observe that $\rho < 0$. So, for large scaling factor κ, $e^{\kappa\rho}$ vanishes.
Now, the OOD input, having nearly equal small cosine similarity $\cos(\Phi) \approx \epsilon$ to all class weights, gives:

$$\begin{aligned}E_{\text{ood}}(\Phi) &= -\log\left(C \cdot e^{\kappa\cos(\Phi)}\right)\\ \implies E_{\text{ood}}(\Phi) &= -\kappa\cos(\Phi) - \log C\\ \implies E_{\text{ood}}(\epsilon) &= -\kappa\epsilon - \log C\end{aligned}$$

Now, subtract the ID energy from the OOD energy:

$$\begin{aligned}\Delta E &= E_{\text{ood}}(\Phi) - E_{\text{id}}(\theta)\\ &\le (-\kappa\epsilon - \log C) + \kappa\left(1 - \frac{\theta^2}{2}\right)\\ &\le -\log C + \kappa\left(1 - \frac{\theta^2}{2} - \epsilon\right)\end{aligned}$$

Here, the inequality flips as we are subtracting the $E_{id}(\theta)$. This completes the proof

2. Related Work

Early approaches to OOD detection focused on softmax-based confidence scores. Hendrycks and Gimpel [6] demonstrated that the maximum softmax probability could serve as a simple baseline for detecting OOD inputs. ODIN [10] improved this by applying temperature scaling and input perturbation to better separate in-distribution (ID) and out-of-distribution (OOD) samples. The Mahalanobis method [9] proposed modeling the ID feature distributions with class-conditional Gaussians and computing confidence based on Mahalanobis distance. However, these methods typically require access to OOD data for hyperparameter tuning, which is often unrealistic in practice.

More advanced approaches attempt to overcome this limitation. For example, GODIN [7] extends ODIN by modeling the joint distribution of labels and domains, effectively decoupling softmax outputs. Energy-based methods [12] draw inspiration from the Helmholtz free energy formulation, providing a theoretically grounded alternative to softmax scores. Wei et al. [20] demonstrated that normalizing logits and using a modified cross-entropy loss improves separation between ID and OOD samples. Decoupled MaxLogit (DML) [24] further refined this by decomposing the logit output into two orthogonal components, i.e., MaxCosine and MaxNorm. While effective, these methods often require changes to the training procedure, limiting their practical applicability. In contrast, our approach introduces no such modifications, making it compatible with a wide range of training pipelines. In addition to logit-based scores, recent work has explored the use of internal model dynamics for OOD detection. GradNorm [8] was the first to highlight that gradient-based signals can help distinguish OOD inputs. ReAct [16] observed that OOD samples tend to produce abnormally high activations in the penultimate layer and proposed clamping these activations to a fixed threshold. ASH [4] further investigated how simple activation shaping techniques - such as norm-based pruning, binarization, and rescaling - can disproportionately affect OOD versus ID data.

While these methods significantly advanced the state of the art, most approaches are empirical in nature ([4, 8, 16]), and such they fail to properly characterize the properties underpinning successful OOD detection. In contrast, ENCORE rethinks the widely used energy score through the lens of neural collapse and proposes a superior OOD detection method based on this intuition. By connecting our proposed modifications to the geometric and statistical regularities observed under neural collapse, ENCORE provides insights into *when* and *why* OOD detection is expected to succeed, thus offering a mechanism to assess its reliability based on measurable properties of the underlying Deep Neural Network (DNN).

3. Implementation Details

DNN Training For the Resnet 18 architecture, we use the pretrained weights provided by OpenOOD [22, 23]. For the vision transformer and ConvNext architectures, we use the pretrained weights from PyTorch [2]. For the alternate architectures, we use RepVGG a2 architecture for CIFAR benchmarks. It is trained for 200 epochs, with a batch size of 256 using SGD optimizer. The hyperparameters of the optimizer are learning rate of 0.1, momentum of 0.9, and weight decay of 0.0005. We also used Cosine annealing learning rate scheduler with minimum learning rate of 10^{-6}.

4. Details of OOD Baselines

We provide an overview of key OOD detection methods that serve as baselines for our work.

4.1. Softmax-Based Methods

Maximum Softmax Probability (MSP) [6] uses the maximum softmax output probability as a confidence score, assuming lower values indicate OOD samples. However, neural networks tend to be overconfident, leading to poor OOD detection.

OpenMax [3] extends softmax by incorporating statistical extreme value theory (EVT) to model the probability of a sample belonging to an unknown class, reducing overconfidence on OOD samples. Unlike traditional classification methods that assume all test samples belong to known classes, OpenMax can effectively recognize when an input does not belong to any of the predefined classes (i.e., out-of-distribution samples). It works by modifying the final softmax layer of a network to output probabilities for both known and unknown classes. This is achieved by fitting a Weibull distribution to the activations of known classes, allowing the network to estimate the likelihood of a sample belonging to an unknown class. The approach significantly improves the model's ability to handle open set scenarios, where the distribution of test data differs from the training distribution.

ODIN [10] adopts the baseline [6] and improves it by incorporating temperature scaling and input preprocessing steps. Specifically, they manipulate the temperature parameter $T \in \mathbb{R}_+$, to increase the separation between the ID and OOD examples. The resulting score function $S_{\hat{y}}(\mathbf{x};T)$ is given by Equation 1:

$$\begin{gathered} S_i(\mathbf{x};T) = \frac{exp(f_i(x)/T)}{\sum_{j=1}^{j=N} exp(f_j(x)/T)} \\ S_{\hat{y}}(\mathbf{x};T) = \max_i \ S_i(\mathbf{x};T) \end{gathered} \tag{1}$$

Here, $f_i(\mathbf{x})$ denote the logit value corresponding to i-th class for DNN $\mathbf{f}$. ODIN further complements the temperature scaling by perturbing the input image before feeding it into the DNN.

4.2. Distance-Based Methods

Mahalanobis Distance (MDS) [9] models the ID features using class-conditional Gaussian distributions and use the Mahalanobis distance as the score for OOD detection. The distance is calculated for each layer separately, while auxiliary OOD data is used to fit the distances collected from different layers into a logistic regression model which greatly diminishes its utility as it requires access to the OOD distribution to be detected and trains a separate logistic regression model for each OOD distribution.

Relative Mahalanobis Distance (RMDS) [14] refines MDS by normalizing distances relative to ID variations, improving robustness to dataset shifts.

KNN-Based OOD Detection [17] computes the distance of a test sample to its k-nearest neighbors in feature space, assuming OOD samples lie farther from ID clusters.

4.3. Feature-Based Methods

GRAM [15] computes Gram matrices from feature maps to analyze higher-order correlations, distinguish-

ing OOD samples based on their activation patterns.
ViM (Virtual Logit Matching) [19] decomposes feature representations into ID-aligned and OOD-related components, using the residual component for OOD detection. The core idea involves generating an additional logit, termed the "virtual logit," derived from the residual of the feature vector against its principal subspace. This virtual logit is then scaled and appended to the original logits. By applying the softmax function to this augmented logit set, the probability corresponding to the virtual logit serves as an indicator of OOD-ness. A higher probability suggests that the input is likely OOD.
DICE [18] is a method designed to enhance the detection of out-of-distribution (OOD) inputs in machine learning models by leveraging sparsification techniques. Traditional models often rely on the full set of parameters, which can include unimportant weights and units, leading to overconfident predictions on OOD data. DICE addresses this issue by ranking model weights based on their contribution and selectively utilizing the most salient ones to derive outputs for OOD detection. This targeted sparsification prunes away noisy signals, resulting in reduced output variance for OOD data and a sharper output distribution, thereby strengthening the separability between ID and OOD samples.
fDBD [11] proposes a closed-form estimation to quantify the distance of feature representations to decision boundaries within the feature space. The distance from a feature vector $\mathbf{z}$ to the decision boundary of class c is given by:

$$d_c(\mathbf{z}) = \frac{\left|\mathbf{w}_c^\top \mathbf{z} + b_c\right|}{\|\mathbf{w}_c\|} \tag{2}$$

where $\mathbf{w}_c$ and b_c are the weight vector and bias term associated with logit layer of the DNN for class c, and $\|\mathbf{w}_c\|$ is the Euclidean norm of $\mathbf{w}_c$.

4.4. Energy-Based and Gradient-Based Methods

Energy-Based OOD Detection [12] replaces softmax probability with an *energy function* derived from the log-sum-exp of logits - called the energy of the input, offering improved OOD separation. The energy function for a classification model with logits $f(\mathbf{x})$ is defined as:

$$E(\mathbf{x}) = -\log \sum_c e^{f_c(\mathbf{x})}$$

where:

- $f_c(\mathbf{x})$ is the logit (pre-softmax score) for class c,
- The summation runs over all classes c.

A sample is classified as out-of-distribution (OOD) if:

$$E(\mathbf{x}) > \tau$$

where τ is a predefined threshold.
GradNorm [8] is a method that leverages the gradients of the parameters of a deep neural network (DNN). It employs a label-agnostic score function to cast the out-of-distribution (OOD) detection problem as a binary classification task. For a DNN parameterized by $\mathbf{w}$, GradNorm defines the score function for an input $\mathbf{x}$ as $S(\mathbf{x}) = ||\frac{\partial D_{KL}(\mathbf{u}||softmax(f(\mathbf{x})))}{\partial \mathbf{w}}||_p$, where $\mathbf{u}$ represents the uniform distribution and D_{KL} denotes the Kullback-Leibler divergence. The key idea behind this formulation is that, for in-distribution (ID) inputs, the model's predictions typically concentrate around the target class, leading to a higher KL divergence and its gradient. The parameter vector $\mathbf{w}$ is a concatenation of the parameters from all layers of the network into a single vector, regardless of their original structure. The authors demonstrate that the gradients from the last layer's parameters are sufficiently informative to differentiate between ID and OOD inputs.
GEN [13] proposed a generalized entropy-based approach. Specifically, the authors used the generalized entropy family defined by $G(\mathbf{p}) = \sum_i p_i^\gamma (1 - p_i)^\gamma$, where $\mathbf{p}$ denotes a categorical probability distribution. This score function, coupled with truncation of very small probabilities, showed competitive result for state-of-the-art models like Swin Transformer and BiT-S R101×1 on ImageNet benchmark.

4.5. Activation Shaping Methods

ReAct [16] uses a rectification operation applied to activations in model space to enhance the separability between ID and OOD samples, thereby improving OOD detection. The activations from the penultimate layer, denoted as $h(\mathbf{x})$, are modified using the ReAct operation, which is defined as

$$\bar{h}(\mathbf{x}) = \mathrm{ReAct}(h(\mathbf{x}); c)),$$

where

$$\mathrm{ReAct}(x; c) = \min(x, c).$$

The rectified activations are then used to compute the model output as

$$f^{\mathrm{ReAct}}(\mathbf{x}; \theta) = \mathbf{W}^T \bar{h}(\mathbf{x}) + \mathbf{b}.$$

These modified outputs can be incorporated into any score function for OOD detection.

ASH [4] modifies the activations of a DNN in a post-hoc manner to facilitate the detection of OOD samples using existing scoring methods. The work assumes that modern overparameterized DNN produce redundant representations for the task it is trained for. Therefore, the representation can be greatly simplified while preserving performance and providing additional benefits in OOD detection. Based on this hypothesis, the authors propose three variants of the ASH algorithm where they set values smaller than p-th percentile of the representation to 0 as a form of activation pruning. The first variant, **ASH-P**, keeps the un-pruned activations unchanged. In the second variant, **ASH-B**, the un-pruned activations are assigned a value such that their total value equals that of the total value of the original activations. The third variant, **ASH-S**, calculates s_1 and s_2, the sums of the activations before and after pruning, and scales the unpruned values with $exp(s_1/s_2)$. They apply this activation shaping to the penultimate layer of the DNN. The effect of the choice of the value of p for setting the pruning threshold is dependent on the architecture and dataset and takes values in the range of [65, 95]. While this method is relatively simple, it integrates the detector into the architecture impacting the performance on ID data. The unique perspective of this method indicates that sparsity can be beneficial to OOD detection.

SCALE [21] shows that pruning or down-weighting the activations as done in ASH [4] can hurt both the accuracy and the OOD detection performance. They theoretically show that instead of pruning or down-weighting the activations, simply up-weighting a percentage of the activations can be beneficial. Based on this they propose to up-weight, for each sample, *top p percent* of the activations.

4.6. Subspace Projection Based

NECO [1] leverages the geometric properties of neural collapse (NC) in DNNs. NECO exploits the orthogonality between ID data, which collapses to a simplex structure, and OOD data, which centers around the null vector. The method calculates a NECO score based on the relative norm of a sample in the subspace occupied by the simplex structure, normalized by the full feature vector norm.

Kernel PCA Based [5] explores the use of Kernel Principal Component Analysis (KPCA) for OOD detection in DNNs. It addresses the limitations of traditional Principal Component Analysis (PCA) by leveraging non-linear kernel mappings, specifically a cosine kernel (method termed as CoP) and a cosine-Gaussian kernel (method termed as CoRP), to better separate ID and OOD features in a transformed feature space. By computing reconstruction errors in this non-linear space, the method improves OOD detection performance while maintaining computational efficiency.

5. Detailed Results

We provide the complete results on different benchmarks in this section.

6. Algorithm for ENCORE

Algorithm 1 ENCORE : Feature-Scaled Cosine Energy Score for OOD Detection

Input: Test input x; trained model F_θ with feature extractor $\phi_\theta(x)$ and classifier weights $\{w_c\}_{c=1}^{C}$; principal subspace $\mathbf{P}$; scaling constant λ
Output: OOD score $S(x)$

Setup (once after training):
Extract features $\{\phi_\theta(x_i)\}_{x_i \in \mathcal{D}_{\text{ID}}}$ from training data
Perform PCA on features to obtain principal subspace $\mathbf{P}$

Inference (for each test input x):
Extract feature $h \leftarrow \phi_\theta(x)$
Project feature: $h_p \leftarrow \text{Proj}_{\mathbf{P}}(h)$
Compute scaling factor: $\kappa \leftarrow \exp\left(\lambda \cdot \frac{\|h_p\|}{\|h\|}\right)$
for each class $c = 1$ to C **do**
    Compute cosine logit: $z_c \leftarrow \kappa \cdot \frac{w_c^\top h}{\|w_c\| \cdot \|h\|}$
end for
Compute energy score: $E(x) \leftarrow -\log \sum_{c=1}^{C} \exp(z_c)$
Return OOD score: $S(x) \leftarrow -E(x)$

Table 1. Comparison of ENCORE with the baseline approaches on CIFAR10 benchmark for Resnet-18.

	CIFAR100		TIN		Near OOD		MNIST		SVHN		Textures		Places365		Far OOD	
	FPR	AUROC	FPR	AUROC	FPR	AUROC	FPR	AUROC	FPR	AUROC	FPR	AUROC	FPR	AUROC	FPR	AUROC
MSP	59.90	86.73	47.32	88.64	53.61	87.69	19.23	93.95	23.82	91.68	40.16	89.13	41.67	89.35	31.22	91.03
Openmax	52.66	86.48	41.73	87.92	47.19	87.20	19.82	90.69	23.46	90.05	35.82	88.96	38.19	88.48	29.32	89.55
ODIN	84.82	79.55	84.60	81.18	84.71	80.37	15.39	96.60	69.04	84.61	83.00	83.86	76.66	83.87	61.02	87.24
MDS	49.66	85.95	42.42	87.63	46.04	86.79	28.38	88.61	24.01	91.83	23.21	93.78	45.42	86.63	30.26	90.21
RMDS	48.77	88.38	35.74	90.76	42.26	89.57	18.88	94.20	20.90	92.24	25.88	91.93	31.59	91.48	24.31	92.46
GRAM	94.77	52.53	92.99	52.34	93.88	52.43	60.00	71.32	31.68	90.09	97.32	52.32	88.34	65.07	69.34	69.70
Energy	72.71	85.55	63.64	88.35	68.18	86.95	15.49	96.32	29.31	92.60	60.46	88.63	56.39	89.63	40.41	91.80
Gradnorm	95.66	52.74	94.87	55.11	95.26	53.93	78.93	70.06	91.00	50.29	98.00	46.29	89.48	67.95	89.35	58.65
React	75.50	85.24	66.63	87.79	71.07	86.51	18.42	95.38	42.28	90.30	67.63	87.27	40.00	91.41	42.08	91.09
ViM	53.60	87.44	41.79	89.67	47.69	88.55	18.06	94.25	18.29	94.50	21.86	94.76	44.51	89.15	25.68	93.17
KNN	47.90	85.75	30.93	91.71	39.42	88.73	20.61	94.41	20.43	93.01	24.43	93.02	29.46	92.10	23.73	93.13
DICE	84.32	76.05	76.71	79.31	80.52	77.68	24.50	95.72	35.77	91.86	62.50	85.01	92.37	69.19	53.78	85.44
ASH	89.97	72.63	88.00	75.77	88.98	74.20	67.90	84.07	83.33	70.61	86.95	74.05	67.25	85.09	76.36	78.46
GEN	63.68	86.71	52.10	88.89	57.89	87.80	18.02	95.00	24.07	92.18	46.13	89.36	46.12	89.74	33.59	91.57
FDBD	**41.59**	89.34	31.66	91.55	36.62	90.45	19.28	94.95	22.83	92.34	24.82	92.86	**27.02**	92.67	23.49	93.20
CoRP	43.46	89.32	33.74	91.60	38.60	90.46	17.24	95.55	18.98	94.18	22.44	94.35	31.26	92.30	22.48	94.12
ENCORE (Ours)	41.84	**89.89**	**30.68**	**92.58**	**36.27**	**91.23**	**13.28**	**97.00**	**17.77**	**94.59**	23.02	**93.78**	30.39	**92.85**	**21.11**	**94.56**

Table 2. Comparison of ENCORE with the baseline approaches on CIFAR10 benchmark for Repvgg-a2.

	CIFAR100		TIN		Near OOD		MNIST		SVHN		Textures		Places365		Far OOD	
	FPR	AUROC	FPR	AUROC	FPR	AUROC	FPR	AUROC	FPR	AUROC	FPR	AUROC	FPR	AUROC	FPR	AUROC
Energy	74.42	86.06	69.72	87.81	72.07	86.94	19.12	95.45	19.70	94.61	83.86	86.21	74.06	87.28	49.18	90.89
Gradnorm	98.38	39.82	98.28	40.66	98.33	40.24	76.96	77.43	83.31	69.16	98.58	49.50	98.59	37.81	89.36	58.47
React	62.39	78.58	53.56	81.73	57.97	80.16	67.13	72.53	44.82	82.52	63.09	76.85	58.32	78.39	58.34	77.57
ViM	40.71	88.25	36.23	90.88	38.47	89.57	20.01	91.87	17.46	93.57	38.09	90.45	40.47	90.34	29.01	91.56
KNN	**34.84**	**90.00**	37.55	91.40	36.20	90.70	25.53	92.09	20.86	93.44	**32.31**	**91.15**	**35.12**	**91.10**	28.46	91.95
DICE	81.06	81.61	78.63	83.65	79.84	82.63	**10.23**	**98.04**	50.32	91.61	90.01	80.87	68.24	86.31	54.70	89.21
ASH	96.91	59.27	97.16	60.85	97.03	60.06	80.99	77.93	79.71	77.55	96.51	65.85	97.73	55.04	88.74	69.09
GEN	72.88	86.22	69.03	87.87	70.96	87.05	19.80	95.05	19.47	94.45	82.50	86.30	73.08	87.29	48.71	90.77
FDBD	36.27	89.76	32.21	91.28	34.24	90.52	25.43	92.66	20.74	93.81	38.48	90.32	33.54	90.75	29.55	91.89
CoRP	38.37	88.55	34.32	90.45	36.35	89.5	24.43	92.05	22.44	92.45	39.32	90.03	38.35	88.95	31.14	90.87
ENCORE (Ours)	37.09	89.85	**32.34**	**91.69**	**34.72**	**90.77**	23.01	93.64	**15.04**	**96.25**	37.26	91.05	38.07	90.61	28.35	**92.89**

Table 3. Comparison of ENCORE with the baselines on Imagenet benchmark for ConvNext Small architecture.

	SSB Hard		NINCO		Near OOD		Inaturalist		Textures		Openimage-O		Far OOD	
	FPR	AUROC	FPR	AUROC	FPR	AUROC	FPR	AUROC	FPR	AUROC	FPR	AUROC	FPR	AUROC
MSP	80.75	73.95	70.73	81.20	75.74	77.58	40.49	90.63	93.81	75.76	75.64	82.96	69.98	83.12
React	82.16	68.61	69.41	76.60	75.79	72.61	56.68	78.08	86.52	76.71	70.14	77.86	71.11	77.55
ViM	81.67	**73.21**	58.59	84.11	70.13	78.66	**17.29**	93.95	38.45	89.42	**29.35**	**88.76**	28.36	90.71
KNN	80.42	71.85	54.78	83.96	67.60	77.90	21.04	94.17	37.32	89.60	26.44	**92.92**	**28.27**	**92.23**
DICE	93.14	59.52	95.69	60.40	94.41	59.96	87.38	65.43	79.29	78.21	85.46	72.09	84.04	71.91
ASH	98.96	40.40	98.43	44.19	98.70	42.29	98.16	36.62	94.56	65.61	99.00	46.06	97.24	49.43
GEN	79.81	72.00	54.36	82.65	67.08	77.33	17.78	**95.50**	81.16	84.06	38.90	90.67	45.95	90.08
CoRP	80.25	71.65	54.76	72.60	67.51	72.13	25.32	88.65	35.62	90.12	25.52	90.91	29.82	89.89
ENCORE (Ours)	**79.07**	72.28	**52.95**	**83.12**	**66.01**	**77.70**	22.54	90.64	**36.70**	**90.06**	**26.36**	91.45	28.53	90.72

Table 4. Comparison of ENCORE with the baselines on Imagenet benchmark for ViT-Huge architecture.

	SSB Hard		NINCO		Near OOD		Inaturalist		Textures		Openimage-O		Far OOD	
	FPR	AUROC	FPR	AUROC	FPR	AUROC	FPR	AUROC	FPR	AUROC	FPR	AUROC	FPR	AUROC
MSP	67.22	79.64	45.73	88.67	56.47	84.15	17.28	97.00	46.82	89.26	30.43	93.56	31.51	93.27
React	55.49	85.76	34.23	92.21	44.86	88.99	3.98	98.94	30.42	92.39	10.92	97.67	15.11	96.33
ViM	58.54	84.63	32.28	92.91	45.41	88.77	**2.70**	**99.45**	37.15	87.63	24.10	94.12	21.32	93.74
KNN	79.24	73.21	43.84	88.07	61.54	80.64	2.76	99.36	46.43	89.31	28.89	93.80	26.03	94.16
DICE	72.21	71.81	61.80	82.21	67.01	77.01	33.25	90.83	36.77	90.04	48.67	83.00	39.56	87.95
ASH	87.87	59.38	82.12	68.99	84.99	64.18	64.75	79.46	77.83	73.94	76.40	75.88	72.99	76.43
GEN	55.48	84.76	**32.80**	**92.96**	44.14	88.86	3.89	98.85	29.31	93.51	**10.86**	**97.62**	14.69	96.66
CoRP	56.25	84.22	34.55	91.33	45.40	87.78	4.10	98.56	29.34	93.48	12.35	96.40	15.26	96.15
ENCORE (Ours)	54.05	86.19	33.06	92.75	**43.55**	**89.47**	3.19	99.18	28.84	94.07	11.83	97.43	**14.62**	**96.89**

References

[1] Mouïn Ben Ammar, Nacim Belkhir, Sebastian Popescu, Antoine Manzanera, and Gianni Franchi. Neco: Neural collapse based out-of-distribution detection, 2023. 4

[2] Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, Geeta Chauhan, Anjali Chourdia, Will Constable, Alban Desmaison, Zachary DeVito, Elias Ellison, Will Feng, Jiong Gong, Michael Gschwind, Brian Hirsh, Sherlock Huang, Kshiteej Kalambarkar, Laurent Kirsch, Michael Lazos, Mario Lezcano, Yanbo Liang, Jason Liang, Yinghai Lu, CK Luk, Bert Maher, Yunjie Pan, Christian Puhrsch, Matthias Reso, Mark Saroufim, Marcos Yukio Siraichi, Helen Suk, Michael Suo, Phil Tillet, Eikan Wang, Xiaodong Wang, William Wen, Shunting Zhang, Xu Zhao, Keren Zhou, Richard Zou, Ajit Mathews, Gregory Chanan, Peng Wu, and Soumith Chintala. Pytorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation. In *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24)*. ACM, Apr 2024. 2

[3] Abhijit Bendale and Terrance E. Boult. Towards open set deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1563–1572. IEEE, 2016. 2

[4] Andrija Djurisic, Nebojsa Bozanic, Arjun Ashok, and Rosanne Liu. Extremely simple activation shaping for out-of-distribution detection. In *The Eleventh International Conference on Learning Representations*, 2022. 2, 4

[5] Kun Fang, Qinghua Tao, Kexin Lv, Mingzhen He, Xiaolin Huang, and JIE YANG. Kernel pca for out-of-distribution detection. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 134317–134344. Curran Associates, Inc., 2024. 4

[6] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2016. 1, 2

[7] 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. 1

[8] Rui Huang, Andrew Geng, and Yixuan Li. On the importance of gradients for detecting distributional shifts in the wild. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 677–689. Curran Associates, Inc., 2021. 2, 3

[9] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31, 2018. 1, 2

[10] Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*, 2018. 1, 2

[11] Litian Liu and Yao Qin. Fast decision boundary based out-of-distribution detector. *ICML*, 2024. 3

[12] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in neural information processing systems*, 33:21464–21475, 2020. 1, 3

[13] Xixi Liu, Yaroslava Lochman, and Christopher Zach. Gen: Pushing the limits of softmax-based out-of-distribution detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23946–23955, 2023. 3

[14] Jie Ren, Peter J Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark Depristo, Joshua Dillon, and Balaji Lakshminarayanan. A simple fix to mahalanobis distance for improving near-ood detection. In *NeurIPS*, 2021. 2

[15] Chandramouli Shama Sastry and Sageev Oore. Detecting Out-of-Distribution Examples with Gram Matrices. In *International Conference on Machine Learning*, pages 8491–8501. PMLR, 2020. 2

[16] Yiyou Sun, Chuan Guo, and Yixuan Li. React: Out-of-distribution detection with rectified activations. *Advances in Neural Information Processing Systems*, 34:144–157, 2021. 2, 3

[17] Yiyou Sun, Yifei Ming, Xiaojin Zhu, and Yixuan Li. Out-of-distribution Detection with Deep Nearest Neighbors. *ICML*, 2022. 2

[18] Zhen Sun, Chuan Guo, and Yixuan Li. Dice: Distance-concentrated representation learning for out-of-distribution detection. *arXiv preprint arXiv:2303.08388*, 2023. 3

[19] Haoqi Wang, Zhizhong Li, Litong Feng, and Wayne Zhang. Vim: Out-of-distribution with virtual-logit matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4921–4930, 2022. 3

[20] Hongxin Wei, Renchunzi Xie, Hao Cheng, Lei Feng, Bo An, and Yixuan Li. Mitigating neural network overconfidence with logit normalization. In *International Conference on Machine Learning*, pages 23631–23644. PMLR, 2022. 1

[21] Kai Xu, Rongyu Chen, Gianni Franchi, and Angela Yao. Scaling for training time and post-hoc out-of-distribution detection enhancement. In *The Twelfth International Conference on Learning Representations*, 2024. 4

[22] Jingkang Yang, Pengyun Wang, Dejian Zou, Zitang Zhou, Kunyuan Ding, Wenxuan Peng, Haoqi Wang, Guangyao Chen, Bo Li, Yiyou Sun, Xuefeng Du, Kaiyang Zhou, Wayne Zhang, Dan Hendrycks, Yixuan Li, and Ziwei Liu. Openood: Benchmarking generalized out-of-distribution detection. 2022. 2

[23] Jingyang Zhang, Jingkang Yang, Pengyun Wang, Haoqi Wang, Yueqian Lin, Haoran Zhang, Yiyou Sun, Xuefeng Du, Yixuan Li, Ziwei Liu, Yiran Chen, and Hai Li. Openood v1.5: Enhanced benchmark for out-of-distribution detection. *arXiv preprint arXiv:2306.09301*, 2023. 2

[24] Zihan Zhang and Xiang Xiang. Decoupling maxlogit for out-of-distribution detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3388–3397, 2023. 1