IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 26, 2022, accepted February 17, 2022, date of publication February 22, 2022, date of current version March 4, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3153723

A Simple Framework for Robust Out-of-Distribution Detection

YOUNGBUM HUR[1], EUNHO YANG[2,3], AND SUNG JU HWANG[2,3]
[1]Department of Industrial Engineering, Inha University, Incheon 22212, South Korea
[2]School of Artificial Intelligence, Korea Advanced Institute of Science and Technology (KAIST), Daejeon 34141, South Korea
[3]AITRICS, Seoul 06627, South Korea

Corresponding author: Youngbum Hur (youngbum.hur@inha.ac.kr)

This work was supported in part by Inha University Research Grant.

ABSTRACT Out-of-distribution (OOD) detection, i.e., identifying whether a given test sample is drawn from outside the training distribution, is essential for a deep classifier to be deployed in a real-world application. The existing state-of-the-art methods of OOD detection tackle this issue by utilizing the internal feature of the classification network. However, we found that such detection methods inherently struggle to detect hard OOD images, i.e., drawn near from the training distribution: a naive softmax-based baseline even outperforms them. Motivated by this, we propose a simple yet effective training scheme for further calibrating the softmax probability of a classifier to achieve high OOD detection performance under both hard and easy scenarios. In particular, we suggest to optimize consistency regularization and self-supervised loss during training. Our experiments demonstrate the superiority of our simple method under various OOD detection scenarios.

INDEX TERMS Out-of-distribution detection, network calibration, deep neural networks, consistency regularization, self-supervised learning.

I. INTRODUCTION

Deep neural networks (DNNs) have demonstrated remarkable performance on many classification tasks such as image classification [8], medical diagnosis [3], and video prediction [32]. However, it is widely known that well trained deep classifiers are often overconfident even for novel examples, unseen during training [10].[1] This can become a serious problem when deployed in real-world vision systems [35], because the miss-prediction of DNN can cause a deadly accident when a self-driving car encounters a new/unseen street. To handle this overconfidence issue, the problem of detecting a test sample from out-of-distribution has gotten much attention recently [12], [19], [20], [28].

Given a deep image classifier, the conventional way for out-of-distribution (OOD) detection is to design a confidence score based on its softmax probability over classes. For example, an input is detected as OOD if the maximum probability among classes is low or the entropy of the softmax probability is high [12], [21]. Recently, more advanced OOD detection methods [20], [28] design a score using the information of internal layers and are claimed to achieve near-optimal detection rates, e.g., over 99% of area under the receiver operating characteristic curve (AUROC) for most tested cases.

A. CONTRIBUTION

First, we found that the existing state-of-the-art methods utilizing internal layers perform poorly on *hard OOD datasets*, i.e., drawn near from the training distribution: in our newly proposed hard OOD detection scenario, they perform even worse than the naive baseline using the maximum softmax probability [12] (see more details in Section III-A). This is because such hard OOD images are not clearly separable in the internal layers as they use similar (low-level) features with the training (in-distribution) images (see Figure 1). Instead, the final softmax probability is rather more informative to detect them, as it captures the most discriminative structural (high-level) information. This motivates us to revisit to the question, how to calibrate the softmax probability for improved OOD detection under both hard and easy scenarios.

In this paper, we suggest to optimize a simple *consistency regularization* term in addition to the standard training loss

The associate editor coordinating the review of this manuscript and approving it for publication was Li He.

[1]For example, DNNs trained to classify MNIST images often produce a high confident probability of 91% even for random noise [12].

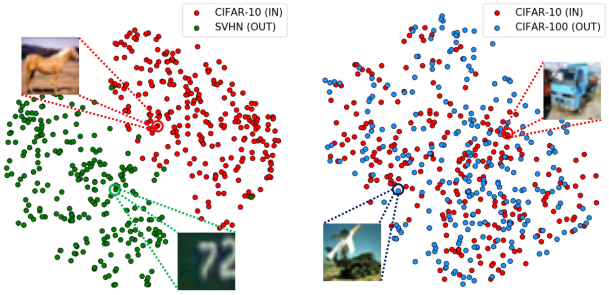


FIGURE 1. t-SNE visualization of an internal layer of ResNet-34 pre-trained on CIFAR-10 where SVHN (left) and CIFAR-100 (right) are used as easy and hard OOD, respectively.

(e.g., cross-entropy) to train a classifier with better calibrated softmax probability for OOD detection. To be specific, the proposed regularization loss forces the network to predict the consistent softmax probability over data augmentations, which injects a strong inductive bias to the model itself. Intuitively, the model learns more informative and consistent softmax probability aggregated over augmentations, which leads to improved calibration for OOD detection. To fully utilize the effect of our proposed training scheme, we also consider a test-time augmentation scheme (of hyperparameter-free) for inference. Finally, we show that our method can be further enhanced using a broader class of augmentations, e.g.rotation, whose predictions are not necessarily consistent: to incorporate them, one can use the self-supervised loss [13].

We verify the effectiveness of our method under various environments of detecting OOD, including standard classification (e.g., CIFAR-10 [17]) and fine-grained classification (e.g., CUB-200 [33]). Overall, our method achieves high performance for all tested datasets and especially shows robust results in hard OOD detection scenarios. In particular, our method improves the AUROC, compared to the baseline: 86.36% to 92.97% on CIFAR-10, when CIFAR-100 [17] is considered as OOD. We also demonstrate that our method improves the expected calibration error (ECE) [10], compared to the cross-entropy training by 3.64% to 1.51%, 20.35% to 6.78% on CIFAR-10 and CUB-200, respectively.

In summary, our major contributions are two-fold:

- We provide important observations that the existing state-of-the-art methods (based on the internal feature of the network) underperform than even a simple softmax-based baseline to detect hard OOD images.
- Motivated by this, we aim for training a better confidence-calibrated classifier with respect to the softmax probability for OOD detection, and found that a combination of consistency regularization and self-supervised loss over data augmentations is quite effective for the purpose.

We remark that detecting hard OOD samples is a more challenging and important but under-explored problem. We believe our new observations, experimental setups and method can be important guidelines when researchers pursue similar tasks in the future.

B. ORGANIZATION

In Section II, we discuss the related work. In Section III, we analyze the behavior of the easy and hard OOD datasets. Then we propose simple methods for OOD detection in Section IV. In Section V, we present the experimental results, and, in Section VI, we conclude our work.

II. RELATED WORK

A. OUT-OF-DISTRIBUTION DETECTION

In this paper, we focus on the out-of-distribution (OOD) detection task using a classification model when in-distribution (or training) samples are drawn from a multi-class dataset, which is a popular setup in the field [12], [19]–[21]. Here, the goal is two-fold: (*a*) predict the true label of the in-distribution sample, while (*b*) detecting OOD samples correctly.

One of the research directions tackles this problem by developing an inference method upon a pre-trained classifier. Hendrycks and Gimpel [12] study a baseline method for detecting OOD samples by utilizing the maximum softmax probability of the prediction, as an detection score. Liang *et al.* [21] make a sensible change to the baseline [12] by combining temperature scaling and adding small controlled perturbations to the input. Lee *et al.* [20] assume the data distribution in the representation as a class conditional Gaussian distribution and measures the Mahalanobis distance of the given sample as a detection score. Sastry and Oore [28] propose to detect OOD by identifying inconsistencies between activity patterns by Gram matrices and class predicted. Among the listed methods, some [20], [21], [28] require hyperparameters to choose where they are often selected upon the OOD validation set. This may be infeasible in practice. On the other hand, we only consider simpler hyperparameter-free inference schemes for our method. More importantly, as mentioned earlier, the inference methods which utilize the internal features of the network [20], [28] lead to degradation in detecting hard OOD samples.

Another research direction focuses on learning representation for discriminating in-and-out distributions [19]. Hendrycks *et al.* [13] utilize a self-supervised auxiliary loss such as rotation prediction for learning better representation. Tack *et al.* [30] use distribution-augmented contrastive learning for training and suggest a new detection score specific to their training scheme for OOD detection. Our method can also be categorized into a learning method which shows robust performance in detecting hard OOD samples. We note that [30] is highly sensitive to the augmentation policy (see Section V-B, Table 2c) and also need high cost for training,[2] while our simpler method shows robust results along the considered experiments (see Section V-B).

B. CONSISTENCY REGULARIZATION

The concept of the consistency regularization is to enforce the model output unchanged when the input is perturbed. Sajjadi *et al.* [27] first propose the concept of consistency

[2]It costs 15 times longer training time compared to our proposed method.

regularization. Tarvainen and Valpola [31] firstly encourage consistency between the prediction of the current network and the previous network of itself. Miyato *et al.* [23] use consistency regularization for unlabeled samples with its adversarial example. Berthelot *et al.* [1] measure the consistency between weakly and strongly augmented images. Meanwhile, Hendrycks *et al.* [14] propose a method to synthesize mixing multiple augmented images in addition to consistency loss, and show the method is robust at OOD generalization. On the other hand, we show consistency regularization helps confidence calibration for OOD detection.

III. TOWARDS BETTER OOD BENCHMARKS

This section points out that existing state-of-the-art methods [20], [28] in out-of-distribution (OOD) detection do not generalize well on detecting hard OOD datasets and show that the cause lies in their usages of the internal layer of the network. We first briefly explain the high-level concept of easy and hard OOD and then propose a new detection scenario, where the naive baseline [12] outperforms the existing advanced methods in Section III-A. In Section III-B, we give an intuitive explanation for the observations by visualizing the internal layer of the network with t-SNE [22].

A. EASY AND HARD OOD DATASETS

At a high level, we denote easy OOD as a sufficiently far away distribution from the training distribution, while hard OOD is not. We remark that detecting hard OOD is an important and essential component for a real-world deployment. Here, we propose a new hard OOD detection scenario coined "one-class remove CIFAR-10" which has never been explored in the literature. In this setup, a given class labeled samples of CIFAR-10 [17] becomes the OOD dataset, and samples with the remaining nine classes are used as training or in-distribution.

As shown in the Table 1, one can observe that the state-of-the-art methods [20], [28] (See Section VI-B for the details) show poor performance in all cases, while the simple baseline with maximum softmax probability [12] outperforms the others. This motivates us to take a closer look at the issue: the recent methods commonly use the ensembles of the internal layer feature for OOD detection.

B. ANALYSIS OF THE INTERNAL REPRESENTATION

To analyze the internal layer of the network, we visualize the data representation with t-SNE [22]. We select CIFAR-10 as in-distribution and train ResNet-34 [11] with standard cross-entropy loss. Then we choose SVHN [25], resized LSUN, and ImageNet [21] as easy OOD (the listed datasets are easily detected with simple statistics [30], e.g.., 95% area under the receiver operating characteristic curve (AUROC)). For hard OOD, we use CIFAR-100 [17], fixed versions[3] of LSUN, and ImageNet [30].

[3]New hard OOD benchmarks for CIFAR proposed by [30]

One can observe that easy OOD datasets are clearly separated with in-distribution datasets while hard OOD are still entangled in the internal layer (see Figure 2a, 2b). This observation aligns with our claim that the internal layer cannot capture the difference between hard OOD and in-distribution, as they are naturally expected to have similar low-level features. Meanwhile, both easy and hard OOD are separated well from the in-distribution in the penultimate layer of the network (see Figure 2c, 2d), as it captures the most discriminative structural (high-level) information. Motivated by this finding, we focus on developing a method for calibrating the softmax probability of the network for improved OOD detection.

IV. IMPROVING CONFIDENCE-CALIBRATION FOR HARD AND EASY OOD DETECTION

In this section, we revisit the softmax probability of a deep classifier, and introduce simple yet effective methods for confidence calibration to detect both hard and easy out-of-distribution (OOD) samples well. Our proposed framework is described in the following Sections IV-A, IV-B and IV-C.

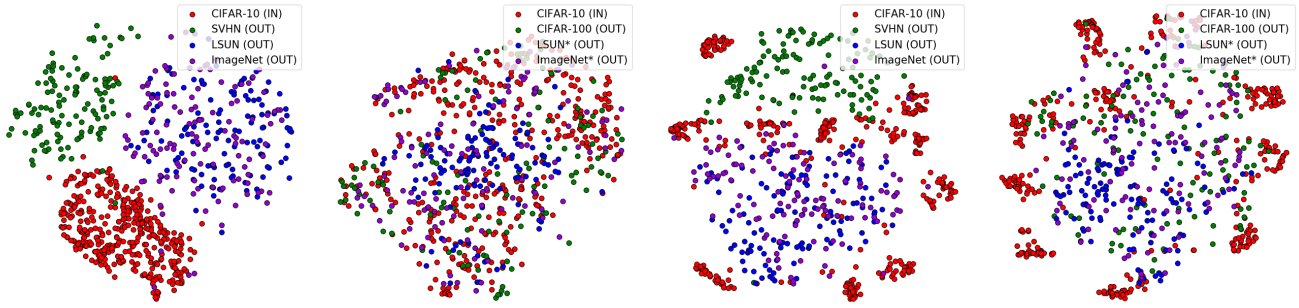
We consider a classification task with K classes dataset $\mathcal{D} = \{(x_m, y_m)\}_{m=1}^{M} \subseteq \mathcal{X} \times \mathcal{Y}$ where $\mathcal{Y} := \{1, \ldots, K\}$. Let $p_\theta(y|x)$ be a classifier that maps an input x to a softmax probability $y \in \mathbb{R}^K$. The goal of our work is to train a classifier with better calibrated softmax probability, $p_\theta(y|x)$ for OOD detection i.e., for a given OOD sample $x_{\texttt{out}}$, the classifier shows high entropy than the in-distribution sample $x_{\texttt{in}}$, $\mathcal{H}(p_\theta(y|x_{\texttt{out}})) > \mathcal{H}(p_\theta(y|x_{\texttt{in}}))$ where $\mathcal{H}$ is the entropy of the probability; $\mathcal{H}(p_\theta(y|x)) := -\sum_{i=1}^{K} p(y_i|x) \log(p(y_i|x))$ where $p(y_i|x)$ is the prediction probability for class i.

A. TRAINING: CONSISTENCY REGULARIZATION

The high-level idea of *consistency regularization* is to regularize the classifier's softmax probability to be consistent over the pre-defined augmentation policy. First, we define two types of augmentations, weak and strong where strong augmentation are likely to shift the input distribution [1]. Here, we force the prediction of a strongly augmented sample to mimic the prediction of a weakly augmented sample. Intuitively, such knowledge distillation will guide the model to learn a strong inductive bias. Formally, we define two augmentation family: $\mathcal{T}_{\texttt{weak}}$ and $\mathcal{T}_{\texttt{strong}}$. For a given labeled input $(x_{\texttt{in}}, y_{\texttt{in}}) \sim \mathcal{D}$, let x_w, x_s be augmented samples of $x_{\texttt{in}}$ under the pre-defined policy: $x_w = T_w(x_{\texttt{in}}), x_s = T_s(x_{\texttt{in}})$ where $T_w \sim \mathcal{T}_{\texttt{weak}}, T_s \sim \mathcal{T}_{\texttt{strong}}$. Then the propose regularization can be defined as follows:

$$\mathcal{L}_{\texttt{con}}(x_{\texttt{in}}) := \mathrm{KL}\Big(p_{\tilde{\theta}}(y|x_w) \parallel p_\theta(y|x_s)\Big) \tag{1}$$

where KL denotes the Kullback-Leibler (KL) divergence, and $\tilde{\theta}$ is a fixed copy of parameter θ for preventing model collapse by stopping the gradient propagation through $\tilde{\theta}$ [23], [34], [36]. We follow the data augmentation policy from [1] (see Section V-A for the details).



(a) Internal: easy OOD (b) Internal: hard OOD (c) Penultimate: easy OOD (d) Penultimate: hard OOD

FIGURE 2. The t-SNE visualization of the internal (residual block 2) and penultimate feature of ResNet-34 pre-trained on CIFAR-10 where SVHN, LSUN and ImageNet are used as easy OOD, and CIFAR-100, fixed version of LSUN and ImageNet are used as hard OOD. * denotes the fixed version dataset.

TABLE 1. AUROC (%) of ResNet-18 trained on one-class remove CIFAR-10: samples from the given class are considered as OOD, while the remaining samples of the nine classes are considered for training (in-distribution). "Inter." denotes the methods that utilize the internal feature of the network. "Ours" indicates the network trained on our objective, the extension version of consistency regularization, and tested with the proposed inference method, Augment-Entropy. The final column indicates the mean AUROC across all the classes and the bold denotes results within 1% from the highest result.

Method	Inter.	Plane	Car	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck	Mean
Baseline [12]	-	86.53	**74.67**	87.70	85.76	89.01	77.60	88.30	85.92	78.18	**82.86**	83.65
Mahalanobis [20]	✓	72.00	58.13	65.19	61.24	51.99	57.57	54.50	57.78	49.54	43.60	57.15
Gram matrix [28]	✓	78.28	62.63	73.04	65.77	72.31	57.68	69.27	62.39	63.06	55.65	66.01
Ours	-	**91.57**	**75.10**	**94.09**	**90.68**	**93.94**	**85.16**	**94.15**	**87.67**	**89.40**	**82.24**	**88.40**

B. INFERENCE: AUGMENT-ENTROPY

To further improve the softmax probability calibration of the network, we develop a test-time augmentation framework [16] coined, *Augment-Entropy*. For a given data point, the Augment-Entropy calculates the expectation of the softmax probability over the data augmentation space, then measures the entropy as a detection score. The high-level intuition of Augment-Entropy is that such ensemble prediction can smoothen the output probability which leads to a better calibration. For a given test input x and a pre-trained classifier θ, we calculate the following score for the OOD detection:

$$s_{\texttt{aug-ent}}(x) := \mathcal{H}\Big(\mathbb{E}_{T\sim\mathcal{T}}\Big[p_\theta\big(y|T(x)\big)\Big]\Big) \tag{2}$$

where $\mathcal{T}$ is a pre-defined augmentation family, and $\mathcal{H}$ is the entropy function. We approximate the proposed score (2) via Monte Carlo integration with n randomly sampled augmentations from $\mathcal{T}$. See the details of augmentation policy and sampling number in Section V-A.

C. EXTENSION VIA SELF-SUPERVISED AUGMENTATION

We further improve the model calibration by adapting broader class of data augmentations (e.g.., rotation [7]). In particular, we utilize the concept of self-supervised training from [13] by jointly optimizing the rotation prediction loss along with our objective (1). For a given labeled input $(x_{\texttt{in}}, y_{\texttt{in}}) \sim \mathcal{D}$, and a rotation augmentation R_r: rotate the image with r angle, where $r \in \mathcal{R} := \{0°, 90°, 180°, 270°\}$, the goal is to classify the applied augmentation angle r of the given sample. Namely, for a rotation prediction classifier $p_{\texttt{rot}}$: which shares the same penultimate feature with p_θ, and standard cross-entropy $\mathcal{L}_{\texttt{CE}}(x, y)$, the auxiliary self-supervised loss is as follows:

$$\mathcal{L}_{\texttt{rot}}(x_{\texttt{in}}) := \frac{1}{4}\sum_{\hat{r}\in\mathcal{R}} \mathcal{L}_{\texttt{CE}}\Big(p_{\texttt{rot}}\big(r|R_{\hat{r}}(x_{\texttt{in}})\big), \hat{r}\Big). \tag{3}$$

1) ROTATION AS AN UNLABELED SAMPLE

Note that, we do not train the classifier p_θ with the rotated sample (non-identity samples), since such augmentation is known to be harmful for learning representation when forced as the original label [4], [18]. Therefore, we rather assume the *rotated sample as an unlabeled sample* and train with consistency regularization (1). Note that such consistency regularization can be applicable for any unlabeled samples. For a given input x and the rotation angle $r \in \mathcal{R}\backslash\{0°\}$, the regularization is as follows:

$$\mathcal{L}_{\texttt{rot-unlabel}}(x_{\texttt{in}}) := \frac{1}{3}\sum_{r\in\mathcal{R}\backslash\{0°\}} \mathcal{L}_{\texttt{con}}(R_r(x_{\texttt{in}})) \tag{4}$$

Finally, the objective of the extended version of our proposed method can be defined by simply combining the defined objectives with the standard cross-entropy loss:

$$\begin{aligned}&\mathcal{L}_{\texttt{extend}}(x_{\texttt{in}}, y_{\texttt{in}})\\ &\quad := \mathcal{L}_{\texttt{CE}}(x_{\texttt{in}}, y_{\texttt{in}}) + \mathcal{L}_{\texttt{con}}(x_{\texttt{in}}) + \mathcal{L}_{\texttt{rot}}(x_{\texttt{in}})\\ &\qquad +\mathcal{L}_{\texttt{rot-unlabel}}(x_{\texttt{in}})\end{aligned} \tag{5}$$

We observed that combining such methods indeed helps the model to have more informative softmax probability, which leads the model to be more robust at detecting OOD samples.

2) INFERENCE METHOD

We simply adapt the rotation prediction loss as a additional detection score along with the Augment-Entropy. Namely, the new inference method is as follows:

$$s_{\texttt{extend}}(x) := s_{\texttt{aug-ent}}(x) + s_{\texttt{rot}}(x) \tag{6}$$

where $s_{\texttt{rot}} := \mathcal{L}_{\texttt{rot}}$ by following [13].

V. EXPERIMENTS

We evaluate our method on various out-of-distribution (OOD) detection scenarios and demonstrated the softmax probability calibration effect, on standard classification benchmark (e.g., CIFAR-10 [17], CIFAR-100 [17]), and fine-grained classification (e.g., CUB-200 [33]). To evaluate the detection and calibration performance, we mainly report the area under the receiver operating characteristic curve (AUROC) as a threshold-free evaluation metric for a detection score and the expected calibration error (ECE) [10], [24]. Here, ECE estimates whether a classifier can indicate when they are likely to be incorrect for test samples (from in-distribution) by measuring the difference between prediction confidence and accuracy. The formal description of the metrics can be found in the Appendix. Overall, our results clearly demonstrate that consistency regularization for classifiers improves the performance of detecting hard OOD samples and shows significant improvement of the confidence calibration in all tested datasets. Additionally, we verify our assumption in Section III-A that the existing state-of-the-art methods have difficulties detecting hard OOD samples. We also give an ablation study to verify the components of our work.

A. EXPERIMENTAL SETUP

1) NOTATION

Unless otherwise specified, the following expression does not change during the experimental section. "Consist.", and "Rotate" denotes the training method that is jointly optimized standard cross-entropy with consistency loss (1) and, rotation loss (3), respectively. "Consist.+Rotate" denotes, the objective of the extension (5). For inference method, "Aug-Ent.", "Rotate", and "Aug-Ent. + Rotate" denote the Augment-Entropy (2), rotation loss (3) and the score for the extension (6), respectively.

2) BASELINE METHODS

We consider a wide range of OOD detection methods, which are largely categorized into two, inference methods: used a pre-trained model, and training methods: learn to discriminate in- vs. out-of-distributions (most of the training methods suggest their own inference method). Among the inference methods, we compare with (*a*) baseline [12], (*b*) ODIN [21], (*c*) Mahalanobis [20], and (*d*) Gram matrix [28]. The details about baseline methods can be found in VI-B.

For the training methods, we consider baselines as listed in what follows: (*a*) Cross-Ent.: training on the cross-entropy loss, (*b*) Rotate [13], and (*c*) CSI [30].

In this paper, we *do not assume an OOD validation set is available* at test-time, while some baseline methods such as ODIN, Mahalanobis, and Gram matrix,[4] are known to select hyperparameters, possibly by using the OOD validation set. We believe assuming the OOD validation set at test-time is not a realistic detection scenario (see Section V-C for a more comprehensive discussion). Therefore, we assume no such validation set and use a fixed hyperparameter for each baseline over all experiments unless otherwise stated: to be fair comparisons as much as possible, we choose the hyperparameter values recommended in the original papers: $T = 1000$, $\epsilon = 0.0012$ for ODIN, the uniform internal layer ensemble weight for Mahalanobis,[5] and same set of orders $P = \{1, \ldots, 10\}$ of the Gram matrix.

3) MODELS AND DATA AUGMENTATIONS

We use ResNet-18 and ResNet-34 [11] architecture for all experiments in our paper. For weak augmentation, we use random crop with padding, horizontal flip in standard classification, and Inception crop [29], horizontal flip in fine-grained classification. For all datasets, we use RandAugment [5] and Cutout [6] additionally for a strong augmentation by following [1].

4) TRAINING DETAIL

For CIFAR-10 and CIFAR-100 image classification, we train the model for 200 epochs with batch size 128, using stochastic gradient descent with momentum 0.9 and weight decay with 0.0001. The learning rate starts at 0.1 and is dropped by a factor of 10 at 50%, and 75%, of the training progress. For the fine-grained dataset, we follow the same training process expect with a batch size of 32, due to the smaller number of training samples.

5) INFERENCE DETAIL

Unless otherwise noted, we set the sampling number $n = 4$ for the Augment-Entropy approximation. For the data augmentation family $\mathcal{T}$ for equation (2), we choose random crop with horizontal flip.

B. MAIN RESULTS

1) STANDARD CLASSIFICATION

We start by considering the standard classification setup: we assume that in-distribution samples are from a specific multi-class dataset and train a classifier, then test on various external datasets as out-of-distribution. In addition to the conventional OOD detection setup [20], [21]; where

[4]The Gram matrix also requires hyperparameters. The method computes the p-th order Gram matrix [28] over the pre-defined set $p \in P$, and we observed that P can be highly sensitive to choose in some cases. We discuss more in the Appendix

[5]We also applied the Mahalanobis distance using the final layer only, but it is worse than the result using uniform ensemble in overall.

TABLE 2. AUROC (%) of ResNet trained on CIFAR-10, CIFAR-100, and CUB-200. Here, we consider various *easy and hard OOD* detection scenarios. "Consist.", "Consist. + Rotate", "Cross-Ent.", "Aug-Ent.", "Inter.", "HParam.", and "Cost." denote the proposed consistency regularization, extended version of consistency regularization, the cross-entropy, the Augment-Entropy, methods that utilize the internal feature of the network, methods that require hyperparameters, and the relative training cost compare to the standard cross-entropy training, respectively. The datasets with * indicate the fixed version (i.e.hard OOD). The final column indicates the mean AUROC across all the OOD datasets and the bold denotes results within 1% from the highest result.

					CIFAR-10 →			CIFAR-100 →		
Train	Inference	Inter.	HParam.	Cost.	SVHN	LSUN	ImageNet	SVHN	LSUN	ImageNet
Cross-Ent.	Baseline [12]	-	-	×1	88.03	87.45	84.58	85.17	71.51	71.82
Cross-Ent.	ODIN [21]	-	✓	×1	76.09	89.20	83.37	89.65	78.49	77.76
Cross-Ent.	Mahalanobis [20]	✓	✓	×1	94.98	96.74	96.35	85.33	**95.22**	**95.24**
Cross-Ent.	Gram matrix [28]	✓	✓	×1	**99.41**	**98.85**	**98.22**	**96.97**	**95.79**	**95.12**
CSI [30]	Baseline [12]	-	-	×30	98.07	97.69	**97.61**	86.25	86.13	84.85
CSI [30]	CSI-ens [30]	-	-	×30	**98.97**	**98.68**	**98.56**	88.70	88.71	87.13
Rotate [13]	Rotate [13]	-	-	×4	**98.64**	94.99	94.48	94.12	90.90	92.09
Consist.	Aug-Ent.	-	-	×2	97.15	96.74	95.73	86.23	84.95	84.46
Consist. + Rotate	Aug-Ent. + Rotate	-	-	×8	**98.80**	96.66	97.22	95.34	90.60	92.22

(a) ResNet-34 trained on CIFAR-10 / CIFAR-100 and tested under various *easy OOD* datasets.

					CIFAR-10 →						
Train	Inference	Inter.	HParam.	Cost.	LSUN*	ImageNet*	CIFAR-100	Food-101	Caltech-256	MIT-67	Mean
Cross-Ent.	Baseline [12]	-	-	×1	89.73	88.29	86.36	86.56	85.75	89.62	87.72
Cross-Ent.	ODIN [21]	-	✓	×1	85.55	81.00	76.94	76.65	80.76	85.71	81.10
Cross-Ent.	Mahalanobis [20]	✓	✓	×1	77.76	81.74	78.74	85.76	82.67	81.41	81.35
Cross-Ent.	Gram matrix [28]	✓	✓	×1	76.81	77.70	73.75	73.05	78.44	78.79	76.42
CSI [30]	Baseline [12]	-	-	×30	**94.81**	95.22	92.67	95.63	92.18	**96.29**	94.47
CSI [30]	CSI-ens [30]	-	-	×30	**95.78**	**96.26**	**93.91**	**96.52**	93.27	**97.09**	**95.47**
Rotate [13]	Rotate [13]	-	-	×4	86.32	89.13	85.06	93.83	87.11	91.25	88.78
Consist.	Aug-Ent.	-	-	×2	93.84	94.07	**93.32**	92.63	**94.39**	95.35	93.93
Consist. + Rotate	Aug-Ent. + Rotate	-	-	×8	94.14	**95.29**	**92.97**	**97.51**	**94.49**	**96.73**	**95.19**

(b) ResNet-34 trained on CIFAR-10 and tested under various *hard OOD* datasets.

					CUB-200 →					
Train	Inference	Inter.	HParam.	Cost.	Food-101	Caltech-256	MIT-67	Places-365	Dogs	Mean
Cross-Ent.	Baseline [12]	-	-	×1	72.71	73.41	75.16	75.03	75.09	74.28
Cross-Ent.	ODIN [21]	-	✓	×1	79.80	79.95	82.47	81.99	79.58	80.76
Cross-Ent.	Mahalanobis [20]	✓	✓	×1	84.20	81.24	78.70	76.36	62.32	76.56
Cross-Ent.	Gram matrix [28]	✓	✓	×1	81.04	81.74	74.64	74.00	72.15	76.71
CSI [30]	Baseline [12]	-	-	×30	75.99	76.81	84.45	81.22	69.09	77.51
CSI [30]	CSI-ens [30]	-	-	×30	82.33	82.84	92.97	89.64	76.34	84.82
Rotate [13]	Rotate [13]	-	-	×4	94.83	93.62	96.14	94.95	93.80	94.67
Consist.	Aug-Ent.	-	-	×2	87.17	86.08	88.58	87.17	88.84	87.57
Consist. + Rotate	Aug-Ent. + Rotate	-	-	×8	**97.12**	**96.45**	**97.98**	**97.01**	**97.38**	**97.19**

(c) ResNet-18 trained on CUB-200 and tested under various *hard OOD* datasets.

CIFAR-10 [17] is in-distribution dataset while SVHN [25], resized LSUN and ImageNet [21] are out-of-distribution, we additionally consider the following hard OOD datasets: CIFAR-100, and fixed LSUN* and ImageNet* [30], Food-101 [2], Caltech-256 [9], and MIT-67 [26]. Note that we followed the same resizing operation as Tack *et al.* [30] for high resolution datasets to avoid generating easy OOD datasets.

The results in Table 2 support our belief that the state-of-the-art inference methods that utilize the internal feature

TABLE 3. Test accuracy (%) and ECE (%) of classifiers trained on various image classification tasks. We train ResNet-34 for CIFAR datasets and ResNet-18 for the fine-grained datasets. "Consist. + Rotate" denotes the extended version of consistency regularization. The arrow on the right side of the evaluation metric indicates the ascending or descending order of the value. Values in parentheses indicate the relative rate of the performance from the cross-entropy, and the bold denotes the best results.

Evaluation metric	Train method	CIFAR-10	CIFAR-100	CUB-200
Test accuracy ↑	Cross-entropy	94.75	73.94	52.71
	Rotate [13]	96.72	77.47	60.98
	CSI [30]	95.58	77.94	43.34
	Consist. + Rotate (ours)	**96.79 (+2.15%)**	**80.78 (+9.25%)**	**61.63 (+16.92%)**
ECE ↓	Cross-entropy	3.64	11.81	20.35
	Rotate [13]	2.65	11.88	17.49
	CSI [30]	2.37	8.56	26.21
	Consist. + Rotate (ours)	**1.51 (-58.51%)**	**8.17 (-30.82%)**	**6.78 (-66.68%)**

TABLE 4. Ablation study on each component of our proposed training loss. We measure test accuracy (%), ECE (%), and AUROC (%) where CIFAR-100 is in-distribution, and CIFAR-10 is out-of-distribution. For all objective, we used the same score (i.e., baseline [12]) and network architecture (i.e., ResNet-34). No checkmark and full checkmark denote the standard cross-entropy training, and the extended version (5), respectively. The best results are denoted in bold.

Consist. ($\mathcal{L}_{\texttt{con}}$)	Rotation ($\mathcal{L}_{\texttt{rot}}$)	Rot-unlabel. ($\mathcal{L}_{\texttt{rot-unlabel}}$)	Test acc	ECE	AUROC
-	-	-	73.94	11.81	76.2
✓	-	-	77.81	10.73	77.7
-	✓	-	77.47	11.88	76.94
✓	✓	-	80.65	9.87	**78.75**
✓	✓	✓	**80.78**	**8.17**	78.55

of the deep network (e.g., Mahalanobis and Gram matrix) do not generalize in detecting hard OOD datasets (e.g., LSUN*). While our proposed method significantly outperforms the above inference methods in hard OOD detection while achieving comparable results in easy OOD detection. By combining our method with auxiliary rotation loss, we achieve comparable results to CSI, which is known to be one of the most strong baselines, while the training computation is much efficient (73.33% less training time[6]).

2) FINE-GRAINED CLASSIFICATION

In this setup, we assume training a classifier with in-distribution samples that are from a fine-grained dataset. We run our experiments with CUB-200 [33] as in-distribution while the following datasets are considered as out-of-distribution: Food-101 [2], Caltech-256 [9], MIT-67 [26], and Dogs [15].

Overall, our proposed method significantly outperforms prior methods in all datasets tested as shown in Table 2c. We remark that consistency regularization significantly and consistently improves the performance of OOD detection. Interestingly, one can observe that CSI is not as effective as the previous results under the standard classification benchmark; Table 2b. This is because contrastive learning performance is sensitive to the data augmentation policy and should use the suitable policy for each dataset. The proposed policy from CSI fails to generalize in the current setup. On the other hand, our method performs well, even in the fine-grained datasets.

[6]Without rotation loss, our method costs 93.33% less training time.

TABLE 5. Ablation study on the effect of sampling number n for Augment-Entropy approximation. We measure AUROC (%) where CIFAR-100 is in-distribution, and CIFAR-10 is out-of-distribution, and the network is ResNet-34.

n	1	2	4	8	16
AUROC	78.77	79.61	80.20	80.25	80.35

3) TEST ACCURACY AND CALIBRATION

We demonstrate the effectiveness of our method on prediction and calibration performance under various image classification datasets: CIFAR-10, CIFAR-100, and CUB-200. As shown in Table 3, the proposed consistency regularization significantly and consistently improves both prediction accuracy and confidence calibration. We remark that the proposed method is specialized for calibration: other methods (e.g., rotation) also increase the accuracy, but the calibration may not improve.

TABLE 6. AUROC (%) value of detection under logistic regression trained using an OOD validation set. We train the logistic regression upon the same pre-trained network (i.e., ResNet-34) and validation set from the previous work [20].

In-distribution	OOD	AUROC
CIFAR-10	SVHN	98.9
	ImageNet	99.0
	LSUN	99.4
CIFAR-100	SVHN	97.8
	ImageNet	96.3
	LSUN	98.4

C. ABLATION STUDY

1) COMPONENT ANALYSIS

In Table 4, we assess the individual effects of each component of our final training objective and adding consistency loss improves all metrics such as test accuracy, calibration error and AUROC. Even though adding each component might not always improve the all metrics, we observe that adding components in the training objective is beneficial in most cases.

2) EFFECT OF THE SAMPLING NUMBER

In Table 5, we investigate an effect of the sampling number used for approximating Augment-Entropy (2). Surprisingly, we found that the sampling number of 4 is sufficient. The reason is that our consistency regularization forces the augmentation to be concentrated. As a result, we can effectively approximate the expectation with a small number of samples. Therefore, our method can be effectively used at a small cost.

3) OOD VALIDATION SET

We give empirical evidence that one should avoid using an out-of-distribution validation set. In such a detection scenario, we observed that the problem can be solved by a straightforward method, and it is comparable to most of the baseline work. The high-level idea is to learn a logistic regression on top of the pre-trained representation by utilizing the OOD validation set. Consider a depth-d pre-trained network which output for given input x is $f(x; W_{1:d}) = W_d\sigma(\ldots\sigma(W_1x))$ where W_i is a i^{th} layer weight and σ is an activation function. We then aggregate (i.e., concatenate) all the internal feature and use as the input feature for logistic regression: $x_{\texttt{input}} = \texttt{aggregate}(\{f(x; W_{1:i})\}_{i=1}^{d})$. Finally, we train a logistic regression function with the extracted features of in-and-out validation samples. For training the logistic regression, we use the same pre-trained network (i.e., ResNet-34) and validation set from the previous work [20].[7]

The result in Table 6 shows that the simple regression achieves very high AUROC performance (over 96%) for popular OOD setups, even comparable to the state-of-the-art results when the OOD validation set is used. Note that the logistic regression achieves the result without even careful input preprocessing [20], [21] used in the prior work.

VI. CONCLUSION

In this paper, we show that the existing OOD detection methods that are using the internal feature of the network perform poorly for hard OOD images. Motivated by this, we propose a simple yet effective training scheme for better OOD detection and network calibration. We evaluate our methods under various scenarios and show that our proposed method is robust for easy and hard OOD datasets. Due to the simplicity of our method, we think it could enjoy a broader usage under various applications in the future.

One can view the OOD detection task as a binary classification, where datasets of one of two classes are not available at training time. Hence, which types of unseen (OOD) test datasets are used in evaluation is very important to measure the superiority of a method. We believe that our work sheds a new angle for the sensitive, yet important issue: our new benchmark setups can guide other researchers to further investigate more realistic out-of-distribution detection tasks.

APPENDIX A
EXPERIMENTAL DETAILS

We provide the implementation of our framework in the supplementary material. For all baseline methods, we use the official implementation and the same network architecture.

A. DATASET DETAILS

For in-distribution datasets, we consider CIFAR-10 [17], CIFAR-100 [17], and CUB-200 [33]. CIFAR-10 and CIFAR-100 consist of 50,000 training and 10,000 test images with 10 and 100 image classes, respectively. CUB-200 contains 200 species (classes) of birds, each with roughly 30 training images (total 5,994) and 30 testing images (total 5,794).

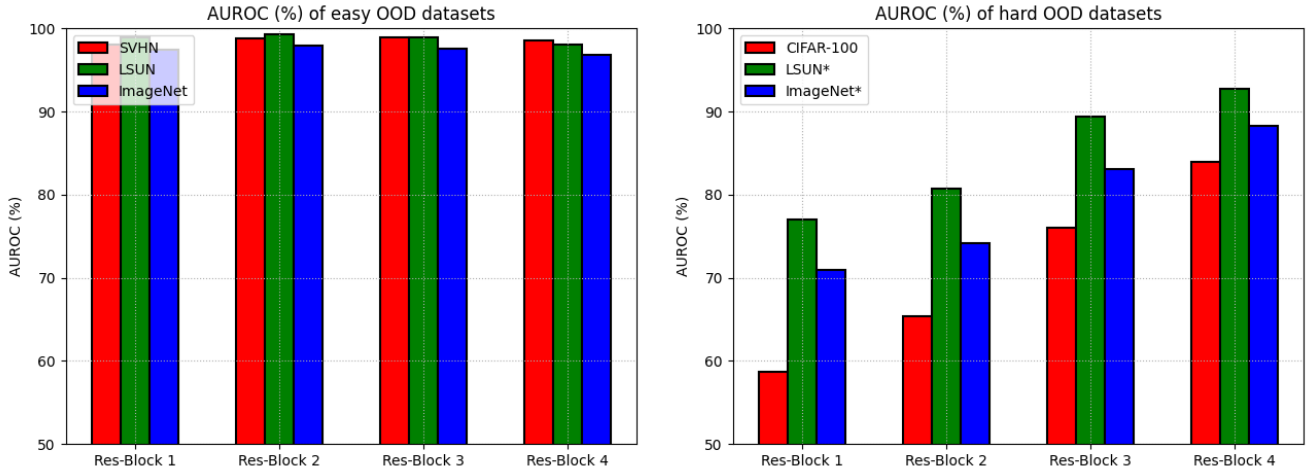
For CIFAR datasets, the out-of-distribution (OOD) datasets are as follows: SVHN [25] consists of 26,032 test images with 10 digits, resized LSUN [21] consists of 10,000 test images of 10 different scenes, resized ImageNet [21] consists of 10,000 test images with 200 images classes from a subset of full ImageNet dataset, fixed version of LSUN* [30][8] consists of 10,000 test images of 10 different scenes, fixed version of ImageNet* [30] consists of 10,000 test images with 30 images classes from a subset of full ImageNet dataset, Food-101 [2] consists of 101 food categories with 101,000 images, Caltech-256 [9] consists of object categories containing a total of 30,607 images, and MIT-67 [26] consists of 67 Indoor categories, and a total of 15,620 images. We resized the high images into 32 by 32 resolution by using the correct resizing operation torchvision.transforms.Resize() by

[7]The pre-trained network and validation set are available at https://github.com/pokaxpoka/deep_Mahalanobis_detector.

[8]use PyTorch torchvision.transforms.Resize() operation for correct resizing.

TABLE 7. AUROC (%) based on the hyperparameter selection of the Gram matrix. We use ResNet-18 trained on CUB-200 with standard cross-entropy loss. The final column indicates the mean AUROC across all OOD datasets.

	CUB-200 →					
Hyperparameter	Food-101	Caltech-256	MIT-67	Places-365	Dogs	Mean
$P = \{1\}$	81.04	81.74	74.64	74.00	72.15	76.71
$P = \{1, \ldots, 9\}$	32.27	32.42	30.55	30.47	31.00	31.34
$P = \{1, \ldots, 10\}$	0.00	0.00	0.00	0.00	0.00	0.00



(a) Easy OOD detection result

(b) Hard OOD detection result

FIGURE 3. AUROC (%) of the logistic regression trained upon each residual block (i.e., internal representation) of the ResNet-34. ResNet-34 is pre-trained on CIFAR-10, and for logistic regression training, we use in-and-out distribution validation sets. * denotes the fixed version dataset (i.e., hard OOD dataset), and "Res-Block *i*" indicates the *i*-th residual block of the ResNet.

following [30], and center crop; Food-101, Caltech-256, and MIT-67 are high-resolution datasets.

For CUB-200 dataset, the considered out-of-distribution datasets are as follows: Food-101, Caltech-256, MIT-67, Places-365 [37] with small images (256 * 256) validation set contains 36,500 images of scene categories, and Stanford Dogs [15] which consists 120 classes, and a total of 20,580 images.

B. EVALUATION METRICS

For evaluation, we measure the two metrics that each measures (a) the effectiveness of the proposed score in distinguishing in- and out-of-distribution images, (b) the confidence calibration of softmax classifier.

- **Area under the receiver operating characteristic curve (AUROC).** Let TP, TN, FP, and FN denote true positive, true negative, false positive and false negative, respectively. The ROC curve is a graph plotting true positive rate = TP / (TP+FN) against the false positive rate = FP / (FP+TN) by varying a threshold.
- **Expected calibration error (ECE).** For a given test data $\{(x_n, y_n)\}_{n=1}^{N}$, we group the predictions into M interval bins (each of size $1/M$). Let B_m be the set of indices of samples whose prediction confidence falls into the interval $(\frac{m-1}{M}, \frac{m}{M}]$. Then, the expected calibration error (ECE) [10], [24] is follows:

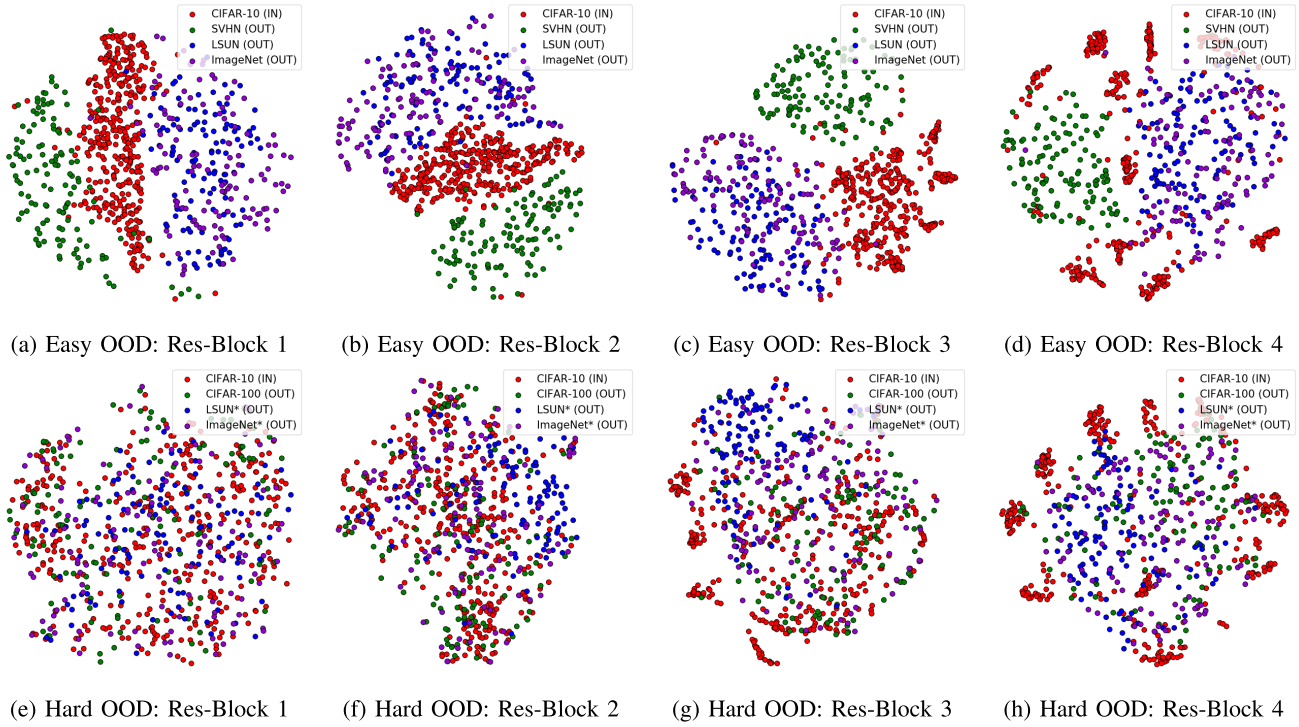
$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{N} |\mathrm{acc}(B_m) - \mathrm{conf}(B_m)|, \tag{7}$$

where $\mathrm{acc}(B_m)$ is accuracy of B_m: $\mathrm{acc}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \mathbb{1}_{\{y_i = \arg\max_y p(y|x_i)\}}$ where $\mathbb{1}$ is indicator function and $\mathrm{conf}(B_m)$ is confidence of B_m: $\mathrm{conf}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} q(x_i)$ where $q(x_i)$ is the confidence of data x_i. For all experiments, we set the number of bins to 20 i.e., $M = 20$.

APPENDIX B
BASELINES

In this section, we review the baseline inference methods of the out-of-distribution (OOD) detection, i.e., baseline [12], ODIN [21], Mahalanobis [20] and Gram matrix [28]. Then we show that the detection through the Gram matrix is sensitive to hyperparameter choices in some cases.

The goal of the inference method is to design a score function s upon the pre-trained classifier p_θ, so that the score of the in-distribution sample $x_{\mathtt{in}}$ is higher than the score of the OOD sample $x_{\mathtt{out}}$, i.e., $s(x_{\mathtt{in}}) > s(x_{\mathtt{out}})$. Some of



(a) Easy OOD: Res-Block 1 (b) Easy OOD: Res-Block 2 (c) Easy OOD: Res-Block 3 (d) Easy OOD: Res-Block 4

(e) Hard OOD: Res-Block 1 (f) Hard OOD: Res-Block 2 (g) Hard OOD: Res-Block 3 (h) Hard OOD: Res-Block 4

FIGURE 4. The t-SNE visualization at each residual block (i.e., internal representation) of the ResNet-34, which is pre-trained on CIFAR-10. SVHN, resized LSUN and ImageNet are used as easy OOD, while CIFAR-100, fixed version of LSUN and ImageNet are used as hard OOD. * denotes the fixed version dataset (i.e., hard OOD dataset), and "Res-Block *i*" indicates the *i*-th residual block of the ResNet.

the inference methods require an OOD validation set for the hyperparameter selection; however, in this paper, we do not assume any OOD validation set is available at test-time. The following is the definition of each score function and how to select hyperparameters without any OOD validation set.

- **Baseline** [12]**.** They use a maximum value of softmax probablity as a score function. For a given input x the score s is as follows: $s(x) := \max_i p_\theta(y_i|x)$ where $p_\theta(y_i|x) = \frac{\exp(g_i(x))}{\sum_j \exp(g_j(x)))}$ is the prediction probability of the class i of a pre-trained classifier p_θ, and g_i is the logit value of the class i.
- **ODIN** [21]**.** They use a mix of temperature scaling at the softmax layer and input perturbations and also use a maximum value of softmax probability as a confidence score. For a given input x, temperature T, and input perturbation scale ϵ, the score s is as follows: $s(x) := \max_i p_{\tilde{\theta}}(y_i|\tilde{x})$ where $p_{\tilde{\theta}}(y_i|x) = \frac{\exp(g_i(x)/T)}{\sum_j \exp(g_j(x)/T))}$, and $\tilde{x} = x - \epsilon \text{sign}(-\nabla_x \log \max_i p_{\tilde{\theta}}(y_i|x))$. For all experiments, we fix the hyperparameter values; $T = 1000$, and $\epsilon = 0.0012$.
- **Mahalanobis** [20]**.** They compute the Mahalanobis distance between test sample's feature representations and the class-conditional Gaussian distribution at each layer, then they represent each sample as a vector of the Mahalanobis distances. Here, we assume to have weights of each layer's contribution α_l, and the mean and covariance of the Gaussian distribution $\hat{\mu}_{l,i}, \widehat{\Sigma}_l$ where l, i denotes the index of the layer and the class, respectively (assume that the covariance is the same for all classes). Then the detection score is as follows: $s(x) := \sum_l \alpha_l M_l$ where $M_l = \max_i -(f_l(x) - \hat{\mu}_{l,i})^T \widehat{\Sigma}_l^{-1}(f_l(x) - \hat{\mu}_{l,i})$, and f_l is the internal layer activation of index l. In this paper, we assume uniform ensemble of each layer, i.e., the contribution of the layers are equal $\forall l$, $\alpha_l = 1$.
- **Gram matrix** [28]**.** The high-level idea of Gram matrices is to identify inconsistency between activity patterns and predicted class. They detect anomalies in the Gram matrices by comparing each value with its respective range observed over the training data. In the paper, they introduce p-th order Gram matrix: $G_l^p = \left(f_l^p f_l^{p\top}\right)^{1/p}$ where f_l denotes the l-th layer activation for given input x (see the paper for the details). To achieve the final detection score, they compute the p-th order detection score over all $p \in P$. Following the paper, we set $P = \{1, \ldots, 10\}$ for CIFAR datasets. For CUB-200 dataset, we set the hyperparameter as $P = \{1\}$.

Hyperparameter sensitivity of the Gram matrix. We observed that the Gram matrix also requires hyperparameter selection, which is highly sensitive in some cases.[9] As shown in Table 7, the Gram matrix fails to detect OOD samples (i.e., AUROC lower than 50, which is a random guess) with the recommended hyperparameter (i.e., $P = \{1, \ldots, 10\}$),

[9]We use the official implementation from https://github.com/VectorInstitute/gram-ood-detection

when CUB-200 is in-distribution. Note that in CUB-200 the detection performance degrades for higher-order Gram matrix $p \in P$, which is an opposite observation from [28]. We therefore simply fix $P = \{1\}$, for CUB-200 experiments.

APPENDIX C
MORE DISCUSSION ON OUT-OF-DISTRIBUTION BENCHMARKS

In this section, we provide a quantitative result, and more visualization that supports our hypothesis in Section III; easy out-of-distribution (OOD) datasets are well separated in the internal layer, while hard OOD datasets are not. We train logistic regression on the internal represention of the classifier, to see how much the in-and-out distribution is separated. To train the regression model, we use 1,000 in-and-out distribution validation samples each and compute the AUROC on the remaining test set as a measurement of separation.

For a given validation set $\mathcal{X}_{\texttt{in}}, \mathcal{X}_{\texttt{out}}$, we extract the feature map of the i-th residual block f_i as follows: $\mathcal{D}^i_{\texttt{inter}} := \{(f_i(x_{\texttt{in}}), 1) | x_{\texttt{in}} \in \mathcal{X}_{\texttt{in}}\} \bigcup \{(f_i(x_{\texttt{out}}), 0) | x_{\texttt{out}} \in \mathcal{X}_{\texttt{out}}\}$. Then for a given $(f_i(\hat{x}), \hat{y}) \sim \mathcal{D}^i_{\texttt{inter}}$, we train the logistic regression p_ϕ with the following loss: $\mathcal{L}_{\texttt{BCE}}(p_\phi(y|f_i(\hat{x})), \hat{y})$ where $\mathcal{L}_{\texttt{BCE}}$ is a binary cross-entropy loss. At the inference time, for a given test sample x, we directly use the prediction of the regression model as the detection score: $s(x) := p_\phi(y|f_i(x))$.

As shown in Figure 3, easy OOD samples are well separated across the entire layer, while hard OOD samples are better separated from the penultimate layer. This result supports our assumption in Section III; some recent methodologies fail to detect hard OOD due to the internal layer usage. We also provide t-SNE visualization of OOD datasets across all residual blocks in Figure 4, which show a consistent observation.

ACKNOWLEDGMENT

The authors gratefully thank Prof. Jinwoo Shin and his Ph.D. student Jihoon Tack at the KAIST for their useful suggestions and valuable comments.

REFERENCES

[1] D. Berthelot, N. Carlini, E. D. Cubuk, A. Kurakin, K. Sohn, H. Zhang, and C. Raffel, "Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–13.

[2] L. Bossard, M. Guillaumin, and L. Van Gool, "Food-101–mining discriminative components with random forests," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 446–461.

[3] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad, "Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2015, pp. 1721–1730.

[4] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.

[5] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "RandAugment: Practical automated data augmentation with a reduced search space," 2019, *arXiv:1909.13719*.

[6] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," 2017, *arXiv:1708.04552*.

[7] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–16.

[8] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[9] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," California Inst. Technol., Pasadena, CA, USA, Tech. Rep., 2007.

[10] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1321–1330.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[12] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–12.

[13] D. Hendrycks, M. Mazeika, S. Kadavath, and D. Song, "Using self-supervised learning can improve model robustness and uncertainty," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 1–12.

[14] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan, "Augmix: A simple method to improve robustness and uncertainty under data shift," in *Proc. Int. Conf. Learn. Represent.*, 2020.

[15] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei, "Novel dataset for fine-grained image categorization: Stanford dogs," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 1–5.

[16] I. Kim, Y. Kim, and S. Kim, "Learning loss for test-time augmentation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 4163–4174.

[17] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 2009.

[18] H. Lee, S. J. Hwang, and J. Shin, "Self-supervised label augmentation via input transformations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 5714–5724.

[19] K. Lee, H. Lee, K. Lee, and J. Shin, "Training confidence-calibrated classifiers for detecting out-of-distribution samples," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–16.

[20] K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1–11.

[21] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–11.

[22] L. V. D. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.

[23] T. Miyato, S.-I. Maeda, M. Koyama, and S. Ishii, "Virtual adversarial training: A regularization method for supervised and semi-supervised learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1979–1993, Aug. 2018.

[24] M. P. Naeini, G. Cooper, and M. Hauskrecht, "Obtaining well calibrated probabilities using Bayesian binning," in *Proc. AAAI Conf. Artif. Intell.*, 2015, pp. 1–7.

[25] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 1–9.

[26] A. Quattoni and A. Torralba, "Recognizing indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 413–420.

[27] M. Sajjadi, M. Javanmardi, and T. Tasdizen, "Regularization with stochastic transformations and perturbations for deep semi-supervised learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 1–9.

[28] C. S. Sastry and S. Oore, "Detecting out-of-distribution examples with Gram matrices," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 8491–8501.

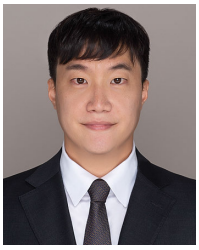
[29] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, Oct. 2015, pp. 1–9.

[30] J. Tack, S. Mo, J. Jeong, and J. Shin, "CSI: Novelty detection via contrastive learning on distributionally shifted instances," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp 11839–11852.

[31] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1–10.

[32] R. Villegas, J. Yang, Y. Zou, S. Sohn, X. Lin, and H. Lee, "Learning to generate long-term future via hierarchical prediction," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3560–3569.

[33] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD birds-200-2011 dataset," California Inst. Technol., Pasadena, CA, USA, Tech. Rep. CNS-TR-2011-001, 2011.
[34] Q. Xie, Z. Dai, E. Hovy, M.-T. Luong, and Q. V. Le, "Unsupervised data augmentation for consistency training," 2019, *arXiv:1904.12848*.
[35] R. V. Yampolskiy and M. S. Spellchecker, "Artificial intelligence safety and cybersecurity: A timeline of ai failures," 2016, *arXiv:1610.07997*.
[36] S. Yun, J. Park, K. Lee, and J. Shin, "Regularizing class-wise predictions via self-knowledge distillation," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 13876–13885.
[37] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1452–1464, Jun. 2018.



YOUNGBUM HUR received the Ph.D. degree in operations research and industrial engineering from The University of Texas at Austin, in 2017, under the supervision of Jonathan F. Bard. He worked at Sabre, from 2017 to 2019, and the Samsung Advanced Institute of Technology (SAIT), from 2019 to 2021. He is currently an Assistant Professor with the Department of Industrial Engineering, Inha University. His early work are mostly on operations research. After he graduated, he started to work on machine learning and deep learning.

EUNHO YANG received the B.S. and M.S. degrees in computer science from Seoul National University, in 2004 and 2006, respectively, and the Ph.D. degree in computer science from The University of Texas at Austin, in 2014. From 2014 to 2016, he was a Research Staff Member with the IBM T. J. Watson Research Center. He is currently an Associate Professor with the Graduate School of Artificial Intelligence and the School of Computing, Korea Advanced Institute of Science and Technology (KAIST). His research interest includes diverse machine learning algorithms with the recent focus on the generative models.

SUNG JU HWANG received the Ph.D. degree in computer science from The University of Texas at Austin, under the supervision of Prof. Kristen Grauman. He is currently an Associate Professor with the Kim Jaechul School of Artificial Intelligence and the School of Computing, KAIST. Prior to working at KAIST, he was an Assistant Professor with the School of Electric and Computer Engineering, UNIST, and before that he was a Postdoctoral Research Associate at Disney Research, working under the supervision of Prof. Leonid Sigal.