iDECODe: In-distribution Equivariance for Conformal Out-of-distribution Detection

Ramneet Kaur[1], Susmit Jha[2], Anirban Roy[2], Sangdon Park[13], Edgar Dobriban[4], Oleg Sokolsky[1], Insup Lee[1]

[1] Department of Computer and Information Science, University of Pennsylvania, Philadelphia, USA
[2] Computer Science Laboratory, SRI International, Menlo Park, USA
[3] School of Computer Science, Georgia Institute of Technology
[4] Statistics & Computer Science, University of Pennsylvania
ramneetk@seas.upenn.edu,{susmit.jha,anirban.roy}@sri.com,
dobriban@wharton.upenn.edu, {sangdonp, sokolsky, lee}@cis.upenn.edu

Abstract

Machine learning methods such as deep neural networks (DNNs), despite their success across different domains, are known to often generate incorrect predictions with high confidence on inputs outside their training distribution. The deployment of DNNs in safety-critical domains requires detection of out-of-distribution (OOD) data so that DNNs can abstain from making predictions on those. A number of methods have been recently developed for OOD detection, but there is still room for improvement. We propose the new method iDECODe, leveraging in-distribution equivariance for conformal OOD detection. It relies on a novel base non-conformity measure and a new aggregation method, used in the inductive conformal anomaly detection framework, thereby guaranteeing a bounded false detection rate. We demonstrate the efficacy of iDECODe by experiments on image and audio datasets, obtaining state-of-the-art results. We also show that iDECODe can detect adversarial examples.

Introduction

Powerful modern machine learning methods, such as deep neural networks (DNNs) exhibit remarkable performance in domains such as computer vision (Gkioxari, Girshick, and Malik 2015), audio recognition (Hannun et al. 2014), and natural language processing (Majumder et al. 2017). However, DNNs are known to generate overconfident and incorrect predictions on inputs outside their training distribution (Hendrycks and Gimpel 2016). The responsible deployment of machine learning (ML) in safety-critical domains such as autonomous vehicles (Bojarski et al. 2016), and medicine (De Fauw et al. 2018) requires detection of out-of-distribution (OOD) data, so that these ML models can abstain from making predictions on those. A great number of methods have been developed for OOD detection, but there is still significant room for improvement. In this paper, we propose iDECODe, a novel OOD detection method based on conformal prediction with transformation equivariance learned on in-distribution (iD) data.

Equivariance of outputs to certain geometric data transforms is a general desired property of ML systems. For example, it is desirable for a classifier trained on images of upright cats to also correctly classify rotated images of cats. In other words, classifiers should learn a representation that is invariant to the orientation of the training data. Sharing of kernels in convolutional neural networks (CNNs), and more generally group CNNs, leads to learning features equivariant to translations, and more generally to group transforms. Therefore, group CNNs are mathematically guaranteed to be equivariant to translations for *all inputs*, which has played a critical role in the success of CNNs (Cohen and Welling 2016; Sabour, Frosst, and Hinton 2017). Another common approach to encode these transformations is data augmentation (Baird 1992; Krizhevsky, Sutskever, and Hinton 2012; Chen, Dobriban, and Lee 2020; Chatzipantazis et al. 2021). This is not guaranteed to lead to equivariance for all inputs, and is more likely to work for *in-distribution data* used for training than for *out-of-distribution data* dissimilar to that used for training. This is the crucial insight for us: we propose *using deviations from equivariance to test OOD-ness*.

To get rigorous control on the false detection rate, we leverage conformal prediction (Vovk, Gammerman, and Shafer 2005; Balasubramanian, Ho, and Vovk 2014), which is a general methodology to test if an input conforms to the training data. It uses a non-conformity measure (NCM) to quantitatively estimate how different an input is from the training distribution. Commonly used NCMs are based on the properties of the input's k-nearest neighbors from the training data (Balasubramanian, Ho, and Vovk 2014; Papernot and McDaniel 2018) and kernel density estimation methods (Smith et al. 2014). Inductive conformal anomaly detection (ICAD) (Laxhammar and Falkman 2015) uses an NCM to assign a non-conformity score to the input for computing its p-value indicating anomalous behavior. The performance of ICAD can depend strongly on the choice of the NCM (Balasubramanian, Ho, and Vovk 2014). We propose using the deviation (or error) in the predictable behavior of a model *equivariant in-distribution (iD) with respect to a set G of transformations* as the NCM for OOD detection.

ICAD uses a single score from the NCM to compute the p-value. We instead propose using a vector of n non-conformity scores computed from the proposed NCM with n transformations sampled as independent and identically distributed (IID) variables from a distribution over G. Intuitively, with a single transformation, an OOD datapoint might behave as a transformed iD datapoint, but the likelihood of this decreases with the number of transformations n.

The contributions of this paper are summarized as follows:

- **Novel base NCM.** We propose a novel base NCM for detecting the OOD nature of an input as the error in the iD equivariance learned by a model with respect to a set G of transforms.
- **Novel aggregation method.** We propose a novel approach to increase performance by aggregating n scores computed from the proposed base NCM on n IID transformations sampled from a distribution over G, leading to an aggregated NCM.
- **iDECODe.** Using aggregated NCM in the ICAD framework leads to our proposed iDECODe method for OOD detection with a bounded false detection rate (FDR).
- **Experiments.** We demonstrate the efficacy of iDECODe on OOD detection over image and audio datasets, obtaining state-of-the-art (SOTA) results. We also show that iDECODe can be used for adversarial example detection.

Related Work. There are a great deal of techniques for OOD detection, broadly in three categories:

Supervised: These techniques assume access to the OOD data or a proxy during the training phase of the detector. Lee et al. (2017) propose training the OOD detector with a predictive distribution following a label-dependent probability for the iD and a uniform distribution for the OOD datapoints. Similarly, Hendrycks, Mazeika, and Dietterich (2019) propose training an OOD detector based on a distinct classification or density loss for the iD and OOD datapoints. Meinke and Hein (2019) use a Bayesian framework for modeling iD and OOD datapoints separately. Lee et al. (2018) consider the Mahalanobis distance in the iD feature space to detect OOD datapoints. Logistic regression, trained on a small set of iD and OOD datapoints, is used to assign a score to the input by computing Mahalanobis distance of the noisy input from all layers of a classifier (trained for classification of the iD data). This score is expected to be higher for the iD than for OOD data. Zisselman and Tamar (2020) extend this idea by replacing the Mahalanobis distance with a likelihood function learned from residual-flow models. (Guan and Tibshirani 2019) develop methods for conformal classification and OOD detection based on learning classifiers to discriminate between the classes. Kaur et al. (2021) propose an integrated approach based on ensemble of different scores (softmax, mahalanobis etc.) for OOD detection.

Self-supervised: These techniques use a self-labeled dataset for OOD detection (Golan and El-Yaniv 2018; Bergman and Hoshen 2020; Hendrycks et al. 2019). This dataset is created by applying transformations to the iD data and labeling the transformed data with the applied transformation. A classifier is trained for the auxiliary task of predicting the applied transformation on the self-labeled dataset. The error in the classifier's prediction of the applied transformation is used as a score to detect OOD-ness of an input.

Unsupervised: These detection techniques use only the iD data for OOD detection. Hendrycks and Gimpel (2016) propose using the maximum softmax score from a classifier trained on the iD data as the baseline method (SBP) for OOD detection. These scores are expected to be higher for the iD and lower for the OOD datapoints. ODIN (Liang, Li, and Srikant 2017) was proposed as an enhancement to SBP by further separating these scores for iD and OOD datapoints after adding perturbations to the input and temperature scaling to the classifier's confidence. Recently, Macedo et al. (2021) proposed replacing softmax scores with isomax scores and entropy maximization for OOD detection. Other unsupervised detection techniques based on the difference in the density estimates (Mahmood, Oliva, and Styner 2021), energy scores (Liu et al. 2020), trust scores (Jiang et al. 2018), likelihood ratio (Ren et al. 2019), activation path (Sastry and Oore 2020) between the iD and OOD datapoints have been proposed for OOD detection.

Compared to supervised approaches, iDECODe does not require access to OOD data. It only requires the model to learn iD equivariance with respect to a set G of transforms, which is a desirable property leading to the accuracy boost of classifiers (Cohen and Welling 2016; Sabour, Frosst, and Hinton 2017). Data augmentation with G during the training of a classifier on the iD data is one way to learn G-invariant classification of the iD data (Krizhevsky, Sutskever, and Hinton 2012; Chen, Dobriban, and Lee 2020). The auxiliary task of predicting the applied transformation on a self-labeled dataset also encourages the classifier to learn G-equivariant representations of the iD data (Qi et al. 2019).

To our knowledge none of the above self-supervised and unsupervised methods provide any theoretical guarantees on OOD detection. Liu et al. (2018) provide PAC-style guarantees on the OOD detection aiming to minimize false detections. This approach is however supervised, as it requires OOD datapoints to train the detector which may not generalize to unseen OOD datapoints. Recently, there has been interest in unsupervised detection based on ICAD (Cai and Koutsoukos 2020; Bates et al. 2021). Cai and Koutsoukos (2020) propose to use Martingale test (Fedorova et al. 2012) on p-values from NCM based on either variational autoencoders (VAE) or deep support vector data description (SVDD) for OOD detection in time series data, where a batch of data is available for detection. Bates et al. (2021) focus on problems that arise in conformal detection when multiple points are tested for OOD-ness. iDECODe proposed for the detection of a single point as OOD is also built on ICAD framework, which guarantees a bounded false detection rate (FDR).

Background

Equivariance. For a set X, a function f is equivariant with respect to a set of transformations G, if there is an explicit relationship between the transformation $g \in G$ of the function input and a corresponding transformation g' of its output[1]:

$$f(g(x)) = g'(f(x)), \forall x \in X. \tag{1}$$

Invariance is a special case of equivariance, where g' is the identity function, so the output is unaffected by the transformation g of the input.

[1] Equivariant maps, also known as intertwiners, or homomorphisms between representations, are studied in representation theory, e.g., in Schur's lemma from 1905 (Hall 2015).

Inductive conformal prediction and inductive conformal anomaly detection. Conformal prediction (CP) (Vovk, Gammerman, and Shafer 2005; Balasubramanian, Ho, and Vovk 2014) aims to test if a new datapoint x_{l+1} conforms to the set $X = \{x_i : 1 \leq i \leq l\}$ of the training datapoints, quantitatively measured by a non-conformity measure (NCM). Some examples of previously proposed NCMs are based on the k-nearest neighbours algorithm (Vovk, Gammerman, and Shafer 2005; Papernot and McDaniel 2018), ridge regression (Vovk, Gammerman, and Shafer 2005), support vector machines (Vovk, Gammerman, and Shafer 2005), random forests (Devetyarov and Nouretdinov 2010), VAE and SVDD (Cai and Koutsoukos 2020).

An NCM is a real-valued function $A(X, x_{l+1})$, that assigns a non-conformity score α_{l+1} for x_{l+1} relative to datapoints in the training set X. The score α_{l+1} indicates how different x_{l+1} is relative to X. Higher α_{l+1} indicates that the new x_{l+1} is more different from the training dataset.

Conformal anomaly detection (CAD) (Laxhammar and Falkman 2011) uses this non-conformity score for the new datapoint x_{l+1} to compute its p-value p_{l+1}:

$$p_{l+1} = \frac{|\{i = 1, ..., l\} : \alpha_i \geq \alpha_{l+1}| + 1}{l + 1}.$$

Here $\{\alpha_i : i = 1, ..., l\}$ is the set of non-conformity scores for the training set computed from the new set composed of the training set and x_{l+1}. If p_{l+1} is smaller than a given anomaly threshold $\varepsilon \in (0, 1)$, then x_{l+1} is classified as a conformal anomaly (Laxhammar and Falkman 2011). CAD can be viewed as performing statistical hypothesis testing; the null hypothesis is a new datapoint x_{l+1} and training datapoints x_i for $1 \leq i \leq l$ are IID, and it is tested at the significance level ε. Under the null, the probability of rejection is at most ε (Shafer and Vovk 2008).

A drawback of the CAD framework is that it can be computationally inefficient. For every new datapoint, recalculating α_i for all the training datapoints relative to the new set containing this datapoint might become computationally infeasible if evaluating the underlying NCM A is expensive. To resolve this issue, inductive conformal anomaly detection (ICAD) (Laxhammar and Falkman 2015) was introduced based on the inductive conformal prediction (ICP) framework (Vovk, Gammerman, and Shafer 2005). Here, the training set is split into a *proper training set* $X_{\text{tr}} = \{x_i : i = 1, \ldots, m\}$ and a *calibration set* $X_{\text{cal}} = \{x_j : j = m+1, \ldots, l\}$. Further, the non-conformity scores for each datapoint $x_j \in X_{\text{cal}}$ and the new datapoint x_{l+1} are calculated relative to the proper training set:

$$\alpha_j = A(X_{\text{tr}}, x_j) \qquad \text{and} \qquad \alpha_{l+1} = A(X_{\text{tr}}, x_{l+1}).$$

ICAD uses these non-conformity scores to compute the p-value of x_{l+1}

$$p_{l+1} = \frac{|\{j = m+1, ..., l : \ \alpha_j \geq \alpha_{l+1}\}| + 1}{l - m + 1}.$$

Again, x_{l+1} is classified as conformal anomaly if $p_{l+1} < \varepsilon$.

OOD Detection with Conformal Prediction

We propose to use ICAD for OOD detection with a model trained to learn iD equivariance with respect to a set G of transformations. Here, we first define a novel NCM and nonconformity score and then formalize iDECODe's algorithm for OOD detection with a bounded false detection rate.

Novel base and aggregated NCMs

The proposed base NCM. For an input x and a transformation $g \in G$, we define a novel NCM (that we also refer to as a "base NCM") as the error in the expected behavior of the transformation-equivariance learned by a model M for the transformations G on the proper training set X_{tr}:

$$A(X_{\text{tr}}, x; g) := \text{Error}(M, x, g) = L[M(g(x)), g'M(x)]. \quad (2)$$

Here, L is a loss function, and recall that g' is an "output transform" that depends on g.

Example NCM based on data augmentation. Data augmentation with G during the training of a classifier on the iD data is used to learn invariant $(g' = I)$ classification of the iD data (Baird 1992; Krizhevsky, Sutskever, and Hinton 2012; Chen, Dobriban, and Lee 2020; Chatzipantazis et al. 2021). Non-conformance in the label prediction between the original and the transformed input can thus be used as the base NCM:

$$L[M(g(x)), g'M(x)] = \|M(g(x)) - M(x)\|_2^2. \quad (3)$$

The choice of the loss can be significant. For instance, the KL-divergence may be low both in- and out-of-distribution, for different reasons. For iD data, it can be low because the model learned the correct equivariance, while for OOD data it may be low because the predicted distribution is close to uniform for both the original and transformed OOD datapoints[2]. In our experiments, KL-divergence of the softmax scores as the base NCM lead to a relatively poor performance.

Example NCM based on auxiliary task. One can add the objective (or auxiliary task) of predicting a transformation $g \in G$ applied to the iD datapoint to "encourage" learning G-equivariant representations of the iD data (Qi et al. 2019; Golan and El-Yaniv 2018; Hendrycks et al. 2019; Bergman and Hoshen 2020). The error in the prediction of the transformation can thus be used as the base NCM:

$$L[M(g(x)), g'M(x)] = L[M(g(x)), g].$$

Here we formally set g' such that its action $g'M(x)$ on any input $M(x)$ simply equals g, which is a special type of equivariance where the output does not depend on the input. Here, $L[M(g(x)), g]$ could be $\|M(g(x)) - g\|_2^2$ if G is the set of parameterized transformations (such as affine or projective transformations) and $M(g(x))$ predicts the parameters of g. For discrete transformations, $L[M(g(x)), g]$ could be $\text{CrossEntropyLoss}(M(g(x)), g)$.

The proposed aggregated non-conformity score. Instead

[2] Some existing OOD detection techniques (Hendrycks and Gimpel 2016; Lee et al. 2017) assume that the softmax distribution for OOD datapoints is close to the uniform distribution.

of using a single transformation g as in the expression for A, we propose combining scores corresponding to several transformations. The intuition is that a single $A(X_{\text{tr}}, x, g)$ might provide only noisy information of the OOD-ness of the input. By combining information over multiple transformations, we may reduce this noise, as it is less likely for OOD datapoints to behave as iD samples under multiple transformations.

Given n transformations $g_{1:n} = (g_1, \ldots, g_n)$, we define the vector of base NCMs as

$$\mathcal{V}(x, X_{\text{tr}}; g_{1:n}) := (A(X_{\text{tr}}, x; g_1), \ldots, A(X_{\text{tr}}, x; g_n)) . \tag{4}$$

Our final non-conformity score is obtained by applying an aggregation function $F : \mathbb{R}^n \to \mathbb{R}$ to the vector $\mathcal{V}$ to aggregate the coordinate scores. We define $F(\mathcal{V}(x, X_{\text{tr}}; g_{1:n}))$ as the *aggregated NCM*. It will always be clear from the context whether we refer to the base or aggregated NCM.

The only requirement for F is that Theorem 1 showing the correctness of the resulting p-value should apply. In particular, if the distribution of the data has a density, it is sufficient if F is a coordinate-wise increasing function in the non-conformity scores. In our experiments we use $F(\mathcal{V}(x, X_{\text{tr}}; g_{1:n})) = \sum_{i=1}^{n} A(X_{\text{tr}}, x; g_i)$.

Our base NCM A and score vector $\mathcal{V}$ are defined for fixed g and $g_{1:n}$ respectively. However, to ensure that the resulting scores are valid for use in conformal inference, we need the scores to be exchangeable random variables. For this, we will randomly and independently sample all transformations from some distribution Q_G over G[3] for computing the p-value.

Algorithm and guarantee for OOD detection

A key theoretical result guarantees that the conformal inference framework applies to this construction. Compared with standard conformal prediction (Vovk, Gammerman, and Shafer 2005; Balasubramanian, Ho, and Vovk 2014), we need to make sure that the exchangeability of scores still holds in the presence of randomness in the transforms. For simplicity, in the next result, we assume that ties between $F(\mathcal{V}(x))$ and $F(\mathcal{V}(x_j))$ occur with zero probability. This holds under a broad set of practical assumptions, for instance if the distribution of the data is absolutely continuous with respect to Lebesgue measure (and thus has a density), and if F is coordinate-wise strictly increasing. If this does not hold, then we can use smoothed p-values as in (Vovk, Gammerman, and Shafer 2005). Although our experimental results are reported assuming no ties between $F(\mathcal{V}(x))$ and $F(\mathcal{V}(x_j))$, we obtained essentially the same results with smoothed p-values.

Theorem 1. *Let G be a set of transformations. For each datapoint x_j in the calibration set X_{cal}, let*

$$\mathcal{V}(x_j) = \mathcal{V}(x_j, X_{tr}; g_{j1}, \ldots, g_{jn}) \text{ as defined in (4),}$$

where for each $i = 1, \ldots, n$, g_{ji} is sampled independently from some distribution Q_G over G. Given a test datapoint x, let $\mathcal{V}(x) = \mathcal{V}(x, X_{tr}; g_{x1}, \ldots, g_{xn})$ as in (4), where for $i = 1, \ldots, n$, g_{xi} is also sampled independently from Q_G. If x is in the training distribution D, then for any $F : \mathbb{R}^n \to \mathbb{R}$, the p-value of x

$$P = \frac{|\{j = m+1, \ldots, l : F(\mathcal{V}(x_j)) \geq F(\mathcal{V}(x))\}| + 1}{l - m + 1} \tag{5}$$

is uniformly distributed over $\{1/(k+1), 2/(k+1), ..., 1\}$, where $k = l - m$.

Algorithm 1: iD equivariance for conformal OOD detection

```
Input: test datapoint $x$, model $M$ trained on proper training set, distribution $Q_G$ over transforms, loss function $L$ for equivariance, vectors of scores $\{\mathcal{V}(x_j) : j = m+1, \ldots, l\}$ over calibration set, aggregation function $F$, desired false positive rate $\varepsilon \in (0, 1)$
Output: ”1” if $x$ is detected as OOD; ”0” otherwise
$(g_1, \cdots, g_n) \sim Q_G^n$
$\mathcal{V}(x) = \{L[M(g_i(x)), g_i' M(x)] \mid 1 \leq i \leq n\}$
Set $p$-value $P = \frac{|\{j=m+1,\ldots,l : F(\mathcal{V}(x_j)) \geq F(\mathcal{V}(x))\}|+1}{l-m+1}$
Return $\mathbb{1}(P < \varepsilon)$
```

See Appendix A for a proof. Under the null hypothesis that an input datapoint x is from the training distribution (i.e., $x \sim D$), and assuming ties occur with zero probability, the p-value from equation 5 is distributed uniformly over $\{1/(k+1), \ldots, 1\}$, where $k = m - l$. We can detect if a datapoint is OOD by rejecting the null; if the p-value falls below a detection threshold $\varepsilon \in (0, 1)$, i.e., $P < \varepsilon$, then x is an OOD data point (see Algorithm 1).[4] The next result states that the false detection probability is bounded; this is a consequence of standard results on conformal inference (Balasubramanian, Ho, and Vovk 2014), and we provide the proof only for completeness in Appendix B.

Corollary 1. *The probability of false OOD detection by iDECODe is upper bounded by ε.*

Thus, if the test datapoint x and the datapoints in the training set are IID, then ε is an upper bound on the probability of detecting x as OOD (Laxhammar and Falkman 2015).

Experimental Results

Score from the proposed base NCM can be used to indicate the OOD-ness of an input. Higher score indicates more OOD-ness of an input. We use these scores for OOD detection and refer to it as the *base score* method in our experiments. We will use the shorthand ICAD for running ICAD with the proposed base NCM for OOD detection, i.e., ICAD is iDECODe with $|\mathcal{V}(x)| = 1$. For all experiments, g_i for $i = 1, \ldots, n$ is sampled independently from the uniform distribution Q_G over G. All the reported results for base score method, ICAD and iDECODe are averaged over five runs with random sampling of the set of transformations.

[3] This requires a σ-algebra on G, which is used tacitly in our work with details given in Appendix D.

[4] We remark that in statistics, it is common to use $P \leq \varepsilon$ for rejecting a null hypothesis (Lehmann and Casella 1998); however here we follow Laxhammar and Falkman (2015) to write $P < \varepsilon$.

Results on vision datasets

We first describe the model M used in the experiments. Then, we illustrate the efficacy of iDECODe with results on ablation studies and comparison with SOTA results.

Learning G-equivariant iD representations from the proper training set. Recently, Qi et al. (2019) proposed "Autoencoding Variational Transformations" (AVT) for learning transformation equivariant representations via VAE. Given a dataset X and the set G of transformations, AVT trains a VAE to predict the transformation $g \in G$ applied to an input $x \in X$. They argue that the AVT model learns an encoded representation of X that is G-equivariant by maximizing mutual information between the encoded space and G. We use an AVT model M trained to be G-equivariant on the proper training set of the iD data for results on vision datasets.

Ablation studies. Following the experimental conventions for OOD detection on vision datasets (see e.g., Lee et al. 2018; Hendrycks et al. 2019), we perform ablation studies on CIFAR-10 as the iD dataset and SVHN, LSUN, ImageNet, CIFAR100, and Places365 datasets as OOD. The details of these datasets and the AVT model M with ResNet architecture trained to be G-equivariant on the proper training set (90% of the total training data) of CIFAR-10 (with the same set of hyperparameters from Qi et al. (2019)'s AVT model on CIFAR-10) are given in Appendix C.1.1.

The set G and base NCM. Since we use Qi et al. (2019)'s AVT model trained on CIFAR-10 to perform our ablation studies, we use the same set G of projective transformations used by their AVT model to learn the equivariance on CIFAR-10. A projective transformation $g \in G$ is composed of scaling by a factor of $[0.8, 1.2]$, followed by a random rotation with an angle in $\{0°, 90°, 180°, 270°\}$, and random translations of the four corners in the horizontal and the vertical directions by a fraction of up to ± 0.125 of its height and width. A projective transformation is represented by a matrix in $\mathbb{R}^{3\times 3}$. We therefore use mean square error between the parameters of the applied and predicted transformations as the base NCM, i.e. $L[M(g(x)), g'M(x)] = \|M(g(x)) - g\|_2^2$. Here $M(g(x))$ and g are the vectors with parameter values of the predicted and applied transformation respectively.

Results. We call iD datapoints positives and OOD datapoints negatives. We use the True Negative Rate (TNR) at 90% True Positive Rate (TPR), and the Area under Receiver Operating Characteristic curve (AUROC) for evaluation. The details of these metrics are given in Appendix C. We illustrate the effectiveness of iDECODe by performing the following three ablation studies:

a) *Comparison with (1) base score method and (2) ICAD.* Table 1 shows that iDECODe with $|\mathcal{V}(x)| = 5$ outperforms both the base score method and ICAD on all OOD datasets for both TNR and AUROC. Although the performance of base score method and ICAD is quite similar, ICAD detects OOD datapoints with a bounded FDR.

b) *Performance of iDECODe versus $|\mathcal{V}(x)|$.* Figure 2 shows that AUROC increases with the size of $\mathcal{V}(x)$ ($|\mathcal{V}(x)| = 1, \ldots, 20$). Similar performance gain of iDECODe on TNR (90% TPR) with higher $|\mathcal{V}(x)|$ is reported in Appendix C.1.2. This justifies the intuition that it is unlikely for an OOD to behave as iD under multiple transforms in G for which M is iD-equivariant.

c) *Controlling the False Detection Rate.* Figure 1 shows that the FDR of iDECODe with $|\mathcal{V}(x)| = 5$ is upper bounded by ε on average. These plots are generated with $\varepsilon = 0.05 \cdot k$, $k = 1, \ldots, 10$. Calibration set of size 1000 images is randomly sampled with replacement from a held-out (not used in training) set of 5000 images. This is repeated five times, and we show box plots, with the median and inter-quartile range. Similar results on other sampled calibration set sizes are in Appendix C.1.2.

We also perform ablation studies on SVHN as the iD dataset and all other datasets as OOD. The results are similar to the results on CIFAR-10 as iD, and reported in Appendix C.1.3.

Comparison with state-of-the-art. Here, we compare the performance of iDECODe with current SOTA self-supervised and unsupervised OOD detectors. Following the convention of one-class OOD detection on CIFAR-10 with WideResNet model (WRN) (Hendrycks et al. 2019; Bergman and Hoshen 2020), we also train the AVT model with WRN architecture on the proper training set (90% of the total training set) of one class of CIFAR-10 (e.g., class "0"). All other classes (e.g., classes "1" to "9") are considered as OOD. We also evaluate iDECODe on the one-class OOD detection for the 20 superclasses of CIFAR-100. Existing results for one-class OOD detection on CIFAR-100 have been reported on ResNet-18 model (Tack et al. 2020). Therefore, we also train the AVT model with ResNet-18 architecture on the proper training set (90% of the total training set) of CIFAR-100. Model details for both CIFAR-10 and CIFAR-100 are given in Appendix C.1.4.

The set G and the base NCM. For a fair comparison with SOTA, we chose G to be the set of four classes of rotations: rotation by an integer-valued angle in the $[-10°, 10°]$, $[80°, 100°]$, $[170°, 190°]$ and $[260°, 280°]$ range. This is because the current SOTA (Hendrycks et al. 2019) uses prediction of the applied rotation angle from the set of four rotation angles ($\{0°, 90°, 180°, 270°\}$) as the auxiliary task for OOD detection. Since the prediction of the AVT model (trained to learn equivariance with respect to rotation angle ranges) is the softmax scores for the four classes of rotation angle ranges on the input, we use $L[M(g(x)), g'M(x)] = \text{CrossEntropyLoss}(M(g(x)), g)$ as the base NCM. Here g is the one-hot vector with "one" for the applied class and "zero" for the other three, and $M(g(x))$ is the vector of softmax scores of the four classes predicted by M.

Results and methods. Table 2 shows the comparison of AUROC between the existing SOTA methods, ICAD and iDECODe with $|\mathcal{V}(x)| = 5$ on the one-class OOD detection of CIFAR-10 (left) and CIFAR-100 (right). For CIFAR-100, we report the mean AUROC across all classes. Results for individual super classes of CIFAR-100 have been included in Appendix C.1.4. We achieved SOTA on five classes and the overall mean for CIFAR-10. On CIFAR-100, we achieved competitive results on most of the classes and the mean.

One-class SVM (SVM) (Schölkopf et al. 1999) and

Table 1: Comparison of base score, ICAD (iDECODe with $|\mathcal{V}(x)| = 1$) with ours (iDECODe with $|\mathcal{V}(x)| = 5$) on CIFAR-10 as iD.

D_{out}	TNR (90% TPR)			AUROC		
	Base Score	ICAD	Ours	Base Score	ICAD	Ours
SVHN	84.18 ± 0.88	84.11 ± 0.89	**93.81** ± **0.38**	92.86 ± 0.16	92.86 ± 0.16	**95.70** ± **0.07**
LSUN	55.70 ± 0.94	55.66 ± 0.92	**64.22** ± **0.68**	79.47 ± 0.26	79.47 ± 0.26	**85.98** ± **0.13**
ImageNet	60.76 ± 0.72	60.73 ± 0.70	**70.29** ± **0.61**	82.21 ± 0.18	82.21 ± 0.18	**87.97** ± **0.15**
CIFAR100	43.82 ± 0.94	43.764 ± 0.89	**48.70** ± **0.39**	72.61 ± 0.30	72.60 ± 0.30	**78.04** ± **0.20**
Places365	88.58 ± 0.34	88.58± 0.34	**99.97** ± **0.01**	96.87 ± 0.05	96.87 ± 0.06	**99.98** ± **0.01**

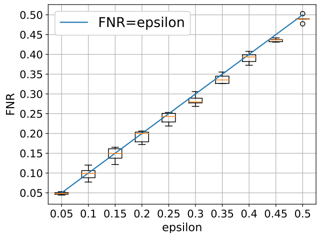


Figure 1: False detection rate of iDECODe is upper bounded by ε on average for $|X_{\text{cal}}|$ = 1000.

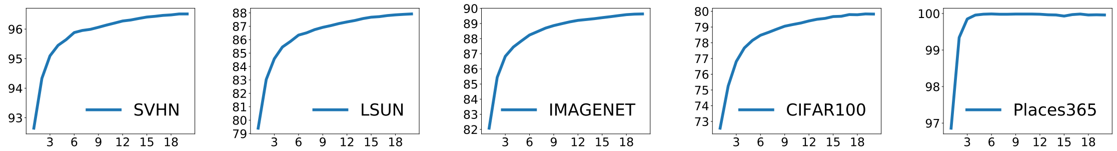


Figure 2: AUROC vs $|\mathcal{V}(x)|$ reported by iDECODe on CIFAR-10 as iD.

DeepSVDD (SVDD) (Ruff et al. 2018) are unsupervised methods, modeling the training distribution as a single class. Points outside of the iD class are detected as OOD datapoints. All other techniques are self-supervised, training a classifier with an auxiliary task of predicting the applied transformation. The error in the prediction is used for OOD detection. GM (Golan and El-Yaniv 2018) uses a set of geometric transformations. GOAD (Bergman and Hoshen 2020) generalizes GM to a class of affine transformations. RNet (Gidaris, Singh, and Komodakis 2018) uses the set of $\{0°, 90°, 180°, 270°\}$ rotations. AUX (Hendrycks et al. 2019) predicts the applied rotation, and the vertical and horizontal translations separately. The error in the three predictions is summed for OOD detection. CSI (Tack et al. 2020) uses distributionally-shifted augmentations of the iD data as negative samples in the contrastive learning framework to differentiate iD and OOD datapoints. An auxiliary task of predicting the applied shifting transform improves the features learnt by contrastive learning. Score functions on these features are used for detection.

The results of the existing techniques on CIFAR-10 are as reported in Hendrycks et al. (2019), while for GOAD they are taken from their paper (Bergman and Hoshen 2020). For CSI, results reported on CIFAR-10 in their paper are on ResNet-18 architecture, so we generate and report CSI results for one-class OOD detection of CIFAR-10 on the same WRN architecture as used for other methods. For CIFAR-100, the results of the existing techniques are as reported in Tack et al. (2020). Although we could not out-perform CSI on most of the CIFAR-100 classes, iDECODe does not require access to any OOD data during training. CSI on the other hand uses distributionally shifted augmentations of iD data as OOD data for contrastive learning of iD (or positive) and OOD (or negative) datapoints. Also, to our knowledge, CSI does not provide any detection guarantees.

Comparison with SBP as SOTA. We also compare with Hendrycks and Gimpel (2016)'s OOD detection technique (SBP) based on maximum softmax score. Since SBP depends on the softmax score of the predicted class, it cannot be applied to the one-class OOD detection problem. We therefore compare the performance of SBP with CIFAR-10 as iD and all other datasets as OOD. Table 3 shows that iDECODe (both with $|V(x)| = 5$ and$|V(x)| = 20$) could outperform SBP in most cases. The training details of the classifier used in SBP method are provided in Appendix C.1.5.

For evaluating the generalizability of iDECODe across different transformation sets, we compare the performance of iDECODe on CIFAR-10 dataset as iD and others as OOD with two transformation sets (one with the projective transformations and the other with rotations). iDECODe with both transformation sets could perform reasonably well and these results are reported in Appendix C.1.6.

Results on audio data

Dataset and iD equivariance. FSDNoisy18k (FSD) (Fonseca et al. 2019) is an audio dataset with verified (correctly labeled) and noisy data on twenty classes. We use only the verified subset of FSD. Based on similarity between classes (e.g., fire and fireworks) we group them into four sets, each containing five distinct classes. Details about these sets are included in Appendix C.2. Similar to the vision experiments, we perform one-class OOD detection on the four sets (e.g., set 0 as the iD dataset and other three sets as OOD). We train a VGG classifier on the proper training data of the iD dataset to classify the classes in the set, using data augmentation to learn iD equivariance. Fifteen datapoints from each class in the iD training set are held out as calibration data for the set (this is the same setting as Iqbal et al. (2020)'s validation set in FSD).

The set G and the base NCM. We use the set G of time and frequency masks proposed in SpecAugment (Park et al. 2019) for data augmentation in speech recognition. We use non-

Table 2: AUROC for one-class OOD detection on CIFAR-10 (left) and mean AUROC across all classes for one-class OOD detection on CIFAR-100 (right) by SOTA (SVM,. . .,CSI), ICAD($|\mathcal{V}(x)| = 1$) and Ours($|\mathcal{V}(x)| = 5$). Best results are in bold and second best are underlined.

Class	SVM	SVDD	GM	RNet	GOAD	AUX	CSI	ICAD	Ours
0	65.6	61.7	76.2	71.9	77.2	77.5	76.51 ± 0.12	85.94 ± 0.09	**86.46** ± **0.03**
1	40.9	65.9	84.8	94.5	96.7	96.9	**98.68** ± **0.02**	97.82 ± 0.09	98.11 ± 0.02
2	65.3	50.8	77.1	78.4	83.3	87.3	**88.30** ± **0.06**	86.14 ± 0.34	86.04 ± 0.46
3	50.1	59.1	73.2	70.0	77.7	80.9	79.38 ± 0.08	81.59 ± 0.23	**82.57** ± **0.11**
4	75.2	60.9	82.8	77.2	87.8	**92.7**	88.15 ± 0.11	90.27 ± 0.13	90.87 ± 0.05
5	51.2	65.7	84.8	86.6	87.8	90.2	**91.79** ± **0.07**	88.57 ± 0.21	89.24 ± 0.12
6	71.8	67.7	82.0	81.6	90.0	**90.9**	88.49 ± 0.04	88.10 ± 0.15	88.18 ± 0.40
7	51.2	67.3	88.7	93.7	96.1	96.5	97.08 ± 0.02	97.55 ± 0.09	**97.79** ± **0.06**
8	67.9	75.9	89.5	90.7	93.8	95.2	96.00 ± 0.04	96.96 ± 0.06	**97.21** ± **0.03**
9	48.5	73.1	83.4	88.8	92.0	93.3	93.31 ± 0.09	95.29 ± 0.08	**95.46** ± **0.10**
Mean	58.8	64.8	82.3	83.3	88.2	90.1	89.97	90.82	**91.19**

Method	AUROC
SVM	63.1
GM	78.7
AUX	77.7
AUX+Trans	79.8
GOAD	74.5
CSI	**89.6**
ICAD	80.31
Ours	80.66

Table 3: Comparison with Hendrycks and Gimpel (2016)'s SOTA method (SBP) on CIFAR-10. Best results are in bold.

D_{out}	TNR (90% TPR)			AUROC		
	SBP (**SOTA**)	Ours ($|(\mathcal{V}(x) = 5)|$)	Ours ($|(\mathcal{V}(x) = 20)|$)	SBP	Ours ($|(\mathcal{V}(x) = 5)|$)	Ours ($|(\mathcal{V}(x) = 20)|$)
SVHN	55.71	93.81 ± 0.38	**97.12** ± **0.13**	87.86	95.70 ± 0.07	**96.50** ± **0.01**
LSUN	66.14	64.22 ± 0.68	**67.54** ± **0.34**	**89.90**	85.98 ± 0.13	87.94 ± 0.03
ImageNet	59.13	70.29 ± 0.61	**73.67** ± **0.45**	87.32	87.97 ± 0.15	**89.58** ± **0.06**
CIFAR100	50.52	48.70 ± 0.39	**51.15** ± **0.23**	**83.13**	78.04 ± 0.20	79.92 ± 0.09
Places365	52.77	99.97 ± 0.01	**100.0** ± **0.0**	83.48	99.98 ± 0.01	**99.97** ± **0.02**

conformance between the label predictions of the original and transformed inputs from (3) as the base NCM.

Results. Table 4 shows the results for AUROC on audio OOD detection. iDECODe with $|\mathcal{V}(x)| = 20$ achieves SOTA results on all the sets.

Detection of adversarial samples

Settings. We use the same settings as in the ablation studies on vision to detect adversarial samples on the CIFAR-10 dataset. Adversarial samples are generated by using the same set (and same settings) of attacks used by Lee et al. (2018), i.e. FGSM (Goodfellow, Shlens, and Szegedy 2014), BIM (Kurakin et al. 2016), DeepFool (DF) (Moosavi-Dezfooli, Fawzi, and Frossard 2016) and CW (Carlini and Wagner 2017). These attacks are generated against two classifiers with different architectures (ResNet and DenseNet) from Lee et al. (2018)'s paper, trained on CIFAR-10. We use the same AVT model as in the ablation studies for detecting adversarial samples against both architectures.

Comparison with SOTA and results. We compare the performance of iDECODe with supervised detectors such as LID (Ma et al. 2018), Mahala (Lee et al. 2018) and a detector based on combining of kernel density estimation (Feinman et al. 2017) and predictive uncertainty (KD+PU). These detectors are trained on adversarial samples generated by FGSM attack. Unsupervised detectors such as Odds-testing (Roth, Kilcher, and Hofmann 2019) and AE-layers (Wójcik et al. 2020) are also considered. The details about these detectors are given in Appendix C.3. Table 5 shows that iDECODe with $|\mathcal{V}(x)| = 5$ achieves SOTA performance on DF and CW attacks against both architectures. Although it could not achieve SOTA results on FGSM and BIM attacks in comparison to the supervised detectors (Mahala and LID saw adversarial data generated by FGSM during its training and BIM is an iterative version of FGSM), it performs consistently well on these two attacks against both architectures. The results for supervised and unsupervised detectors are taken from Lee et al. (2018) and Wójcik et al. (2020), respectively.

Conclusion

We propose a new OOD detection method iDECODe, which leverages iD equivariance of a learned model with respect to a set of transforms. iDECODe uses a novel base NCM and a new aggregation method in ICAD framework, which guarantees a bounded FDR. We illustrate the efficacy of iDECODE on OOD detection for image and audio datasets. We also show that iDECODe performs consistently well on adversarial detection across different architectures.

Acknowledgement

The authors acknowledge support in part by ARO W911NF-17-2-0196, AFRL/DARPA FA8750-18-C-0090, and NSF #1740079. The views, opinions and/or findings expressed are those of the author(s) and should not be interpreted as representing the official views or policies of the U.S. Government.

References

Baird, H. S. 1992. Document image defect models. In *Structured Document Image Analysis*, 546–556. Springer.

Balasubramanian, V.; Ho, S.-S.; and Vovk, V. 2014. *Conformal prediction for reliable machine learning: theory, adaptations and applications*. Newnes.

Bates, S.; Candès, E.; Lei, L.; Romano, Y.; and Sesia, M. 2021. Testing for Outliers with Conformal p-values. *arXiv preprint arXiv:2104.08279*.

Table 4: AUROC for audio OOD detection by SBP as SOTA, base score method, ICAD and iDECODe.

D_{in}	SBP (**SOTA**)	Base Score	ICAD($\|\mathcal{V}(x)\| = 1$)	Ours ($\|\mathcal{V}(x)\| = 5$)	Ours ($\|\mathcal{V}(x)\| = 20$)
Set 0	59.39	61.46 ± 0.72	62.19 ± 0.26	65.07 ± 0.39	**66.21** ± **0.39**
Set 1	53.77	52.88 ± 0.86	52.48 ± 1.72	54.69 ± 0.75	**56.18** ± **0.27**
Set 2	58.04	57.97 ± 0.61	57.41 ± 0.81	60.91 ± 0.71	**61.13** ± **0.18**
Set 3	50.07	55.18 ± 0.67	54.35 ± 1.38	56.56 ± 1.10	**56.83** ± **0.34**
Mean	55.32	56.87	56.61	59.31	**60.09**

Table 5: AUROC for adversarial detection on CIFAR-10 by supervised SOTA (KD+PU, LID, Mahala), unsupervised SOTA (Odds, AE), ICAD (iDECODe with $|\mathcal{V}(x)| = 1$). and Ours (iDECODe with $|\mathcal{V}(x)| = 5$). Best results are in bold and second best are underlined.

Method	ResNet				DenseNet			
	FGSM	BIM	DF	CW	FGSM	BIM	DF	CW
KD+PU	83.51	16.16	76.80	56.30	85.96	3.10	68.34	53.21
LID	99.69	95.38	71.86	77.53	98.20	94.55	70.86	71.50
Mahala	**99.94**	**98.91**	78.06	93.90	**99.94**	**99.51**	83.42	87.95
Odds	46.32	59.85	75.58	57.58	45.23	69.01	58.30	61.29
AE	97.24	94.93	78.19	74.29	78.38	97.51	65.31	68.15
ICAD	93.1 ± 0.1	92.6 ± 0.2	92.6 ± 0.3	91.7 ± 0.1	92.3 ± 0.3	92.2 ± 0.2	90.7 ± 0.2	90.3 ± 0.2
Ours	96.6 ± 0.1	96.4 ± 0.1	**96.4** ± **0.1**	**95.9** ± **0.1**	96.2 ± 0.1	96.1 ± 0.1	**95.2** ± **0.1**	**95.0** ± **0.1**

Bergman, L.; and Hoshen, Y. 2020. Classification-based anomaly detection for general data. *arXiv preprint arXiv:2005.02359*.

Bojarski, M.; Del Testa, D.; Dworakowski, D.; Firner, B.; Flepp, B.; Goyal, P.; Jackel, L. D.; Monfort, M.; Muller, U.; Zhang, J.; et al. 2016. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*.

Cai, F.; and Koutsoukos, X. 2020. Real-time out-of-distribution detection in learning-enabled cyber-physical systems. In *2020 ACM/IEEE 11th International Conference on Cyber-Physical Systems (ICCPS)*, 174–183. IEEE.

Carlini, N.; and Wagner, D. 2017. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, 3–14.

Chatzipantazis, E.; Pertigkiozoglou, S.; Daniilidis, K.; and Dobriban, E. 2021. Learning Augmentation Distributions using Transformed Risk Minimization. *arXiv preprint arXiv:2111.08190*.

Chen, S.; Dobriban, E.; and Lee, J. H. 2020. A group-theoretic framework for data augmentation. *Journal of Machine Learning Research*, 21(245): 1–71.

Cohen, T.; and Welling, M. 2016. Group equivariant convolutional networks. In *International conference on machine learning*, 2990–2999. PMLR.

De Fauw, J.; Ledsam, J. R.; Romera-Paredes, B.; Nikolov, S.; Tomasev, N.; Blackwell, S.; Askham, H.; Glorot, X.; O'Donoghue, B.; Visentin, D.; et al. 2018. Clinically applicable deep learning for diagnosis and referral in retinal disease. *Nature medicine*, 24(9): 1342–1350.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Devetyarov, D.; and Nouretdinov, I. 2010. Prediction with confidence based on a random forest classifier. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, 37–44. Springer.

Fedorova, V.; Gammerman, A.; Nouretdinov, I.; and Vovk, V. 2012. Plug-in martingales for testing exchangeability on-line.

Feinman, R.; Curtin, R. R.; Shintre, S.; and Gardner, A. B. 2017. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*.

Fonseca, E.; Plakal, M.; Ellis, D. P.; Font, F.; Favory, X.; and Serra, X. 2019. Learning sound event classifiers from web audio with noisy labels. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 21–25. IEEE.

Gidaris, S.; Singh, P.; and Komodakis, N. 2018. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*.

Gkioxari, G.; Girshick, R.; and Malik, J. 2015. Contextual action recognition with r* cnn. In *Proceedings of the IEEE international conference on computer vision*, 1080–1088.

Golan, I.; and El-Yaniv, R. 2018. Deep anomaly detection using geometric transformations. In *Advances in Neural Information Processing Systems*, 9758–9769.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Guan, L.; and Tibshirani, R. 2019. Prediction and outlier detection in classification problems. *arXiv preprint arXiv:1905.04396*.

Hall, B. 2015. *Lie groups, Lie algebras, and representations: an elementary introduction*, volume 222. Springer.

Hannun, A.; Case, C.; Casper, J.; Catanzaro, B.; Diamos, G.; Elsen, E.; Prenger, R.; Satheesh, S.; Sengupta, S.; Coates, A.; et al. 2014. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

He, X.; Zhou, Y.; Zhou, Z.; Bai, S.; and Bai, X. 2018. Triplet-center loss for multi-view 3d object retrieval. In *Proceedings* *of the IEEE Conference on Computer Vision and Pattern Recognition*, 1945–1954.

Hendrycks, D.; and Gimpel, K. 2016. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*.

Hendrycks, D.; Mazeika, M.; and Dietterich, T. 2019. Deep anomaly detection with outlier exposure. *In International Conference on Learning Representations*.

Hendrycks, D.; Mazeika, M.; Kadavath, S.; and Song, D. 2019. Using self-supervised learning can improve model robustness and uncertainty. In *Advances in Neural Information Processing Systems*, 15663–15674.

Iqbal, T.; Cao, Y.; Kong, Q.; Plumbley, M. D.; and Wang, W. 2020. Learning with Out-of-Distribution Data for Audio Classification. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 636–640. IEEE.

Jiang, H.; Kim, B.; Guan, M.; and Gupta, M. 2018. To trust or not to trust a classifier. In *Advances in neural information processing systems*, 5541–5552.

Kaur, R.; Jha, S.; Roy, A.; Sokolsky, O.; and Lee, I. 2021. Are all outliers alike? On Understanding the Diversity of Outliers for Detecting OODs. *arXiv preprint arXiv:2103.12628*.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25: 1097–1105.

Kurakin, A.; Goodfellow, I.; Bengio, S.; et al. 2016. Adversarial examples in the physical world.

Laxhammar, R.; and Falkman, G. 2011. Sequential conformal anomaly detection in trajectories based on hausdorff distance. In *14th International Conference on Information Fusion*, 1–8. IEEE.

Laxhammar, R.; and Falkman, G. 2015. Inductive conformal anomaly detection for sequential detection of anomalous sub-trajectories. *Annals of Mathematics and Artificial Intelligence*, 74(1): 67–94.

Lee, K.; Lee, H.; Lee, K.; and Shin, J. 2017. Training confidence-calibrated classifiers for detecting out-of-distribution samples. *arXiv preprint arXiv:1711.09325*.

Lee, K.; Lee, K.; Lee, H.; and Shin, J. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *arXiv preprint arXiv:1807.03888*.

Lehmann, E.; and Casella, G. 1998. Theory of Point Estimation. *Springer Texts in Statistics*.

Lehmann, E. L.; and D'Abrera, H. J. 1975. *Nonparametrics: statistical methods based on ranks.* Holden-day.

Lehmann, E. L.; and Romano, J. P. 2006. *Testing statistical hypotheses*. Springer Science abd Business Media.

Liang, S.; Li, Y.; and Srikant, R. 2017. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*.

Liu, S.; Garrepalli, R.; Dietterich, T.; Fern, A.; and Hendrycks, D. 2018. Open category detection with PAC guarantees. In *International Conference on Machine Learning*, 3169–3178. PMLR.

Liu, W.; Wang, X.; Owens, J. D.; and Li, Y. 2020. Energy-based Out-of-distribution Detection. *arXiv preprint arXiv:2010.03759*.

Ma, X.; Li, B.; Wang, Y.; Erfani, S. M.; Wijewickrema, S.; Schoenebeck, G.; Song, D.; Houle, M. E.; and Bailey, J. 2018. Characterizing adversarial subspaces using local intrinsic dimensionality. *arXiv preprint arXiv:1801.02613*.

Macedo, D.; Ren, T. I.; Zanchettin, C.; Oliveira, A. L. I.; and Ludermir, T. 2021. Entropic Out-of-Distribution Detection. arXiv:1908.05569.

Mahmood, A.; Oliva, J.; and Styner, M. A. 2021. Multiscale Score Matching for Out-of-Distribution Detection. In *International Conference on Learning Representations*.

Majumder, N.; Poria, S.; Gelbukh, A.; and Cambria, E. 2017. Deep learning-based document modeling for personality detection from text. *IEEE Intelligent Systems*, 32(2): 74–79.

Meinke, A.; and Hein, M. 2019. Towards neural networks that provably know when they don't know. *arXiv preprint arXiv:1909.12180*.

Moosavi-Dezfooli, S.-M.; Fawzi, A.; and Frossard, P. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2574–2582.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; and Ng, A. Y. 2011. Reading digits in natural images with unsupervised feature learning.

Papernot, N.; and McDaniel, P. 2018. Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning. *arXiv preprint arXiv:1803.04765*.

Park, D. S.; Chan, W.; Zhang, Y.; Chiu, C.-C.; Zoph, B.; Cubuk, E. D.; and Le, Q. V. 2019. Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779*.

Qi, G.-J.; Zhang, L.; Chen, C. W.; and Tian, Q. 2019. Avt: Unsupervised learning of transformation equivariant representations by autoencoding variational transformations. In *Proceedings of the IEEE International Conference on Computer Vision*, 8130–8139.

Ren, J.; Liu, P. J.; Fertig, E.; Snoek, J.; Poplin, R.; Depristo, M.; Dillon, J.; and Lakshminarayanan, B. 2019. Likelihood ratios for out-of-distribution detection. In *Advances in Neural Information Processing Systems*, 14707–14718.

Roth, K.; Kilcher, Y.; and Hofmann, T. 2019. The odds are odd: A statistical test for detecting adversarial examples. In *International Conference on Machine Learning*, 5498–5507. PMLR.

Ruff, L.; Vandermeulen, R.; Goernitz, N.; Deecke, L.; Siddiqui, S. A.; Binder, A.; Müller, E.; and Kloft, M. 2018. Deep one-class classification. In *International conference on machine learning*, 4393–4402. PMLR.

Sabour, S.; Frosst, N.; and Hinton, G. E. 2017. Dynamic routing between capsules. *arXiv preprint arXiv:1710.09829*.

Sastry, C. S.; and Oore, S. 2020. Detecting out-of-distribution examples with gram matrices. In *International Conference on Machine Learning*, 8491–8501. PMLR.

Schölkopf, B.; Williamson, R. C.; Smola, A. J.; Shawe-Taylor, J.; Platt, J. C.; et al. 1999. Support vector method for novelty detection. In *NIPS*, volume 12, 582–588. Citeseer.

Shafer, G.; and Vovk, V. 2008. A Tutorial on Conformal Prediction. *Journal of Machine Learning Research*, 9(3).

Smith, J.; Nouretdinov, I.; Craddock, R.; Offer, C.; and Gammerman, A. 2014. Anomaly detection of trajectories with kernel density estimation by conformal prediction. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, 271–280. Springer.

Tack, J.; Mo, S.; Jeong, J.; and Shin, J. 2020. Csi: Novelty detection via contrastive learning on distributionally shifted instances. *Advances in Neural Information Processing Systems*, 33.

Vovk, V.; Gammerman, A.; and Shafer, G. 2005. *Algorithmic learning in a random world*. Springer Science & Business Media.

Wójcik, B.; Morawiecki, P.; Śmieja, M.; Krzyżek, T.; Spurek, P.; and Tabor, J. 2020. Adversarial Examples Detection and Analysis with Layer-wise Autoencoders. *arXiv preprint arXiv:2006.10013*.

Yu, F.; Seff, A.; Zhang, Y.; Song, S.; Funkhouser, T.; and Xiao, J. 2015. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*.

Zhou, B.; Lapedriza, A.; Xiao, J.; Torralba, A.; and Oliva, A. 2014. Learning deep features for scene recognition using places database.

Zisselman, E.; and Tamar, A. 2020. Deep residual flow for out of distribution detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13994–14003.

A. Proof of Theorem 1

Recall that data points in the proper training set and the calibration set are IID according to the probability distribution D. If $x \sim D$, then the vectors $\mathcal{V}(x), \mathcal{V}(x_{m+1}), \dots, \mathcal{V}(x_l)$, are also IID conditioned on the proper training set and the set of transforms $\{g_{xi}, 1 \le i \le n\} \cup \{g_{ji} : m+1 \le j \le l, 1 \le i \le n\}$. Indeed, the $n+1$-dimensional vectors $(x, g_{x1}, \dots, g_{xn})$, and $(x_j, g_{j1}, \dots, g_{jn})$ where $j = m+1, \dots, l$ are IID from the product distribution $D \times Q_G^n$. The vectors $\mathcal{V}$ are constructed by applying the same function $\mathcal{A}$ from Equation 4 to these vectors. Moreover $\mathcal{V}$ only depends on the proper training set and the set of transforms $\{g_{xi}, 1 \le i \le n\} \cup \{g_{ji} : m+1 \le j \le l, 1 \le i \le n\}$. Hence, conditioning on these, the vectors $\mathcal{V}(x)$ and $\mathcal{V}(x_j)$ are IID.

Similarly, the $k+1$ random variables $F(\mathcal{V}(x)), F(\mathcal{V}(x_j))$, $j = m+1, \dots, l$ (recall $k = l - m$) are IID conditioned on the proper training set and the set of transforms $\{g_{xi}, 1 \le i \le n\} \cup \{g_{ji} : m+1 \le j \le l, 1 \le i \le n\}$. Under our assumption, we thus have $k+1$ IID random variables, each with a continuous density.

Therefore, the p-value from Equation 5 is uniformly distributed in $\{1/(k+1), 2/(k+1), ..., 1\}$. This is essentially the same claim that is at the core of the validity of conformal prediction, see e.g., Proposition 2.4 in (Vovk, Gammerman, and Shafer 2005), Theorem 1.2 in (Balasubramanian, Ho, and Vovk 2014). Here we sketch the argument making a connection to order statistics. Let us denote the random variables by R_i, $i = 1, \dots, k+1$, namely $R_1 = F(\mathcal{V}(x))$ and $R_a = F(\mathcal{V}(x_{a-1+m}))$ for $a = 2, \dots k+1$. Then, the claim is that the number of indices $j \ge 1$ such that $R_1 \le R_j$ is uniformly distributed over $1, \dots, k+1$. Now, this quantity is precisely the rank of R_1 among $R_1, R_2, \dots, R_{k+1}$ minus unity. Therefore, from classical results on rank statistics, it follows that this random variable is uniformly distributed on $1, \dots, k+1$, see e.g., (Lehmann and D'Abrera 1975; Lehmann and Romano 2006).

B. Proof of Corollary 1

If the p-value for a given data point x from Equation 6 is less than $\varepsilon \in (0,1)$, then iDECODe outputs OOD as the answer. Since the p-value is distributed uniformly over $\{1/(k+1), \dots, 1\}$, the probability of this event equals

$$\sum_{1 \le j \le (k+1)\varepsilon} 1/(k+1) = \lfloor (k+1)\varepsilon \rfloor/(k+1) \le \varepsilon.$$

This error is thus upper bounded by ε.

C. Experimental details and additional results

Evaluation metrics. We call iD data points positives and OOD points negatives. We abbreivate by TP, FN, TN, and FP the notions of true positive, false negative, true negative and false positive. We use True Negative Rate (TNR) at 90% True Positive Rate (TPR), and the Area under Receiver Operating characteristic curve (AUROC) for evaluation.

The first metric is TNR = TN/(TN+FP), when TPR = TP/(TP+FN) is 90%. It indicates the percentage of OOD datapoints detected correctly when 90% of the iD samples are detected correctly. AUROC plots TPR against the false positive rate FP/(FP+TN) by varying detection thresholds. Intuitively, AUROC indicates the probability of the p-value for OOD samples being lower than the p-value of the iD samples. Higher TNR and AUROC scores indicate a good performance of the OOD detector.

C.1. Vision

C.1.1. Details about ablation studies

Datasets. The CIFAR-10 dataset (Krizhevsky, Hinton et al. 2009) consists of 60000 images for 10 classes, with 6000 images per class. The 10 classes in CIFAR-10 are airplane, car, bird, cat, deer, dog, frog, horse, ship, and trucks. There are 50000 training images (5000 per class) and 10000 test images (1000 per class). We randomly split the 50,000 training images into 45,000 proper training datapoints and 5000 calibration points. Following the experimental conventions for OOD detection, (see e.g., Lee et al. 2018; Hendrycks et al. 2019), we use SVHN (Netzer et al. 2011), LSUN (Yu et al. 2015), ImageNet (Deng et al. 2009), CIFAR100 (Krizhevsky, Hinton et al. 2009), and Places365 (Zhou et al. 2014) as the OOD datasets for CIFAR-10. Street View House Numbers (SVHN) is a dataset containing numbers taken from the Google Street View. Places365 is a scene recognition dataset. LSUN is another street recognition dataset but ten times larger in size than Places365. Imagenet is an object recognition dataset. CIFAR100 is the 100-class version of CIFAR-10. The classes in CIFAR-10 and CIFAR100 do not overlap, so CIFAR100 is an OOD dataset for CIFAR-10.

Model details. We use ResNet34 (He et al. 2016) architecture for the AVT model trained on the CIFAR-10 dataset. Following the architecture of the AVT model on CIFAR-10 from Qi et al. (2019), the encoder block consists of a convolutional layer followed by batchnorm and is inserted before the last block of ResNet34. The last block is followed by the global average pool layer. The original and transformed images x, $g(x), g \in G$, are fed into this architecture. The average-pooled features of the original and the transformed images are concatenated and fed into the two layer fully connected decoder network. This is the same decoder architecture used by Qi et al. (2019). The model is trained with the same loss function from Qi et al. (2019)'s AVT model on CIFAR-10, i.e. the mean square loss between the actual and the decoder-predicted values of the transformation matrix.[5]

C.1.2. Additional results on CIFAR-10.

Figure 3 shows that similar to AUROC, TNR (90% TPR) also increases with the size of $\mathcal{V}(x)(|\mathcal{V}(x)| = 1, ..., 20)$. Figure 4 shows more results on the bounded false detection rate with different sizes of the sampled calibration set.

C.1.3. SVHN as in-distribution for ablations study

Settings. We perform ablation studies with SVHN as iD by training the AVT model on SVHN. The model has the same architecture, hyperparameters and loss function as the model trained for ablation studies on CIFAR-10. We use the same set G and the base NCM as in the ablations studies for CIFAR-10. For SVHN, we use CIFAR-10, LSUN, ImageNet, CIFAR100 and Places365 as OOD datasets.

Results. Table 6 compares the performance of the base score method, ICAD and iDECODe with $|\mathcal{V}(x)| = 5$ (Ours). Ours outperforms both the base score method and ICAD on all OOD datasets for both TNR at 90% TPR and AUROC. The results are averaged over five runs of randomly sampled set of transforms. Figure 5 shows that the performance of iDECODe improves with the increase in $|\mathcal{V}(x)|$ for both TNR at 90% TPR and AUROC. These results are similar to the results obtained from ablation studies on CIFAR-10 as iD.

C.1.4. Model details for state-of-the-art results

Model details for CIFAR-10. We use the same WRN architecture as in the other self-supervised and unsupervised OOD detectors[6]. The architecture of the encoder and the decoder blocks are the same as in the AVT model for ablation studies. Again, the encoder block is inserted after the last block of the WRN, followed by the global average pooling layer. The original and transformed images $x, g(x), g \in G$ are fed in this architecture. The average-pooled features of the original and the transformed images are concatenated and fed into the decoder; a two-layer fully connected network. We use ReLU activation after the first layer, and the second layer is a softmax layer predicting the class of the transformation. Similar to Bergman and Hoshen (2020), we use cross-entropy loss with center-triplet loss (He et al. 2018) to train the model. Bergman and Hoshen (2020) propose that center-triplet loss (used along with cross-entropy loss to stabilize the training) yields better performance than cross-entropy loss. We also found the same in our experiments.

Model details and results on CIFAR-100. Here also, we use the same architecture of the AVT model, i.e. ResNet-18 that has been used for one-class OOD detection on CIFAR-100 by the other SOTA unsupervised and self-supervised detectors. All settings except for the model architecture (ResNet-18 instead of WRN) are same as the one-class OOD detection experiments on CIFAR-10. Table 7 shows the comparison of AUROC results for one-class OOD detection results on individual classes of CIFAR-100. We could achieve competitive results on most of the classes and the overall mean.

C.1.5. Details of Hendrycks and Gimpel (2016)'s SOTA OOD detection method (SBP) for OOD detection on CIFAR-10 as iD.

SBP uses the maximum softmax score from a classifier trained on the iD data for OOD detection. As proposed by Qi et al. (2019), we freeze the AVT encoder trained on CIFAR-10 from ablation studies and train a classifier on top of it. The architecture of the classifier is same as the decoder's architecture in the AVT model for CIFAR-10. The trained classifier achieved test accuracy of 91.46% on CIFAR-10 and was used in detection.

C.1.6. Generalizability of iDECODe with respect to the transformation set G.

Table 8 shows generalizability of iDECODe with respect to the set G of transformations. Here we compare the performance of iDECODe (on CIFAR-10 as iD) with two sets of transformations and the corresponding base NCM. The first set of transformations is the set of projective transformations (PT) from the ablation studies with mean square error as the base NCM. The second set of transformations is the set of four rotation ranges (RS) from SOTA experiments with cross-entropy loss as the base NCM.

Discussion on results. The results show that SVHN and Places365 are better detected with $G = PT$, and others (LSUN, Imagenet, and CIFAR-100) are better detected with $G = RS$. We believe that it is an important observation in the domain of self-supervised OOD detection that some OOD sets are better detected with one transformation set vs the other. A possible reason for these results could be the diversity in the OOD-ness of different datasets (ex. CIFAR-100 vs Places365) leading to the difference in detection abilities on these datasets (Kaur et al. 2021). Further investigation on this will be our future work.

[5] Our code for model training is build on top https://github.com/maple-research-lab/AVT-pytorch/tree/master/cifar.

[6] We use the wrn architecure from https://github.com/hendrycks/ss-ood/blob/master/multiclass_ood/models/wrn.py

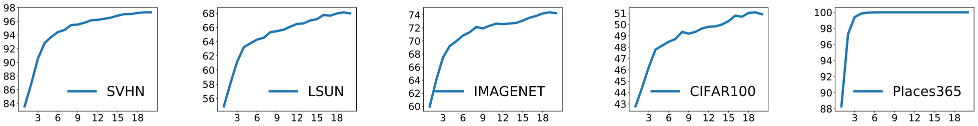
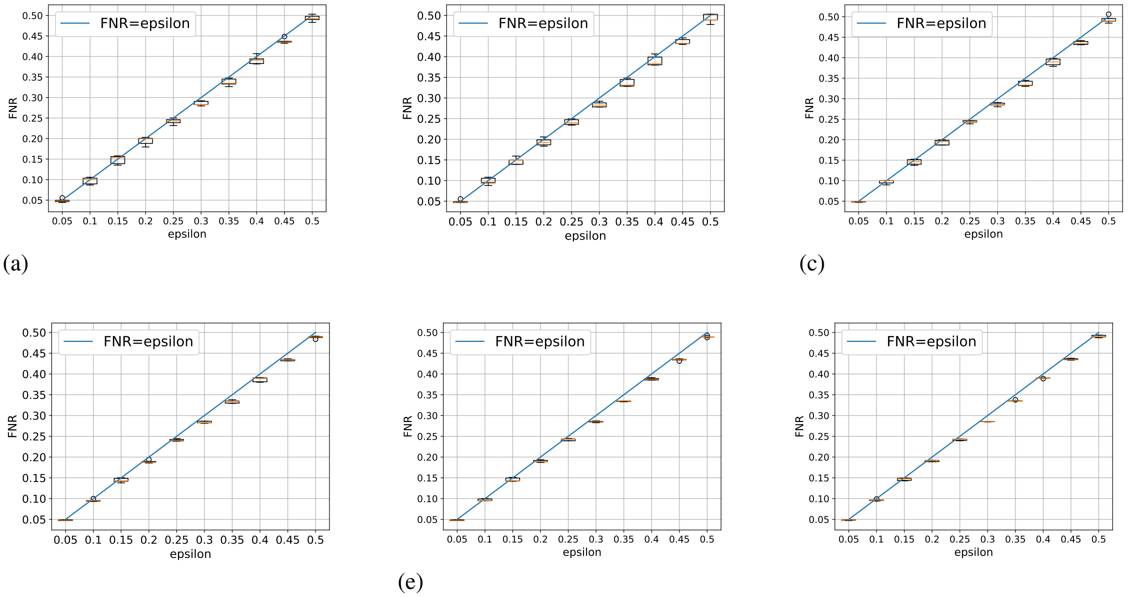


Figure 3: TNR vs $|\mathcal{V}(x)|$ reported by iDECODe on CIFAR-10 as iD.



(a)

(c)

(e)

Figure 4: False detection rate of iDECODe is upper bounded by ε on average. From left to right, $|X_{\text{cal}}| = 1500, 2000, 2500, 3000, 3500, 4000$ on CIFAR-10 as iD.

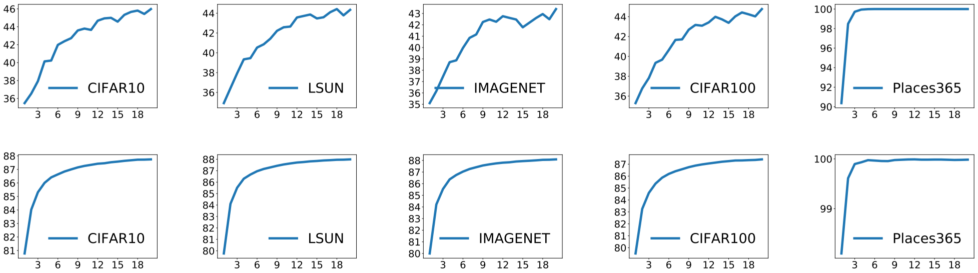


Figure 5: TNR at 90% TPR vs $|\mathcal{V}(x)|$ (top), AUROC vs $|\mathcal{V}(x)|$ (bottom) by iDECODe on SVHN as iD.

C.2. Audio

As shown in Table 9, based on the similarity among the classes, we group twenty classes of the FSDNoisy18k audio dataset into four sets. This grouping is performed based on the similarity of the classes, e.x. acoustic_guitar and bass_guitar are in the set 0 and fart, fire and fireworks are in the set 1.

We use the same VGG classifier trained with data augmentation (for iDECODe[7]) on the iD set for obtaining the SBP results.

C.3. Adversarial

Here we describe the details of SOTA methods used for comparison with iDECODe on adversarial detection.

Supervised detectors. We compare with KD+PU, LID and Mahala as the supervised adversarial detectors. Kernel Density (KD) (Feinman et al., 2017) detects datapoints lying in the

[7] Our code is build on top of Iqbal et al. (2020)'s code for training VGG classifier on the FSD audio dataset from https://github.com/tqbl/ood_audio.

Table 6: Comparison of base score method and ICAD with iDECODe on SVHN as iD.

D_{out}	TNR (90% TPR)			AUROC		
	Base Score	ICAD	Ours	Base Score	ICAD	Ours
CIFAR-10	36.01 ± 0.52	35.87 ± 0.55	**40.75± 0.38**	80.74 ± 0.13	80.74 ± 0.13	**86.35** ± **0.07**
LSUN	35.66 ± 0.62	35.51 ± 0.64	**39.99** ± **0.56**	80.08 ± 0.27	80.08± 0.27	**86.72** ± **0.09**
ImageNet	35.65 ± 0.40	35.50 ± 0.41	**39.53** ± **0.51**	80.21 ± 0.14	80.21 ± 0.14	**86.78** ± **0.11**
CIFAR100	35.88 ± 0.43	35.74 ± 0.43	**40.41** ± **0.41**	79.62± 0.13	79.62 ± 0.13	**85.90** ± **0.10**
Places365	90.42 ± 0.09	90.37 ± 0.0	**99.98** ± **0.01**	98.10 ± 0.02	98.09 ± 0.02	**99.99** ± **0.00**

Table 7: AUROC for one-class OOD detection on 20 superclasses of CIFAR-100 by SOTA, ICAD($|\mathcal{V}(x)| = 1$) and Ours($|\mathcal{V}(x)| = 5$).

Class	SVM	GM	AUX	AUX+Trans	GOAD	CSI	ICAD	Ours
0	68.4	74.7	78.6	79.6	73.9	**86.3**	80.56± 0.16	<u>81.09 ± 0.15</u>
1	63.6	68.5	73.4	73.3	69.2	**84.8**	77.28 ± 0.14	<u>77.79 ± 0.13</u>
2	52.0	74.0	70.1	71.3	67.6	**88.9**	75.72 ± 0.08	<u>75.90 ± 0.07</u>
3	64.7	<u>81.0</u>	68.6	73.9	71.8	**85.7**	65.06± 0.25	65.40 ± 0.13
4	58.2	78.4	78.7	79.7	72.7	**93.7**	81.86 ± 0.21	<u>82.16 ± 0.11</u>
5	54.9	59.1	69.7	<u>72.6</u>	67.0	**81.9**	67.79 ± 0.21	68.15 ± 0.13
6	57.2	81.8	78.8	85.1	80.0	**91.8**	84.96 ± 0.21	<u>85.47 ± 0.09</u>
7	62.9	65.0	62.5	<u>66.8</u>	59.1	**83.9**	64.35± 0.44	64.70 ± 0.15
8	65.6	85.5	84.2	<u>86.0</u>	79.5	**91.6**	84.05± 0.13	84.60 ± 0.08
9	74.1	90.6	86.3	87.3	83.7	**95.0**	94.43± 0.09	<u>94.70 ± 0.05</u>
10	84.1	87.6	87.1	88.6	84.0	**94.0**	88.80± 0.07	<u>88.92 ± 0.04</u>
11	58.0	<u>83.9</u>	76.2	77.1	68.7	**90.1**	78.98 ± 0.20	79.44 ± 0.13
12	68.5	83.2	83.3	84.6	75.1	**90.3**	84.76 ± 0.10	<u>85.26 ± 0.07</u>
13	<u>64.6</u>	58.0	60.7	62.1	56.6	**81.5**	59.91 ± 0.23	59.94 ± 0.11
14	51.2	92.1	87.1	88.0	83.8	**94.4**	92.71 ± 0.14	<u>93.08 ± 0.10</u>
15	62.8	68.3	69.0	<u>71.9</u>	66.9	**85.6**	67.57± 0.19	67.86 ± 0.17
16	66.6	73.5	71.7	<u>75.6</u>	67.5	**83.0**	72.75± 0.31	73.19± 0.15
17	73.7	93.8	92.2	93.5	91.6	97.5	<u>98.10 ± 0.09</u>	**98.28** ± **0.03**
18	52.8	90.7	90.4	91.5	88.0	**95.9**	93.28 ± 0.11	<u>93.66 ± 0.08</u>
19	58.4	85.0	86.5	88.1	82.6	**95.2**	93.36 ± 0.10	<u>93.67± 0.04</u>
Mean	63.1	78.7	77.7	79.8	74.5	**89.6**	80.31	<u>80.66</u>

Table 8: Comparing AUROC of iDECODe for G = set of projective transformations (PT) and G = set of rotations (RT).

D_{out}	$\|\mathcal{V}(x)\| = 1$		$\|\mathcal{V}(x)\| = 5$	
	$G = PT$	$G = RS$	$G = PT$	$G = RS$
SVHN	92.86 ± 0.16	92.09 ± 0.02	95.70 ± 0.07	92.37 ± 0.02
LSUN	79.47± 0.26	86.65 ± 0.06	85.98 ± 0.13	87.52 ± 0.05
ImageNet	82.21 ± 0.18	88.59 ± 0.05	87.97 ± 0.15	89.43 ± 0.05
CIFAR100	72.60 ± 0.30	82.46 ± 0.04	78.04 ± 0.20	83.08 ± 0.04
Places365	96.87 ± 0.06	86.73 ± 0.04	99.98 ± 0.01	88.15 ± 0.03

low Guassian density regions as adversarial. Specifically, KD uses the following score to detect an input x as adversarial:

$$KD(x) = \frac{1}{|X_t|} \sum_{x_i \in X_t} k(x_i, x).$$

Here t is the predicted class label of x, X_t is the set of training samples with the label t, and $k_\sigma(x, y) \sim \exp(-||x - y||^2/\sigma^2)$. Predictive Uncertainty (PU) is the maximum value of the softmax score predicted by the classifier. KD+PU is a logistic regression detector trained on the scores from KD and PU to detect adversarial inputs. Local Intrinsic Dimensionality (LID) (Ma et al. 2018) uses distance distribution of the input's neighboring samples from the training data to detect adversaries. Specifically, LID computes the following score for an input x to detect it as an adversarial input:

$$\widehat{LID}(x) = -\left(\frac{1}{k}\sum_{i=1}^{k} \log \frac{r_i(x)}{r_k(x)}\right)^{-1}.$$

Here $r_i(x)$ is the distance between x and i^{th} training sample from the set of k nearest neighbors and $r_k(x)$ is the distance from farthest sample in the set. Logistic regression detector trained on LID scores from all the layers is used to detect adversarial inputs. Mahala (Lee et al. 2018) computes Mahalanobis distance of the noisy input in the feature space of the training data as a score to detect adversarial inputs. Logistic regression detector trained on mahalanobis distance from all the layers is used to detect adversarial inputs.

Table 9: Grouping of FSDNoisy18k classes into four sets.

D_{in}	FSDNoisy18k classes
Set 0	Acoustic_guitar, Bass_guitar, Clapping, Coin_(dropping), Crash_cymbal
Set 1	Dishes_and_pots_and_pans, Engine, Fart, Fire, Fireworks
Set 2	Glass, Hi-hat, Piano, Rain, Slam
Set 3	Squeak, Tearing, Walk_or_footsteps, Wind, Writing

Unsupervised detectors. We use Odds and AE as the unsupervised adversarial detectors. Odds (Roth, Kilcher, and Hofmann 2019) propose statistical testing for adversarial detection based on the insight that feature variation caused by adding noise to the input tends to have a characteristic direction for adversarial samples. Specifically, they measure the change in pair-wise class logits caused by adding noise to the input to detect adversaries. AE (Wójcik et al. 2020) makes use of autoencoders trained on the feature space from every layer of the classifier to detect adversarial inputs. Specifically, they use reconstruction error and latent norm from the trained autoencoders as scores for adversarial detection.

C.4. Computational resources

We run all our experiments on a server with CUDA version 11.1 and four GPU cards. The specifications of all the four GPU cards are same and are as follows:

1. Card: NVIDIA Corporation [GeForce RTX 2080 Ti Rev. A].
2. Memory: 11019MiB.

D. Details on measurability

Here we provide some standard details on the measurability issues from our paper. We assume we have a probability space (Ω, B, P), where Ω is a base space, B is a sigma-algebra on Ω, and $P : B \to [0, 1]$ is a probability measure on B. The set G of transformations is a subset of all maps

$\mathcal{X} \to \mathcal{X}$, and is also equipped with a sigma-algebra B_G of measurable subsets of G. We assume that $\mathcal{X}$, the feature space, is equipped with a sigma-algebra B_X, and that each $g \in G$ is measurable with respect to B_X.

Choosing a random $g \in G$ is accomplished by a measurable function $g : X \times \Omega \to G$ (measurable with respect to the product sigma-algebra $B_X \times B$ over the input and B_G over the output), which assigns to each $\omega \in \Omega$ a specific $g = g(\cdot; \omega) \in G$. The distribution of $g(\cdot, \omega)$ when $\omega \sim P$ is denoted as $Q_G : B_G \to [0, 1]$, and is a probability measure on (G, B_G). Then, we can construct the product measure $P^{(n+1),(k+1)}$, and thus also the product measure $Q_G^{(n+1),(k+1)}$, allowing us to sample iid $g_{ji}, j \in \{m+1, \ldots, l, x\}, i = 1, \ldots, n$. Moreover, we assume that the map $g \to g'$ mapping the input transform to the output transform is measurable over the sigma algebra B_G for the input and output.

We assume that $M : \mathcal{X} \to \mathcal{X}$ is measurable with respect to B_X, and that the loss function $L : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is measurable as a function between the sigma-algebra $B_X \times B_X$ on $\mathcal{X} \times \mathcal{X}$ and the Borel-sigma algebra on $\mathbb{R}$.

Finally, we also assume that $A(X_{\mathrm{tr}}, \cdot, \cdot) : \mathcal{X} \times G \to \mathbb{R}$ is measurable with respect to the sigma-algebra $B_X \times B_G$, and that the aggregation function F is measurable with respect to the Borel sigma-algebras for its input and the output.