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

Ramneet Kaur, Susmit Jha, Anirban Roy, Sangdon Park, Edgar Dobriban, Oleg Sokolsky, Insup Lee

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
ramneet@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 require 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. Code, pre-trained models, and data are available at https://github.com/ramneet/iDECODe.

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 transformations 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 transformations.
- **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 the classifier. This score is expected to be higher for the iD than for OOD data. Guan and Tibshirani (2019) develop methods for conformal classification and OOD detection based on learning classifiers to discriminate between the classes. Kaur et al. (2021a) propose an 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 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 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 detection.

Compared to supervised approaches, iDECODe does not require access to any OOD data. It only requires the model to learn iD equivariance with respect to a set \( G \) of transformations, 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 detection aiming to minimize false detections. This approach is however supervised, as it requires OOD datapoints for training 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 on variational autoencoders (VAE) or deep support vector data description (SVDD) for 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:

\[
f(g(x)) = g'(f(x)), \forall x \in X.
\] (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.

**Inductive Conformal Prediction and Inductive Conformal Anomaly Detection.** Conformal prediction (CP) (Vovk,
We propose to use ICAD for OOD detection with a model calculating a non-conformity score and then formalize iDECODe’s algorithm for OOD detection with a bounded false detection rate. ICAD uses these non-conformity scores to compute theanthology conformal anomaly if $\alpha_j = A(X_{tr}, x_j)$ and $\alpha_{l+1} = A(X_{tr}, x_{l+1})$. 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 non-conformity 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 $X_{tr}$:

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

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 input and the transformed input can thus be used as the base NCM:

$$L[M(g(x)), g'M(x)] = ||M(g(x)) - M(x)||_2^2.$$ The choice of the loss can be significant. For instance, the KL-divergence may be low both iD 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 (Lee et al. 2017). 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 of-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 CrossEntropyLoss$(M(g(x)), g)$.

The **Proposed Aggregated Non-Conformity Score.** Instead of using a single transformation $g$ in the expression for $A$, we propose combining scores corresponding to several transformations. The intuition is that a single $A(X_{tr}, x; g)$ 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

$$\forall (x, X_{tr}; g_1:n) := (A(X_{tr}, x; g_1), \ldots, A(X_{tr}, x; g_n)).$$
Our final non-conformity score is obtained by applying an aggregation function \( F: \mathbb{R}^n \to \mathbb{R} \) to the vector \( V \) to aggregate the coordinate scores. We define \( F(V(x, X_n; g_{1:n})) \) 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 \( F(V(x, X_n; g_{1:n})) = \sum_{i=1}^n A(X_n, x; g_i) \).

Our base NCM \( A \) and score vector \( V \) are defined for fixed \( g \) and \( g_{1:n} \). 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 \( G \) 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 transformations. For simplicity, in the next result, we assume that ties between \( F(V(x)) \) and \( F(V(x_j)) \) 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 \( F(V(x)) \) and \( F(V(x_j)) \), 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 \( X_{ca} \), let \( V(x_j) = V(x_j, X_n; g_{1}, \ldots, g_{m}) \) 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 \( V(x) = V(x, X_n; g_{x1}, \ldots, g_{xn}) \) 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 \( x \)

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

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

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). 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 \( \varepsilon \).

Thus, if the test datapoint \( x \) and the datapoints in the training set are IID, then \( \varepsilon \) 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 \( |V(x)| = 1 \). 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.

**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 \( q \in G \) 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.
We also perform ablation studies on SVHN as the iD dataset and CIFAR-100, LSUN, ImageNet, CIFAR-10, and Places365 datasets as OOD. The details about 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 the 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 \( \{0^\circ, 90^\circ, 180^\circ, 270^\circ\} \), and random translations of the four corners in the horizontal and the vertical directions by a fraction of up to \( \pm 0.125 \) 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. 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 \( |V(x)| = 5 \) outperforms both the base score method and ICAD on all OOD datasets for both TNR and AUCR. 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 \( |V(x)| \).** Figure 1 (left) shows that AUROC increases with the size of \( |V(x)| \) \( (|V(x)| = 1, \ldots, 20) \). We observed similar performance gain of iDECODe on TNR (90% TPR) with higher \( |V(x)| \). This justifies the intuition that it is unlikely for an OOD to behave as iD under multiple transformations in \( G \) for which \( M \) is iD-equivariant.

c) **Controlling the False Detection Rate.** Figure 1 (right) shows that the FDR of iDECODe with \( |V(x)| = 5 \) is upper bounded by \( \varepsilon \) 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. We obtained similar results on other sampled calibration set sizes of 2000, 3000, and 4000.

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.1.2.

**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 super-classes of CIFAR-10. 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.

**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 \([-10^\circ, 10^\circ], [80^\circ, 100^\circ], [170^\circ, 190^\circ], \text{and } [260^\circ, 280^\circ]\) 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 \( \{0^\circ, 90^\circ, 180^\circ, 270^\circ\} \) 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 range on the input, we use \( L[M(g(x)), g(M(x))] = \text{CrossEntropyLoss}(\hat{M}(g(x)), g) \) 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 \( |V(x)| = 5 \) 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.1.3. 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 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 \( \{0^\circ, 90^\circ, 180^\circ, 270^\circ\} \) 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
<table>
<thead>
<tr>
<th>$D_{out}$</th>
<th>TNR (90% TPR)</th>
<th>AUROC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Base Score</td>
<td>ICAD</td>
</tr>
<tr>
<td>SVHN</td>
<td>84.18 ± 0.88</td>
<td>84.11 ± 0.89</td>
</tr>
<tr>
<td>LSUN</td>
<td>55.70 ± 0.94</td>
<td>55.66 ± 0.92</td>
</tr>
<tr>
<td>ImageNet</td>
<td>60.76 ± 0.72</td>
<td>60.73 ± 0.70</td>
</tr>
<tr>
<td>CIFAR100</td>
<td>43.82 ± 0.94</td>
<td>43.764 ± 0.89</td>
</tr>
<tr>
<td>Places365</td>
<td>88.58 ± 0.34</td>
<td>88.58 ± 0.34</td>
</tr>
</tbody>
</table>

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

![Figure 1: AUROC vs $|\mathcal{V}(x)|$ reported by iDECODe on CIFAR-10 as iD (left). False detection rate of iDECODe is upper bounded by $\varepsilon$ on average for $|X_{cal}| = 1000$ (right).](image)

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 $|\mathcal{V}(x)| = 5$ and $|\mathcal{V}(x)| = 20$) could outperform SBP in most cases. The training details of the classifier used in SBP method are provided in Appendix C.1.4.

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.1.5.

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, the Base NCM, and Results. 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-conformance between the label predictions of the original and transformed inputs from (3) as the base NCM. Table 4 shows the results for AUROC on audio OOD detection. iDECODe with $|\mathcal{V}(x)| = 20$ achieves SOTA results on all the sets. SBP results were obtained by using the same VGG classifier trained with data augmentation (as iDECODe) on the iD set.

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 settings of attacks used by Lee et al. (2018), i.e. FGSM (Goodfellow, Shlens, and Szegedy 2014), BIM (Kurakin, Goodfellow, and Bengio 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 inputs for both architectures.

Comparison with SOTA and Results. We compare the per-
Table 2: AUROC for one-class detection on CIFAR-10 (left) and mean AUROC on all classes for one-class detection on CIFAR-100 (right) by SOTA(SVM,..,CSI), ICAD (iDECODe with $|V(x)| = 1$) and Ours (iDECODe with $|V(x)| = 5$). Best results are in bold and second best are underlined.

<table>
<thead>
<tr>
<th>Class</th>
<th>SVM</th>
<th>SVDD</th>
<th>GM</th>
<th>RNet</th>
<th>GOAD</th>
<th>AUX</th>
<th>CSI</th>
<th>ICAD</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>65.6</td>
<td>61.7</td>
<td>76.2</td>
<td>71.9</td>
<td>77.2</td>
<td>77.5</td>
<td>76.51 ± 0.12</td>
<td>85.94 ± 0.09</td>
<td>86.46 ± 0.03</td>
</tr>
<tr>
<td>1</td>
<td>40.9</td>
<td>65.9</td>
<td>84.8</td>
<td>94.5</td>
<td>96.7</td>
<td>96.9</td>
<td>98.68 ± 0.02</td>
<td>97.82 ± 0.09</td>
<td>98.11 ± 0.02</td>
</tr>
<tr>
<td>2</td>
<td>65.3</td>
<td>50.8</td>
<td>77.1</td>
<td>78.4</td>
<td>83.3</td>
<td>87.3</td>
<td>88.30 ± 0.06</td>
<td>86.14 ± 0.34</td>
<td>86.04 ± 0.46</td>
</tr>
<tr>
<td>3</td>
<td>50.1</td>
<td>59.1</td>
<td>73.2</td>
<td>70.0</td>
<td>77.7</td>
<td>80.9</td>
<td>79.38 ± 0.08</td>
<td>81.59 ± 0.23</td>
<td>82.57 ± 0.11</td>
</tr>
<tr>
<td>4</td>
<td>75.2</td>
<td>60.9</td>
<td>82.8</td>
<td>77.2</td>
<td>87.8</td>
<td>92.7</td>
<td>88.15 ± 0.11</td>
<td>90.27 ± 0.13</td>
<td>90.87 ± 0.05</td>
</tr>
<tr>
<td>5</td>
<td>51.2</td>
<td>65.7</td>
<td>84.8</td>
<td>86.6</td>
<td>87.8</td>
<td>90.2</td>
<td>91.79 ± 0.07</td>
<td>88.57 ± 0.21</td>
<td>89.74 ± 0.12</td>
</tr>
<tr>
<td>6</td>
<td>71.8</td>
<td>67.7</td>
<td>82.0</td>
<td>81.6</td>
<td>90.0</td>
<td>90.9</td>
<td>88.49 ± 0.04</td>
<td>88.10 ± 0.15</td>
<td>88.18 ± 0.40</td>
</tr>
<tr>
<td>7</td>
<td>51.2</td>
<td>67.3</td>
<td>88.7</td>
<td>93.7</td>
<td>96.1</td>
<td>96.5</td>
<td>97.08 ± 0.02</td>
<td>97.55 ± 0.09</td>
<td>97.79 ± 0.06</td>
</tr>
<tr>
<td>8</td>
<td>67.9</td>
<td>75.9</td>
<td>89.5</td>
<td>90.7</td>
<td>93.8</td>
<td>95.2</td>
<td>96.00 ± 0.04</td>
<td>96.96 ± 0.06</td>
<td>97.21 ± 0.03</td>
</tr>
<tr>
<td>9</td>
<td>48.5</td>
<td>73.1</td>
<td>83.4</td>
<td>88.8</td>
<td>92.0</td>
<td>93.3</td>
<td>93.31 ± 0.09</td>
<td>95.29 ± 0.08</td>
<td>95.46 ± 0.10</td>
</tr>
</tbody>
</table>

Mean 58.8 | 64.8 | 82.3 | 83.3 | 88.2 | 90.1 | 89.97 | 90.82 | 91.19

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

<table>
<thead>
<tr>
<th>$D_{out}$</th>
<th>SVHN</th>
<th>LSUN</th>
<th>ImageNet</th>
<th>CIFAR100</th>
<th>Places365</th>
</tr>
</thead>
<tbody>
<tr>
<td>SBP (SOTA)</td>
<td>55.71</td>
<td>66.14</td>
<td>59.13</td>
<td>50.52</td>
<td>57.72</td>
</tr>
<tr>
<td>Ours (SBP (SOTA) $</td>
<td>V(x)</td>
<td>= 5$)</td>
<td>93.81 ± 0.38</td>
<td>64.22 ± 0.68</td>
<td>70.29 ± 0.61</td>
</tr>
<tr>
<td>Ours (SBP (SOTA) $</td>
<td>V(x)</td>
<td>= 20$)</td>
<td>97.12 ± 0.13</td>
<td>67.54 ± 0.34</td>
<td>73.67 ± 0.45</td>
</tr>
<tr>
<td>AUROC</td>
<td>87.86</td>
<td>89.90</td>
<td>87.32</td>
<td>83.13</td>
<td>83.48</td>
</tr>
<tr>
<td>SBP</td>
<td>95.70 ± 0.07</td>
<td>85.98 ± 0.13</td>
<td>87.97 ± 0.15</td>
<td>78.04 ± 0.20</td>
<td>99.98 ± 0.01</td>
</tr>
<tr>
<td>Ours (SBP (SOTA) $</td>
<td>V(x)</td>
<td>= 5$)</td>
<td>96.50 ± 0.01</td>
<td>87.94 ± 0.03</td>
<td>89.58 ± 0.06</td>
</tr>
<tr>
<td>Ours (SBP (SOTA) $</td>
<td>V(x)</td>
<td>= 20$)</td>
<td>96.21 ± 0.39</td>
<td>66.19 ± 0.26</td>
<td>52.48 ± 1.72</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Method</th>
<th>ResNet</th>
<th>DenseNet</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FGSM</td>
<td>BIM</td>
</tr>
<tr>
<td>KD+PU</td>
<td>83.51</td>
<td>16.36</td>
</tr>
<tr>
<td>LID</td>
<td>99.69</td>
<td>95.38</td>
</tr>
<tr>
<td>Mahala</td>
<td>99.94</td>
<td>98.91</td>
</tr>
<tr>
<td>Odds</td>
<td>46.32</td>
<td>59.85</td>
</tr>
<tr>
<td>AE</td>
<td>97.24</td>
<td>94.93</td>
</tr>
<tr>
<td>ICAD</td>
<td>93.1 ± 0.1</td>
<td>92.6 ± 0.2</td>
</tr>
<tr>
<td>Ours</td>
<td>96.6 ± 0.1</td>
<td>96.4 ± 0.1</td>
</tr>
</tbody>
</table>

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

The performance of iDECODe with supervised detectors such as LID (Ma et al. 2018), Mahala (Lee et al. 2018), 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. Table 5 shows that iDECODe with $|V(x)| = 5$ 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 transformations. 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. Additional details on measurability, and experiments are included in the arxiv version of this paper: https://arxiv.org/pdf/2201.02331.pdf.

A. Proof of Theorem 1
Recall that datapoints in the proper training and calibration sets are IID according to the probability distribution $D$. If $x \sim D$, then the vectors $\mathcal{V}(x), \mathcal{V}(x_{m+1}), \ldots, \mathcal{V}(x_1)$, are also IID conditioned on the proper training set and the set of transformations $\{g_{x_1}, 1 \leq i \leq n\} \cup \{g_{x_2}, m+1 \leq j \leq l, 1 \leq i \leq n\}$. Indeed, the $n+1$-dimensional vectors $(x,g_{x_1}, \ldots, g_{x_n})$, and $(x_j, g_{x_2}, \ldots, g_{x_n})$ where $j = m+1, \ldots, l$ are IID from the product distribution $D \times Q^G$. The vectors $\mathcal{V}$ are constructed by applying the same function $A$ from Equation 4 to these vectors. Moreover $\mathcal{V}$ only depends on the proper training set and the set of transformations $\{g_{x_1}, 1 \leq i \leq n\} \cup \{g_{x_2}, m+1 \leq j \leq l, 1 \leq i \leq n\}$. Hence, conditioning on these, the vectors $\mathcal{V}(x)$ and $\mathcal{V}(x_j)$ are IID.

Similarly, the $k+1$ random variables $F(\mathcal{V}(x)), F(\mathcal{V}(x_j)), j = m+1, \ldots, l$ (recall $k = l-m$) are IID conditioned on the proper training set and the set of transformations $\{g_{x_1}, 1 \leq i \leq n\} \cup \{g_{x_2}, m+1 \leq j \leq l, 1 \leq i \leq n\}$. 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 $\{1/(k+1), 2/(k+1), \ldots, 1\}$. 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, Man, and Vovk 2014). Here we sketch the argument making a connection to order statistics. Let us denote the random variables by $R_k, i = 1, \ldots, k+1$, namely $R_k = F(\mathcal{V}(x))$ and $R_0 = F(\mathcal{V}(x_{a+1}))$ for $a = 2, \ldots, k+1$. Then, the claim is that the number of indices $j \geq 1$ such that $R_1 \leq R_j$ is uniformly distributed over $1, \ldots, k+1$. Now, this quantity is precisely the rank of $R_1$ among $R_1, R_2, \ldots, R_{k+1}$ minus unity. Therefore, from classical results on rank statistics, it follows that this random variable is uniformly distributed on $1, \ldots, k+1$, 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 5 is less than $\varepsilon \in (0, 1)$, then iDECODe outputs OOD as the answer. Since the p-value is distributed uniformly over $\{1/(k+1), \ldots, 1\}$, the probability of this event equals

$$\sum_{1 \leq j \leq (k+1)\varepsilon} 1/(k+1) = [(k+1)\varepsilon]/(k+1) \leq \varepsilon.$$

This error is thus upper bounded by $\varepsilon$.

C. Experimental Details and Additional Results

C.1 Vision

C.1.1 Details about Ablation Studies

Model Details. We use ResNet34 (He et al. 2016) architecture as 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 layers 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.1

C.1.2 SVHN as iD for Ablation Studies. We perform ablation studies with SVHN as iD by training an AVT model on SVHN. The model is trained with the same set $G$, base NCM, network architecture, hyperparameters and loss function as the model trained for ablation studies on CIFAR-10. For SVHN, we use CIFAR-10, LSUN, ImageNet, CIFAR100, and Places365 as the OOD datasets. 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. Consistent with the observation on CIFAR-10, the performance of iDECODe on SVHN improves with the increase in $|\mathcal{V}(x)|$ for both TNR and AUROC.

C.1.3 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 detectors2. 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, 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

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

2 We use the wrn architecture from https://github.com/hendrycks/ss-ood/blob/master/multiclass_ood/models/wrn.py.
As proposed by Qi et al. (2019), we freeze the AVT encoder for individual classes of CIFAR-100. We could achieve competitive results on most of the classes and the overall mean.

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.1.4 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.

As shown in Table 9, based on the similarity among the classes, we group twenty classes of the FSDNoisy18k audio dataset into four sets.

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

<table>
<thead>
<tr>
<th>Dataset</th>
<th>AUROC G = PT</th>
<th>AUROC G = RS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVHN</td>
<td>92.86 ± 0.16</td>
<td>92.09 ± 0.02</td>
</tr>
<tr>
<td>LSUN</td>
<td>79.47 ± 0.26</td>
<td>86.65 ± 0.06</td>
</tr>
<tr>
<td>ImageNet</td>
<td>82.21 ± 0.18</td>
<td>88.59 ± 0.05</td>
</tr>
<tr>
<td>CIFAR100</td>
<td>72.60 ± 0.30</td>
<td>82.46 ± 0.04</td>
</tr>
<tr>
<td>Places365</td>
<td>96.87 ± 0.06</td>
<td>86.73 ± 0.04</td>
</tr>
</tbody>
</table>

**Table 9:** Grouping of FSDNoisy18k classes into four sets.

<table>
<thead>
<tr>
<th>SDNoisy18k classes</th>
<th>Set 0</th>
<th>Set 1</th>
<th>Set 2</th>
<th>Set 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Acoustic_guitar, Bass_guitar, Clarinet, Crash cymbal</td>
<td>Dishes_pots_pans, Engine, Fart, Fire, Fireworks</td>
<td>Glass, Hi-hat, Piano, Rain, Slam</td>
<td>Squeak, Tearing, Walk_or_footsteps, Wind, Writing</td>
</tr>
</tbody>
</table>

The trained classifier achieved test accuracy of 91.46% on CIFAR-10 and was used in detection.

**C.1.5 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. 2021b).

**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.

**Acknowledgements**

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

---

3Our training is build on top of the training code https://github.com/tqbl/ood_audio.
References


