A FUNCTIONAL PERSPECTIVE ON MULTI-LAYER OUT-OF-DISTRIBUTION DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

A crucial component for implementing reliable Machine Learning classifiers detecting examples far from the reference (training) distribution, referred to as out-of-distribution (OOD) samples. A key feature of OOD sample detection is the use of the network to extract statistical patterns and relationships through the pre-trained multi-layer classifier. Despite achieving solid results, state-of-the-art methods require either additional OOD examples, expensive computation of gradients, or are tightened to a particular architecture, limiting their applications. This paper overcomes these issues by adopting an original approach based on a functional view of the network that exploits the sample's trajectories through the various layers and their statistical dependencies. In this new framework, OOD detection translates into detecting samples whose trajectories differ from the typical behavior characterized by the training set. Our method significantly decreases the OOD detection error of classifiers trained on ImageNet and outperforms the state-of-the-art methods on average AUROC and TNR at 95% TPR. This validates our approach and empirically demonstrates that the functional signature left by a sample in a network carries relevant information for OOD detection. We hope this work will pave a new way for future research to enhance the safety of AI systems.

1 INTRODUCTION

The current ability of a Deep Neural Network (DNN) to generalize to new data is mostly restricted to priorly known concepts in the training dataset. In real-world scenarios, Machine Learning (ML) models may encounter Out-Of-Distribution (OOD) samples, such as data belonging to novel concepts (classes) Pimentel et al. (2014), abnormal samples Tishby & Zaslavsky (2015), or even carefully crafted attacks designed to exploit the model Szegedy et al. (2013). The behavior of ML systems on unseen data is of great concern for safety-critical applications Amodei et al. (2016b;a), such as medical diagnosis in healthcare Subbaswamy & Saria (2020), autonomous vehicle control in transportation Bojarski et al. (2016) among others. To address safety issues arising from the presence of OOD samples, a successful line of work aims at augmenting ML models with an OOD binary detector to distinguish between abnormal and in-distribution examples Hendrycks & Gimpel (2017). An analogy to the detector is the human body's immune system, with the task of differentiating between antigens and the body itself.

Distinguishing OOD samples is challenging and previous works developed detectors by combining scores at the various layers of the multi-layer pre-trained classifier Sastry & Oore (2020); Lee et al. (2018); Gomes et al. (2022); Huang et al. (2021). These detectors require either a held-out OOD dataset (e.g., adversarially generated or OOD data) or ad-hoc methods to linearly combine OOD scores computed on each layer embedding tightened to a particular architecture. A key observation we leverage in our work is that existing aggregation techniques overlook the sequential nature of the underlying problem and have limited discriminating power: an input sample passes consecutively through each layer and generate a highly correlated signature that can be statistically characterized.

In this paper, we introduce a significant change of perspective. Instead of looking at each score independently, we cast them into a sequential representation that captures the statistical trajectory of an input sample through the various layers of a multi-layer neural network. To this end, we adopt a functional point of view by considering the sequential representation as curves parametrized by each layer. Consequently, we redefine the problem of OOD detection as that of detecting samples

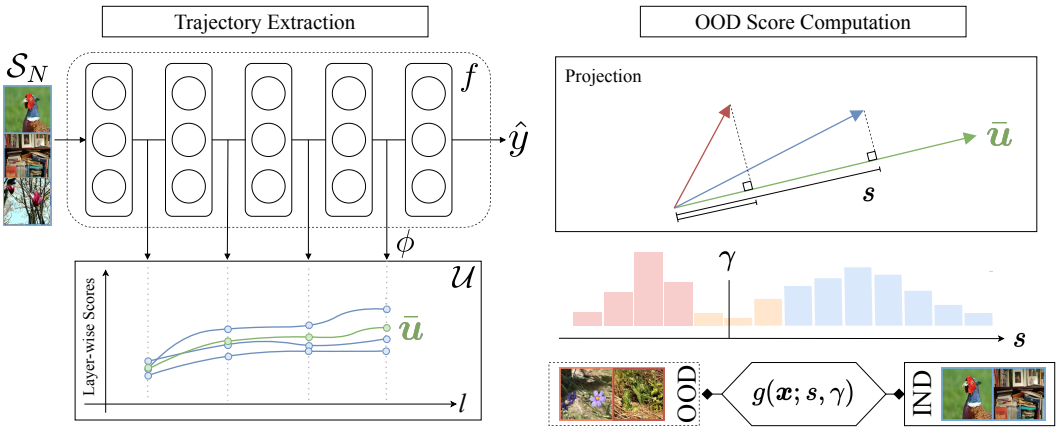


Figure 1: The left hand side shows the mapping of the hidden representations of an input sample into a functional representation. Then, during test time, we compute the OOD score s of a sample. A threshold γ is set to obtain a binary discriminator g.

whose trajectories are abnormal (or atypical) compared to reference trajectories characterized from the training set. Our method, which requires little parameter tuning and perhaps more importantly no additional OOD data, can identify OOD samples from their trajectories. Furthermore, we show that typical multivariate detection methods fail to detect OOD patterns, which may manifest in an isolated fashion by shifting in magnitude or overall shape. Figure 1 summarizes our method.

Contributions. This work presents a new principle and unsupervised method for detecting OOD samples that does not require OOD (or extra) data and brings novel insights into the problem of OOD detection. Our main contributions can be summarized as follows.

1. *A novel problem formulation*. We reformulate the problem of OOD detection through a functional perspective that effectively captures the statistical dependencies of an input sample's path across a multi-layer neural classifier. Moreover, we propose a map from the multivariate feature space (at each the layer) to a functional space which relies on the Cosine similarity between the test sample and a training reference at the layer and the sample's $\ell 2$ norm. It is computationally efficient and simple.

2. *Computing OOD scores from trajectories*. We extract the training trajectories to have a reference average trajectory on which we project test trajectories to measure their similarity w.r.t the training centroid. Low similarity indicates that the test sample is likely sampled from OOD.

3. *Empirical evaluation*. We validate the value of the proposed method using a mid-size OOD detection benchmark on ImageNet-1k. We obtain competitive results, demonstrating an average `ROC` gain of up to 3% across three architectures and four different OOD datasets. We release our code online.

2 Related Works

This section briefly discusses prior work in OOD detection, highlighting confidence-based and feature-based methods without special training as they resonate the most with our work. Another thread of research relies on re-training for learning more appropriate representations adapted to OOD detection Mohseni et al. (2020); Bitterwolf et al. (2020); Mahmood et al. (2021), either through contrastive Hendrycks et al. (2019); Winkens et al. (2020), regularization Lee et al. (2021); Nandy et al. (2021); Hein et al. (2019), generative Schlegl et al. (2017); Vernekar et al. (2019); Xiao et al. (2020); Ren et al. (2019); Zhang et al. (2021) or ensemble Vyas et al. (2018); Choi & Jang (2018) based approaches. Related subfields are open set recognition Geng et al. (2021), novelty detection Pimentel et al. (2014), anomaly detection Chandola et al. (2009); Chalapathy & Chawla (2019), outlier detection Hodge & Austin (2004), and adversarial attacks detection Akhtar & Mian (2018).

Confidence-based OOD detection. A natural measure of uncertainty of the label of a sample is the softmax output of the classification model. Hendrycks & Gimpel (2017) observed that the maximum of the softmax output could be used as a discriminative score between in-distribution and OOD samples. Even though this metric is effective in some cases, Hein et al. (2019) observed that it may also still assign overconfident values to OOD examples. Liang et al. (2018); Hsu et al. (2020) is an incremental step towards ameliorating this method. They propose re-scaling the softmax response with a temperature value and a pre-processing technique that further separates in- from out-of-distribution examples. Liu et al. (2020) proposes an energy-based OOD detection score by substituting the softmax confidence score by the free energy function.

Feature-based OOD detection. This line of research focuses on exploring latent representations for OOD detection. For instance, Haroush et al. (2021) consider using statistical tests, Sastry & Oore (2020) rely on higher-order Grams matrices, Quintanilha et al. (2019) uses mean and standard deviation within feature maps, Sun et al. (2021) proposes clipping the activations to boost OOD detection performance, while recent work Huang et al. (2021) also explores the gradient space. Normalizing and residual flows to estimate the probability distribution of the feature space were proposed in Kirichenko et al. (2020); Zisselman & Tamar (2020). Perhaps one of the most widely used techniques relies on the Gaussian mixture assumption for the hidden representations and the Mahalanobis distance Lee et al. (2018); Ren et al. (2021) or further information geometry tools Gomes et al. (2022). The reasons are its simplicity, performance, and geometrical explanation. Efforts toward combining multiple features to improve performance were previously explored Lee et al. (2018); Sastry & Oore (2020); Gomes et al. (2022). The strategy relies heavily upon having additional data for tuning the detector or focusing on specific model architectures, which are very limiting factors or even unfeasible in real-world applications.

3 PRELIMINARIES

We start by recalling the general setting of the OOD detection problem from a mathematical point of view. Then, we motivate our method through a simple illustrative example.

3.1 BACKGROUND

Let (X, Y) be a random variable valued in a space $\mathcal{X} \times \mathcal{Y}$ with unknown probability density function (pdf) p_{XY} and probability distribution P_{XY}. Here, $\mathcal{X} \subseteq \mathbb{R}^d$ represents the covariate space and $\mathcal{Y} = \{1, \ldots, C\}$ corresponds to the labels attached to elements from $\mathcal{X}$. The training dataset $\mathcal{S}_N = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ is defined as independent and identically distributed (i.i.d) realizations of P_{XY}. From this formulation, detecting OOD samples boils down to building a binary rule $g : \mathcal{X} \to \{0, 1\}$ through a soft scoring function $s : \mathcal{X} \to \mathbb{R}$ and a threshold $\gamma \in \mathbb{R}$. Namely, a new observation $\boldsymbol{x} \in \mathcal{X}$ is then considered as *in-distribution*, i.e., generated by P_{XY}, when $g(\boldsymbol{x}) = 0$ and as OOD when $g(\boldsymbol{x}) = 1$. Finding this rule g from $\mathcal{X}$ can become intractable when the dimension d is large. Thus, previous work rely on a multi-layer pre-trained classifier $f_\theta : \mathcal{X} \to \mathcal{Y}$ defined as:

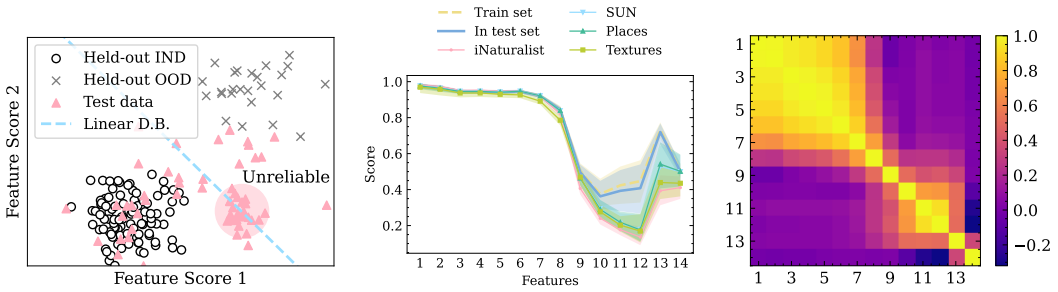
$$f_\theta(\cdot) = h \circ f_L \circ f_{L-1} \circ \cdots \circ f_1(\cdot), \tag{1}$$

with $L \geq 1$ layers, where $f_\ell : \mathbb{R}^{d_{\ell-1}} \to \mathbb{R}^{d_\ell}$ is the ℓ-th layer of the multi-layer neural classifier, d_ℓ denotes the dimension of the latent space induced by the ℓ-th layer ($d_0 = d$) and h indicates the classifier that outputs the logits. We also define $\boldsymbol{z}_\ell = (f_\ell \circ \cdots \circ f_1)(\boldsymbol{x})$ as the latent vectorial representation at the $\ell-$th layer for an input sample $\boldsymbol{x}$. We will refer to the logits as $\boldsymbol{z}_{L+1}$ and h as f_{L+1} to homogenize notation. It is worth emphasizing that the trajectory of $(\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_{L+1})$ corresponding to a test input $\boldsymbol{x}_0$ are dependent random variables whose joint distribution strongly depends on the underlying distribution of the input.

Therefore, the constructed function $g(\cdot)$ is typically based on the three key steps:

(i) A similarity measure $\mathrm{d}(\cdot\,;\cdot)$ (e.g., Cosine similarity, Mahalanobis distance, etc.) between a sample and a population is applied at each layer to measure the similarity (or dissimilarity) of a test input $\boldsymbol{x}_0$ at the ℓ-th layer $\boldsymbol{z}_{\ell,0} = (f_\ell \circ \cdots \circ f_1)(\boldsymbol{x}_0)$ w.r.t. the population of the training examples observed at the same layer $\left\{\boldsymbol{z}_\ell = (f_\ell \circ \cdots \circ f_1)(\boldsymbol{x}) \,:\, \boldsymbol{x} \in \mathcal{S}_N\right\}$.

(ii) The layer-wise score obtained is then mapped to the real line collecting the OOD scores.



(a) Example of a mispecified model in a toy example in 2D caused by held-out OOD dataset.

(b) Trajectory of data through a network with 25% and 75% percentile bounds.

(c) Correlation between layers training scores in a network, highlighting structure in the trajectories.

Figure 2: Figure 2a summarizes the limitation of current supervised methods for aggregating layer scores that rely on held-out OOD or pseudo-OOD data. It biases the decision boundary (D.B) that doesn't generalize well to other types of OOD data. We observed that in-distribution and OOD data have disparate trajectories through a network (Figure 2b). We can observe a clear separation on the last 5 features. Also, these features are correlated in a sequential fashion as observed in Figure 2c.

(iii) Lastly, a threshold is chosen to build the final decision function.

A fundamental ingredient remains in step (ii): *how to efficiently leverage the information collected from multiple layers in an unsupervised manner, without resorting to OOD or pseudo-OOD examples?*

3.2 From Independent Multi-Layer Scores to a Sequential Perspective of OOD Detection

State-of-the-art methods treat step (ii) as a supervised learning problem for which the solution is a linear binary classifier. In a nutshell, the aim is to find a linear combination of the various scores obtained at each layer that will sufficiently separate in-distribution from OOD samples. A held-out OOD dataset is collected from true (or pseudo-generated) OOD samples. The linear soft novelty score function s_α writes:

$$s_\alpha(\boldsymbol{x}_0) = \sum_{\ell=1}^{L} \alpha_\ell \cdot \mathrm{d}\left(\boldsymbol{x}_0; \left\{(f_\ell \circ \cdots \circ f_1)(\boldsymbol{x}) \,:\, \boldsymbol{x} \in \mathcal{S}_N\right\}\right).$$

The shortcomings of this method are the need for extra data or ad-hoc parameters, which results in decision boundaries that underfit the problem and fail to capture certain types of OOD samples. To illustrate this phenomenon, we designed a toy example (see Figure 2a) where scores are extracted from two features fitting a linear discriminator on held-out in-distribution (IND) and OOD samples. We show an area of unreliable predictions where OOD samples cannot be detected due to the misspecification of the linear model. Of course, one could introduce a non-linear discriminator that better captures the geometry of the data for this 2D toy example. However, it becomes challenging as we move to higher dimensions with limited data.

The sequential nature of feature extraction matters. By reformulating the problem from a functional point of view, we can identify trends and typical trajectories presented in data. Figure 2 shows the dispersion of trajectories at step (ii) coming from the in-distribution and OOD samples. These patterns are extracted from multiple latent representations and aligned on a time-series-like object, clearly distinguishing training samples. *We observed that trajectories coming from OOD samples exhibit a different shape when compared to typical trajectories from training data*. Thus, to determine if an instance belongs to a reference distribution, we can test if the observed path is close to the functional data extracted from the training set. We will do so class-wise as we plan to retain the most structural information about the downstream task while showcasing the relevant shape differences.

4 FUNCTIONAL OUT-OF-DISTRIBUTION DETECTION

In this section we present our OOD detection framework which applies to any pre-trained multi-layer neural network with no requirements of OOD samples. We describe our method through two key steps: functional representation of the input sample (see Section 4.1) and test time OOD score computation (see Section 4.2).

4.1 TOWARDS A FUNCTIONAL REPRESENTATION

The first step to obtain a functional representation of the data from the multivariate hidden representations is to reduce each one of them to a scalar or vector. To do so in a meaningful way for the task in hand, we first compute the training population references: layer-wise class-conditional average vectors:

$$\boldsymbol{\mu}_{\ell,y} = \frac{1}{N_y} \sum_{i=1}^{N_y} \boldsymbol{z}_{\ell,i} \tag{2}$$

where $N_y = \left|\{\boldsymbol{z}_{\ell,i} \mid y_i = y, \forall i \in \{1..N\}\}\right|$, $1 \leq \ell \leq L+1$ and $\boldsymbol{z}_{\ell,i} = (f_\ell \circ \cdots \circ f_1)(\boldsymbol{x}_i)$.

Given an arbitrary input example, we compute its features and their similarity w.r.t. the respective training average for the predicted class $\hat{y}$ (including the logits), leading to $L+1$ scores. We rely on the scalar projection[1] the sample features' $\boldsymbol{z}_\ell$ onto the direction of the average training feature for the predicted class as similarity score.

$$\mathrm{d}(\boldsymbol{z}_\ell, \boldsymbol{\mu}_{\ell,\hat{y}}) = \mathrm{proj}_{\boldsymbol{\mu}_{\ell,\hat{y}}} \mathrm{z}_\ell = \mathrm{z}_\ell \cdot \frac{\boldsymbol{\mu}_{\ell,\hat{\mathrm{y}}}}{\|\boldsymbol{\mu}_{\ell,\hat{\mathrm{y}}}\|} = \|\mathrm{z}_\ell\| \cos\big(\angle\left(\mathrm{z}_\ell, \boldsymbol{\mu}_{\ell,\hat{\mathrm{y}}}\right)\big) \tag{3}$$

where $\|\cdot\|$ is the $\ell 2$-norm and $\angle(\cdot,\cdot)$ is the angle between two vectors. Hence, our layer-wise scores simply relies on the notions of vector length and angle between vectors, which can be generalized to any n-dimensional inner product space, without imposing any geometrical constraints to it.

It is worth emphasizing that our layer score possesses some advantages when compared to the class conditional Gaussian model first introduced in Lee et al. (2018) and the Gram matrix based method introduced in Sastry & Oore (2020). Our layer score inglobe a wider class of distributions as we do not suppose an underlying multivariate mixture of Gaussians. Lie so, we avoid computing covariance matrices, which is often ill-conditioned for latent representations of DNNs. Since we do not store covariance matrices, our functional approach has a smaller overhead in terms of memory requirements. Also, our method can also be applied to any vector-based hidden representation, not being restricted to matrix-based representations as in Sastry & Oore (2020). Thus, our method apply to a wider range of models, including transformers, which the hidden representation is usually taken as the [CLS] latent vector-token.

By computing the scalar projection at each layer, we obtain the following *functional neural-representation* extraction function:

$$\begin{aligned} \phi : \mathcal{X} &\to \mathbb{R}^{L+1} \\ \boldsymbol{x} &\mapsto \left[\mathrm{d}\left(\boldsymbol{z}_1, \boldsymbol{\mu}_{1,\hat{y}}\right), \ldots, \mathrm{d}\left(\boldsymbol{z}_L + 1, \boldsymbol{\mu}_{L+1,\hat{y}}\right)\right]. \end{aligned} \tag{4}$$

We apply ϕ to the training input $\boldsymbol{x}_i$ to obtain the representation of the training sample across the network $\boldsymbol{u}_i = \phi(\boldsymbol{x}_i)$. We consider the related vectors $\boldsymbol{u}_i, \forall\, i \in [1 : N]$ as curves parameterized by the layer of the network. Thus, we can map samples from the representation to a functional space while retaining information on the typicality of the sample w.r.t the training dataset. We build a training datasets from these functional representations that will be useful for detecting OOD samples in test time:

$$\mathcal{U} = \{\boldsymbol{u}_i\}_{i=1}^N. \tag{5}$$

We first rescale the training set trajectories w.r.t the maximum value found at each coordinate in order to obtain layer-wise scores more or less on the same scaling for each coordinate so that it is easier

[1] Other metrics to measure similarity of an input w.r.t. the population of examples can also be used.

to combine scores. Hence, for $j \in \{1, \dots, L+1\}$, let $\max(\mathcal{U}) := [\max_i u_{i,1}, \dots, \max_i u_{i,L+1}]^\top$, we apply the following re-scaling on the functional training set:

$$\widetilde{\mathcal{U}} = \left\{ \frac{\boldsymbol{u}_i}{\max(\mathcal{U})} \right\}_{i=1}^{N} \tag{6}$$

From the re-scaled set, we can compute a reference trajectory $\bar{\boldsymbol{u}}$ for the entire training dataset defined in equation 7 that will serve as a global *typical reference* to compare test trajectories with.

$$\bar{\boldsymbol{u}} = \frac{1}{N} \sum_{i=1}^{N} \tilde{\boldsymbol{u}}_i \tag{7}$$

4.2 Computing the OOD Score at Test Time

At inference time, we first re-scale the test sample's trajectory as we did with the training reference:

$$\tilde{\phi}(\boldsymbol{x}) = \frac{\phi(\boldsymbol{x})}{\max(\mathcal{U})}. \tag{8}$$

Then, we compute a similarity score w.r.t this typical reference that will be our OOD score. We choose as metric also the scalar projection of the test vector to the training reference. In practical terms, it boils down to the *inner product* between the test sample's trajectory and the training set's typical reference trajectory, since the norm of the average trajectory is constant for all test samples. Mathematically, our scoring function $s : \mathcal{X} \mapsto \mathbb{R}$ writes:

$$s(\boldsymbol{x}; \bar{\boldsymbol{u}}) = \left\langle \tilde{\phi}(\boldsymbol{x}), \bar{\boldsymbol{u}} \right\rangle = \sum_{j=1}^{L+1} \tilde{\phi}(\boldsymbol{x})_j \bar{\boldsymbol{u}}_j, \tag{9}$$

From the OOD score we can derive a binary classifier g by fixing a threshold $\gamma \in \mathbb{R}$:

$$g(\boldsymbol{x}; s, \gamma) = \begin{cases} 1, & \text{if } s(\boldsymbol{x}) \leq \gamma \\ 0, & \text{otherwise,} \end{cases} \tag{10}$$

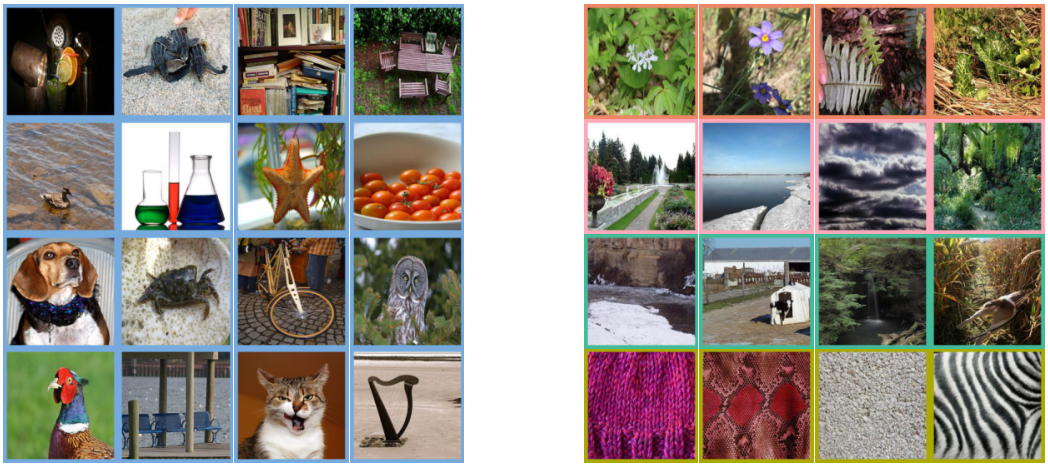
where $g(\boldsymbol{x}) = 1$ means that the input sample $\boldsymbol{x}$ is classified as being out-of-distribution. Please refer to the Appendix (see Section A.1) for further details on the algorithm.

5 Experimental Setting

In this section, we describe the experimental setting, including the datasets used, the pre-trained DNN architectures, the evaluation metrics, and other essential information. We open-source our code in `https://github.com/ood-trajectory/ood-trajectory`.

5.1 Datasets

We set as *in-distribution* dataset *ImageNet-1k* Deng et al. (2009) (= ILSVRC2012) for all of our experiments. This is a challenging mid-size and realistic dataset that gained interest on recent literature in OOD detection Huang & Li (2021). It contains around 1.28M training samples and 50,000 test samples with a total of 1000 different classes. For the *out-of-distribution* datasets, we take the same dataset splits introduced by Huang & Li (2021). The *iNaturalist* Horn et al. (2017) dataset contains over 5,000 species of plant and animals. We consider a split with 10,000 test samples with concepts coming from 110 classes different from the in-distribution ones. The *Sun* Xiao et al. (2010) dataset is a scene dataset of 397 categories. We considered a split with 10,000 randomly sampled test examples belonging to 50 different categories. The *Places365* Zhou et al. (2017) is also a scenes dataset with 365 different concepts. We also considered a random split with 10,000 samples from 50 disjoint categories. For the DTD or *Textures* Cimpoi et al. (2014) dataset is composed of textural patterns. We considered all of the 5,640 available test samples. Figure 3 shows a few examples from these datasets. Note that there is an overlap between the class zebra from ImageNet and the zebra texture from the DTD dataset and others. Whether or not these classes could be considered an overlap, we decided to keep this exact partition of the dataset to be better comparable with Huang et al. (2021).



(a) In-distribution.

(b) Out-of-distribution.

Figure 3: Example of data samples from in-distribution (ImageNet) and OOD samples from iNaturalist, SUN, Places and Textures datasets.

5.2 MODELS

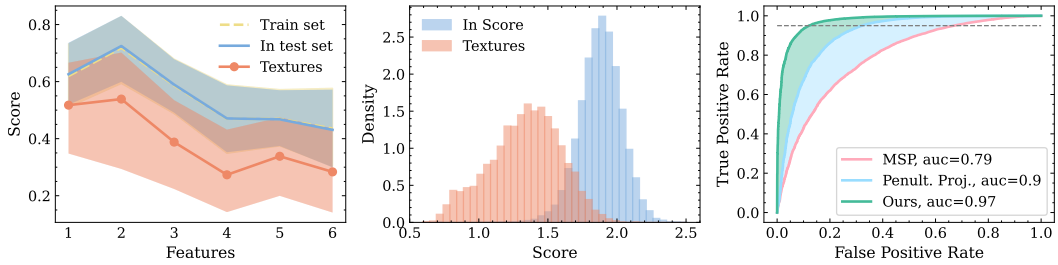
We ran experiments with three models of different architectures. A *DenseNet-121* Huang et al. (2017) pre-trained on ILSVRC-2012 with 8M parameters and test set top-1 accuracy of 74.43%. We reduced the intermediate representations from the transition blocks 1,2 and 3, with an average *adaptive max pooling* obtaining a final vector with dimension equal to the number of channels of each convolution output. The resulting dimensions are 128, 256, 512, and 1024, respectively. We resize input images to 224×224 pixels. A *BiT-S-101* Kolesnikov et al. (2020) model based on a ResNet-101 He et al. (2016) pre-trained on ILSVRC-2012 with top-1 test set accuracy of 77.41% and 44.5M parameters. We extract features from the outputs of layers 1 to 4 and the penultimate layer after max pooling, obtaining representations with sizes 256, 512, 1024, and 2048, respectively. We resize input images to 480×480. We also ran experiments with a large Vision Transformer (*ViT-16-B*) Dosovitskiy et al. (2021), which is a trained on the ILSVRC2012 dataset with 82.64% top-1 test accuracy and 70M parameters. We take the output's class tokens for layers 1 to 13 and the encoder's output as latent representations, totaling 14 features of dimension 768 each. We resize images to 224×224. We download the DenseNet and Vision Transformer weights from PyTorch Paszke et al. (2019) hub and the weights of Big Transfer (ResNet-101) from Kolesnikov et al. (2020). They are all trained from scratch on ImageNet-1k.

5.3 EVALUATION METRICS

For the primary benchmark, we apply the established OOD evaluation metrics introduced in previous works of OOD detection Hendrycks & Gimpel (2017). We evaluate the methods in terms of `ROC` and `TNR`. The Area Under The Receiving Operation Curve (`ROC`) is the Area Under the curve representing the true negative rate against the false positive rate when considering a varying threshold. It measures how well can the OOD score distinguish between out- and in-distribution data in a threshold-independent manner. The True Negative Rate at 95% True Positive Rate (TNR at 95% TPR or `TNR` for short) is a threshold-dependent metric that provides the performance of the detector in a reasonable choice of novelty threshold. It measures the accuracy in detecting OOD samples when the accuracy of detecting in-distribution samples is fixed at 95%. For both measures, the higher, the better.

5.4 BASELINE HYPERPARAMETERS

For ODIN Liang et al. (2018), we set the temperature to 1000 and noise magnitude to zero. For Energy Liu et al. (2020), we take temperature equal to one. For GradNorm Huang et al. (2021)



(a) Functional trajectories. (b) OOD score histogram. (c) ROC curve.

Figure 4: Functional representation with 5 and 95% quantiles, histogram and ROC curve for our OOD score on a DenseNet-121 model with Textures as OOD dataset.

(GradN. for short) we set the temperature to one. For Mahalanobis Lee et al. (2018) (Maha. for short), we take only the scores of the outputs of the penultimate layer of the network. The MSP Hendrycks & Gimpel (2017) does not have any hyperparameters. This setup is identical to the one presented in Huang et al. (2021).

6 Results and Discussion

We report our main results in Table 1 which includes the evaluation metrics for the three pre-trained architectures, four OOD datasets, and six detection methods. Our results are across the board consistent, obtaining an AUROC of 90% on average (see Table 1). It improves existing methods by a margin on all datasets and models.

Table 1: Comparison against post-hoc state-of-the-art methods for OOD detection on the ImageNet benchmark. M+Ours stands for using the Mahalanobis distance based layer score with our proposed unsupervised score aggregation algorithm based on trajectory similarity. Values are in percentage.

		iNaturalist		SUN		Places		Textures		Average	
		TNR	ROC	TNR	ROC	TNR	ROC	TNR	ROC	TNR	ROC
BiT-S-101	MSP	35.9	87.9	28.8	81.9	21.3	79.3	21.9	77.5	27.0	81.7
	ODIN	29.3	86.7	36.7	86.8	26.0	82.7	24.1	79.3	29.0	83.9
	Energy	25.0	84.5	39.8	87.3	27.2	82.7	24.2	78.8	29.0	83.3
	Maha.	16.5	78.3	13.2	74.5	10.5	69.6	**86.6**	**97.3**	31.7	79.9
	GradN.	41.3	86.0	**55.2**	**88.2**	**39.0**	**83.3**	41.2	81.0	44.2	84.6
	Ours	**62.1**	**89.8**	50.7	87.0	34.0	79.6	83.9	96.3	**57.7**	**88.2**
DenseNet-121	MSP	50.7	89.1	33.0	81.5	30.8	81.1	32.9	79.2	36.9	82.7
	ODIN	60.4	92.8	45.2	87.0	40.3	85.1	45.3	85.0	47.8	87.5
	Energy	60.3	92.7	48.0	87.4	42.2	85.2	47.9	85.4	49.6	87.7
	Maha.	3.5	59.7	4.8	57.0	4.6	54.8	54.4	88.3	16.8	64.9
	GradN.	**73.3**	**93.4**	59.1	88.8	48.0	84.1	56.7	87.7	59.3	88.5
	Ours	61.5	91.6	**65.3**	**91.3**	**49.8**	**86.2**	**88.0**	**97.3**	**66.2**	**91.6**
ViT-B-16	MSP	48.5	88.2	33.5	80.9	31.3	80.4	39.8	83.0	38.3	83.1
	ODIN	49.9	86.0	31.5	75.2	33.7	76.5	42.6	81.2	39.4	79.7
	Energy	35.9	79.2	27.2	70.2	25.7	68.4	41.5	79.3	32.6	74.3
	Maha.	**81.2**	**96.0**	**40.7**	**85.3**	**40.0**	**84.2**	46.3	87.5	52.1	88.2
	GradN.	53.5	91.2	41.5	85.3	38.3	83.4	48.1	86.5	45.3	86.6
	Ours	54.0	92.8	27.1	80.2	27.6	78.9	54.5	90.6	40.8	85.6
	M+Ours	77.9	95.5	34.8	83.2	32.6	81.4	**79.1**	**94.9**	**56.1**	**88.8**

Multivariate OOD Scores is Not Enough. Even though well-known multivariate novelty (or anomaly) detection techniques, such as One-class SVM Cortes & Vapnik (1995), Isolation Forest Liu et al. (2008), Extended Isolation Forest Hariri et al. (2021), Local Outlier Factor (LOF; Breunig et al. (2000)), k-NN approaches Fix & Hodges (1989) and distance-based approaches Mahalanobis (1936) are adapted to various scenarios, they showed to be inefficient for integrating layers' information. Unfortunately, classic multivariate methods cannot capture this inter-layer dependence effectively. A hypothesis that explains this failure is the important sequential dependence pattern we noticed in the in-distribution layer-wise scores.

Table 2 shows the performance of a few unsupervised aggregation methods based on a multivariate OOD detection paradigm. We tried typical methods: evaluating the Euclidean distance w.r.t the training set Gaussian representation, fitting an Isolation Forest, and a One-class SVM. We compare the results with the performance of taking *only* the penultimate layer Mahalanobis Lee et al. (2018) scores, and we observe that the multivariate aggregation fails to improve the scores in three out of four datasets for ViT-16-B.

Table 2: The first two rows show single layer baselines where *proj. penult.* is our layer score on the penultimate layer of the model. The subsequent rows show performance of other unsupervised multivariate aggregation methods and our method. We ran experiments with a ViT-16-B model.

	iNaturalist		SUN		Places		Textures		Average	
	TNR	ROC	TNR	ROC	TNR	ROC	TNR	ROC	TNR	ROC
Grad Norm	73.3	93.4	59.1	88.8	48.0	84.1	56.7	87.7	59.3	88.5
Proj. penult. (Ours)	**76.0**	**94.4**	59.8	89.2	49.2	85.0	67.8	90.3	63.2	89.7
Euclidean distance	57.1	86.2	47.6	82.5	35.2	75.7	82.4	95.2	55.6	84.9
Isolation Forest	10.7	60.7	6.7	57.5	5.3	55.4	6.3	53.0	7.2	56.6
One Class SVM	10.6	64.8	7.9	65.0	8.1	65.4	5.5	55.9	8.0	62.8
Proj. (Ours)	61.5	91.6	**65.3**	**91.3**	**49.8**	**86.2**	**88.0**	**97.3**	**66.2**	**91.6**

Qualitative Evaluation of the Constructed Functional Dataset. The test in-distribution trajectories follows a well defined trend very similar to the training distribution (see Figure 4a). While the OOD trajectories manifest mainly as shape and magnitude anomalies (w.r.t. the taxonomy of Hubert et al. (2015)). These characteristics are reflected on the histogram of our detection score (see Figure 4b). The in-distribution histogram is generally symmetric, while the histogram for the OOD data is typically skewed and has a smaller average. This difference results to excellent performance and steep ROC curves (see Figure 4c). Please refer to the Appendix A.4 for further similar figures.

Potential Shortcomings of our Method. We only show mild ameliorations on transformers, that encodes most of the useful information for OOD detection on the encoder output, at least for this benchmark. This indicates there's room for improvement on the aggregation of scores multiple layers of a transformer based model. Future directions could be exploring the self-attention matrices for OOD detection.

7 Conclusion

In this work, we introduced an original approach to perform OOD detection based on a functional view of the pre-trained multi-layer neural network that leverages the sample's trajectories through the various layers. In a simple yet original way, we shifted the problem from a multivariate to a functional perspective. Our method detects samples whose trajectories differ from the typical behavior characterized from the training set. The key ingredient relies on the statistical dependencies of the scores extracted at each layer, using a purely self-supervised algorithm, which makes subsequent OOD detection task much more manageable. Beyond the novelty and practical advantages of the algorithm we developed, our results establish the value of using a functional approach as an unsupervised technique to detect OOD samples, which offers an exciting alternative to usual single-layer approaches.

REFERENCES

Naveed Akhtar and Ajmal Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *CoRR*, abs/1801.00553, 2018. URL http://arxiv.org/abs/1801.00553.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016a.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul F. Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *CoRR*, abs/1606.06565, 2016b. URL http://arxiv.org/abs/1606.06565.

Julian Bitterwolf, Alexander Meinke, and Matthias Hein. Certifiably adversarially robust detection of out-of-distribution data. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 16085–16095. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/b90c46963248e6d7aab1e0f429743ca0-Paper.pdf.

Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. Lof: Identifying density-based local outliers. *SIGMOD Rec.*, 29(2):93–104, may 2000. ISSN 0163-5808. doi: 10.1145/335191.335388. URL https://doi.org/10.1145/335191.335388.

Raghavendra Chalapathy and Sanjay Chawla. Deep learning for anomaly detection: A survey. *CoRR*, abs/1901.03407, 2019. URL http://arxiv.org/abs/1901.03407.

Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3), jul 2009. ISSN 0360-0300. doi: 10.1145/1541880.1541882. URL https://doi.org/10.1145/1541880.1541882.

Hyun-Jae Choi and Eric Jang. Generative ensembles for robust anomaly detection. *ArXiv*, abs/1810.01392, 2018.

M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ArXiv*, abs/2010.11929, 2021.

Evelyn Fix and J. L. Hodges. Discriminatory analysis. nonparametric discrimination: Consistency properties. *International Statistical Review / Revue Internationale de Statistique*, 57(3):238–247, 1989. ISSN 03067734, 17515823. URL http://www.jstor.org/stable/1403797.

Chuanxing Geng, Sheng-Jun Huang, and Songcan Chen. Recent advances in open set recognition: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(10):3614–3631, oct 2021. doi: 10.1109/tpami.2020.2981604. URL https://doi.org/10.1109%2Ftpami.2020.2981604.

Eduardo Dadalto Camara Gomes, Florence Alberge, Pierre Duhamel, and Pablo Piantanida. Igeood: An information geometry approach to out-of-distribution detection. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=mfwdY3U_9ea.

Sahand Hariri, Matias Carrasco Kind, and Robert J. Brunner. Extended isolation forest. *IEEE Transactions on Knowledge and Data Engineering*, 33:1479–1489, 2021.

Matan Haroush, Tzviel Frostig, Ruth Heller, and Daniel Soudry. A statistical framework for efficient out of distribution detection in deep neural networks, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90.

Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 41–50, 2019.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2017.

Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=HyxCxhRcY7`.

Victoria Hodge and Jim Austin. A survey of outlier detection methodologies. *Artif. Intell. Rev.*, 22(2):85–126, oct 2004. ISSN 0269-2821. doi: 10.1023/B:AIRE.0000045502.10941.a9. URL `https://doi.org/10.1023/B:AIRE.0000045502.10941.a9`.

Grant Van Horn, Oisin Mac Aodha, Yang Song, Alexander Shepard, Hartwig Adam, Pietro Perona, and Serge J. Belongie. The inaturalist challenge 2017 dataset. *ArXiv*, abs/1707.06642, 2017.

Yen-Chang Hsu, Yilin Shen, Hongxia Jin, and Zsolt Kira. Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10948–10957, 2020.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269, 2017. doi: 10.1109/CVPR.2017.243.

Rui Huang and Yixuan Li. Mos: Towards scaling out-of-distribution detection for large semantic space. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8706–8715, 2021.

Rui Huang, Andrew Geng, and Yixuan Li. On the importance of gradients for detecting distributional shifts in the wild. *ArXiv*, abs/2110.00218, 2021.

M. Hubert, P.J. Rousseeuw, and Pieter Segaert. Multivariate functional outlier detection. *Statistical Methods and Applications*, 24(2):177–202, 2015. ISSN 1613-981X. doi: 10.1007/s10260-015-0297-8. URL `https://doi.org/10.1007/s10260-015-0297-8`.

Polina Kirichenko, Pavel Izmailov, and Andrew G Wilson. Why normalizing flows fail to detect out-of-distribution data. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 20578–20589. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/ecb9fe2fbb99c31f567e9823e884dbec-Paper.pdf`.

Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *ECCV*, 2020.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 31*, pp. 7167–7177. Curran Associates, Inc., 2018. URL `http://papers.nips.cc/paper/7947-a-simple-unified-framework-for-detecting-out-of-distribution-samples-and-adve` `pdf`.

Saehyung Lee, Changhwa Park, Hyungyu Lee, Jihun Yi, Jonghyun Lee, and Sungroh Yoon. Removing undesirable feature contributions using out-of-distribution data. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=eIHYL6fpbkA.

Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1VGkIxRZ.

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pp. 413–422, 2008. doi: 10.1109/ICDM.2008.17.

Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems*, 2020.

Prasanta Chandra Mahalanobis. On the generalized distance in statistics. *Proceedings of the National Institute of Sciences (Calcutta)*, 2:49–55, 1936.

Ahsan Mahmood, Junier Oliva, and Martin Styner. Multiscale score matching for out-of-distribution detection, 2021.

Sina Mohseni, Mandar Pitale, JBS Yadawa, and Zhangyang Wang. Self-supervised learning for generalizable out-of-distribution detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):5216–5223, Apr. 2020. doi: 10.1609/aaai.v34i04.5966. URL https://ojs.aaai.org/index.php/AAAI/article/view/5966.

Jay Nandy, Wynne Hsu, and Mong Li Lee. Towards maximizing the representation gap between in-domain & out-of-distribution examples, 2021.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Marco AF Pimentel, David A Clifton, Lei Clifton, and Lionel Tarassenko. A review of novelty detection. *Signal processing*, 99:215–249, 2014.

Igor M. Quintanilha, Roberto de M. E. Filho, José Lezama, Mauricio Delbracio, and Leonardo O. Nunes. Detecting out-of-distribution samples using low-order deep features statistics, 2019. URL https://openreview.net/forum?id=rkgpCoRctm.

Jie Ren, Peter J. Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark Depristo, Joshua Dillon, and Balaji Lakshminarayanan. Likelihood ratios for out-of-distribution detection. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/1e79596878b2320cac26dd792a6c51c9-Paper.pdf.

Jie Ren, Stanislav Fort, Jeremiah Liu, Abhijit Guha Roy, Shreyas Padhy, and Balaji Lakshminarayanan. A simple fix to mahalanobis distance for improving near-ood detection, 2021.

Chandramouli Shama Sastry and Sageev Oore. Detecting out-of-distribution examples with Gram matrices. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8491–8501. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/sastry20a.html.

Thomas Schlegl, Philipp Seeböck, Sebastian M. Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery, 2017.

Adarsh Subbaswamy and Suchi Saria. From development to deployment: dataset shift, causality, and shift-stable models in health ai. *Biostatistics*, 21(2):345–352, April 2020. ISSN 1465-4644. doi: 10.1093/biostatistics/kxz041.

Yiyou Sun, Chuan Guo, and Yixuan Li. React: Out-of-distribution detection with rectified activations. *ArXiv*, abs/2111.12797, 2021.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 ieee information theory workshop (itw)*, pp. 1–5. IEEE, 2015.

Sachin Vernekar, Ashish Gaurav, Vahdat Abdelzad, Taylor Denouden, Rick Salay, and Krzysztof Czarnecki. Out-of-distribution detection in classifiers via generation. In *Neural Information Processing Systems (NeurIPS 2019), Safety and Robustness in Decision Making Workshop*. https://sites.google.com/view/neurips19-safe-robust-workshop, https://sites.google.com/view/neurips19-safe-robust-workshop, 12/2019 2019. URL https://drive.google.com/file/d/0B3mY6u_lryzdel9WOW1XTVA0aDIwazJDcG9ORlZrZWFOd0xJ/view.

Apoorv Vyas, Nataraj Jammalamadaka, Xia Zhu, Dipankar Das, Bharat Kaul, and Theodore L. Willke. Out-of-distribution detection using an ensemble of self supervised leave-out classifiers. In *ECCV (8)*, pp. 560–574, 2018. URL https://doi.org/10.1007/978-3-030-01237-3_34.

Jim Winkens, Rudy Bunel, Abhijit Guha Roy, Robert Stanforth, Vivek Natarajan, Joseph R. Ledsam, Patricia MacWilliams, Pushmeet Kohli, Alan Karthikesalingam, Simon A. A. Kohl, taylan. cemgil, S. M. Ali Eslami, and Olaf Ronneberger. Contrastive training for improved out-of-distribution detection. *ArXiv*, abs/2007.05566, 2020.

J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3485–3492, 2010. doi: 10.1109/CVPR.2010.5539970.

Zhisheng Xiao, Qing Yan, and Yali Amit. Likelihood regret: An out-of-distribution detection score for variational auto-encoder. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 20685–20696. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/eddea82ad2755b24c4e168c5fc2ebd40-Paper.pdf.

Yufeng Zhang, Wanwei Liu, Zhenbang Chen, Ji Wang, Zhiming Liu, Kenli Li, and Hongmei Wei. Out-of-distribution detection with distance guarantee in deep generative models, 2021.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

Ev Zisselman and Aviv Tamar. Deep residual flow for out of distribution detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

A APPENDIX

A.1 ALGORITHM AND COMPUTATIONAL DETAILS

In this section we introduce further details on the computation algorithm and resources. Algorithms 1, describe how to extract the neural functional representations from the samples and compute the OOD score from test samples, respectively.

...

Algorithm 1: Neural functional representation extractor algorithm.

Input : Training dataset $\mathcal{S}_N = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ and a pre-trained DNN $f = f_L \circ \cdots \circ f_1$.
Output: Empirical mean vectors $\boldsymbol{\mu}_\ell$, empirical covariance matrices Σ_ℓ, and class conditional training dataset from functional representations $\mathcal{U}_y$.

for $\ell \in \{1, \ldots, L\}$ **do**
  $\boldsymbol{z}_{\ell,i} \leftarrow (f_\ell \circ \cdots \circ f_1)(\boldsymbol{x}_i)$
  $\boldsymbol{\mu}_\ell \leftarrow \frac{1}{N} \sum_{i=1}^N \boldsymbol{z}_{\ell,i}$
end
for $y \in \mathcal{Y}$ **do**
  for $i \in \{1, \ldots, N\}$ ***where*** $y_i = y$ **do**
    `// Functional Feature Extractor` ϕ
    $\boldsymbol{u}_i \leftarrow [\mathrm{d}(\boldsymbol{z}_{1,i}; \boldsymbol{\mu}_1), \ldots, \mathrm{d}(\boldsymbol{z}_{L,i}; \boldsymbol{\mu}_{L+1})]$
  end
  $\mathcal{U}_y \leftarrow \{\boldsymbol{u}_i : y_i = y\}_{i=1}^N$
end

A.2 LATENT FEATURES

DenseNet-121

```
['body.block1', 'body.block2', 'body.block3', 'body.block4',
'head.flatten']
```

BiT-S-101

```
['features.transition1.pool', 'features.transition2.pool',
'features.transition3.pool', 'features.norm5', 'flatten',
'classifier']
```

ViT-B-16

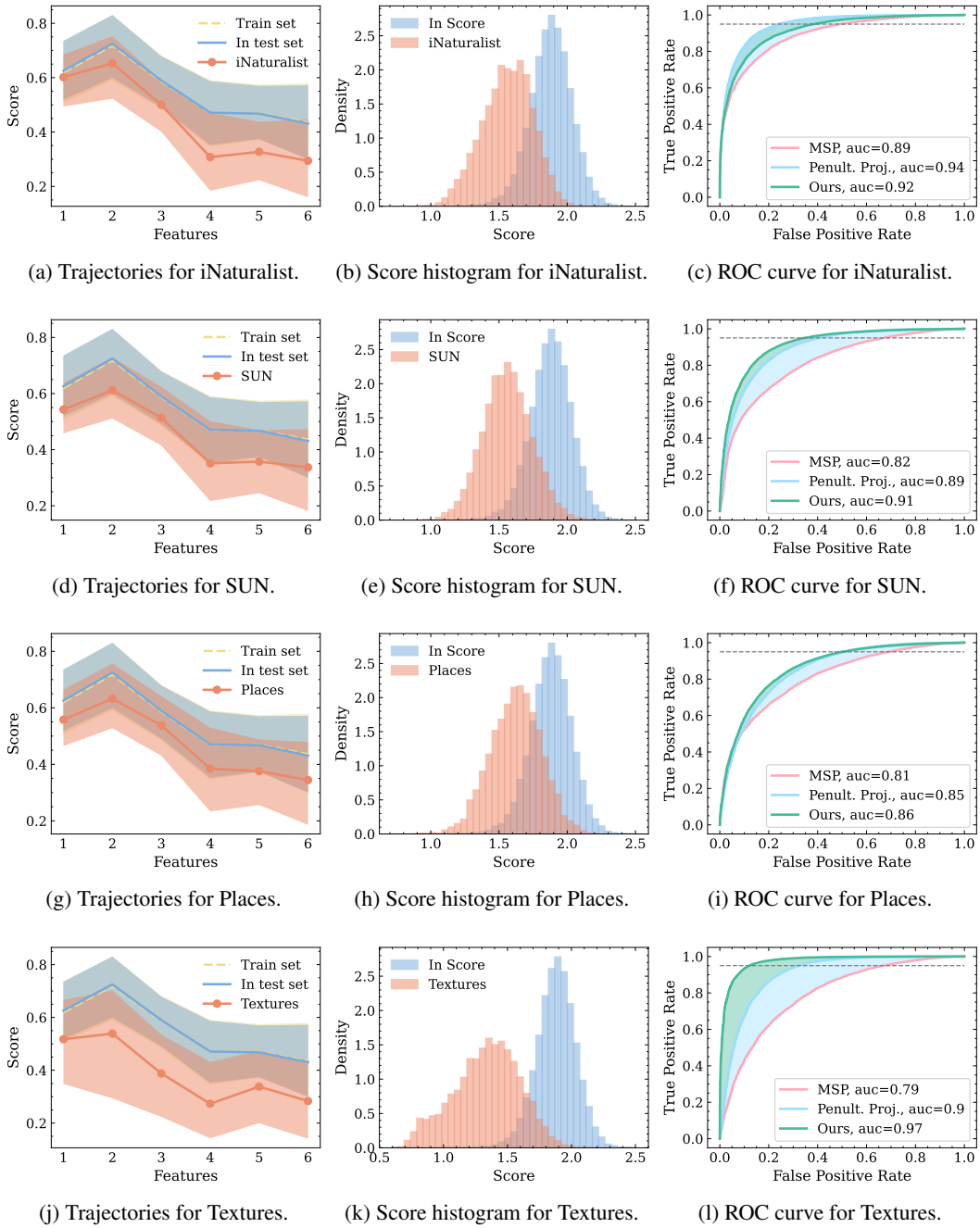
```
['encoder.layers.encoder_layer_0', 'encoder.layers.encoder_layer_1',
'encoder.layers.encoder_layer_2', 'encoder.layers.encoder_layer_3',
'encoder.layers.encoder_layer_4', 'encoder.layers.encoder_layer_5',
'encoder.layers.encoder_layer_6', 'encoder.layers.encoder_layer_7',
'encoder.layers.encoder_layer_8', 'encoder.layers.encoder_layer_9',
'encoder.layers.encoder_layer_10', 'encoder.layers.encoder_layer_11',
'encoder.ln', 'getitem_5', 'heads.head']
```

A.3 COMPUTING RESOURCES

We run our experiments on an internal cluster. Since we leverage pre-trained popular models, we were not constrained to retrain the deep models. Thus, our results should be reproducible with a single GPU. Since we are dealing with ImageNet datasets (approximately 150GB), a large storage capacity is expected. We save the features in memory to speed up downstream analysis of the algorithm, which may occupy around 200GB of storage.

A.4 Additional Plots, Functional Dataset, Histograms and ROC Curves

We display additional plots in Figures 6 and 5 for the observed functional data, the histogram of our scores showing separation between in-distribution and OOD data and the ROC curves for all of our experiments.



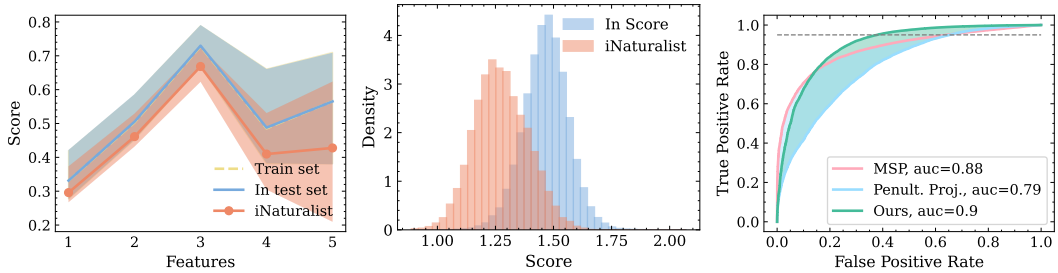
(a) Trajectories for iNaturalist. (b) Score histogram for iNaturalist. (c) ROC curve for iNaturalist.

(d) Trajectories for SUN. (e) Score histogram for SUN. (f) ROC curve for SUN.

(g) Trajectories for Places. (h) Score histogram for Places. (i) ROC curve for Places.

(j) Trajectories for Textures. (k) Score histogram for Textures. (l) ROC curve for Textures.

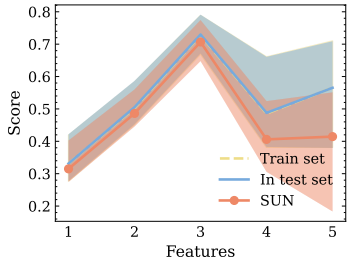
Figure 5: Average trajectories, OOD detection score histogram and ROC curve for the DenseNet-121 model.



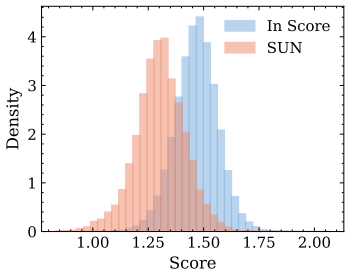
(a) Trajectories for iNaturalist.

(b) Score histogram for iNaturalist.

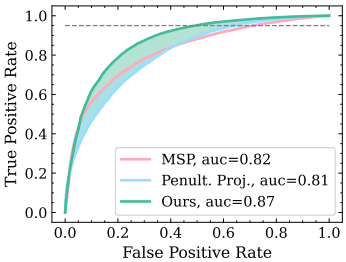
(c) ROC curve for iNaturalist.



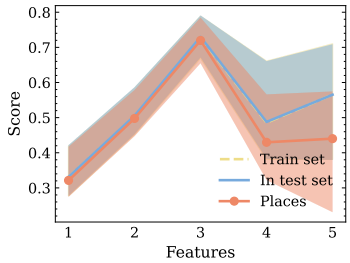
(d) Trajectories for SUN.



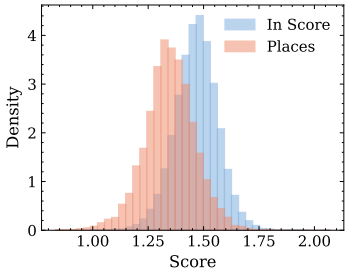
(e) Score histogram for SUN.



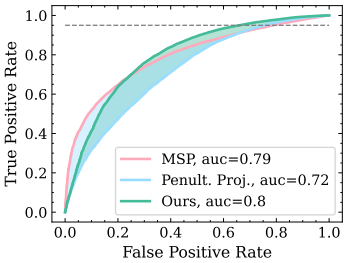
(f) ROC curve for SUN.



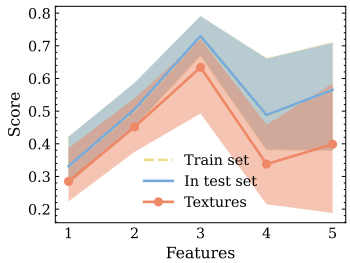
(g) Trajectories for Places.



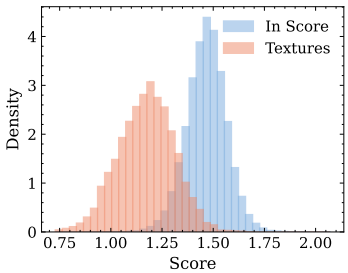
(h) Score histogram for Places.



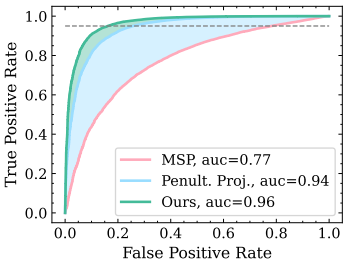
(i) ROC curve for Places.



(j) Trajectories for Textures.



(k) Score histogram for Textures.



(l) ROC curve for Textures.

Figure 6: Average trajectories, OOD detection score histogram and ROC curve for the BiT-R-101 model.