ADVANCING OUT-OF-DISTRIBUTION DETECTION VIA LOCAL NEUROPLASTICITY

Alessandro Canevaro[*1 2] **Julian Schmidt**[1] **Mohammad Sajad Marvi**[1]
Hang Yu[1] **Georg Martius**[2] **Julian Jordan**[1]
[1]Mercedes-Benz AG, Sindelfingen, Germany [2]University of Tübingen, Tübingen, Germany
[*] alessandro.canevaro@mercedes-benz.com

ABSTRACT

In the domain of machine learning, the assumption that training and test data share the same distribution is often violated in real-world scenarios, requiring effective out-of-distribution (OOD) detection. This paper presents a novel OOD detection method that leverages the unique local neuroplasticity property of Kolmogorov-Arnold Networks (KANs). Unlike traditional multilayer perceptrons, KANs exhibit local plasticity, allowing them to preserve learned information while adapting to new tasks. Our method compares the activation patterns of a trained KAN against its untrained counterpart to detect OOD samples. We validate our approach on benchmarks from image and medical domains, demonstrating superior performance and robustness compared to state-of-the-art techniques. These results underscore the potential of KANs in enhancing the reliability of machine learning systems in diverse environments.

1 INTRODUCTION

Most machine learning algorithms operate under the assumption that training and test data share the same distribution. However, this assumption frequently fails in real-world scenarios where models encounter out-of-distribution (OOD) data—samples that deviate from the training distribution, such as those belonging to novel categories. This mismatch can significantly impair a model's accuracy and reliability. As a result, OOD detection has become a critical area of research, aiming to discern when inputs fall outside the scope of the distribution used for training. Effective OOD detection not only enhances a model's robustness by identifying and handling these anomalous inputs but also ensures that the model maintains reliable performance on known, in-distribution data (Yang et al., 2022).

OOD detection poses a significant challenge due to the diverse nature of OOD types. While many OOD detectors excel when tested against specific OOD datasets, they often struggle to maintain high performance across a broad range of OOD samples. As stated by Zhang et al. (2023a) *there is no single winner that always outperforms others across multiple datasets*. One reason for this inconsistency is that OOD instances can vary widely, from subtle variations near the distribution boundary to completely dissimilar and far-off examples. As a result, developing a universal OOD detection method that performs robustly across multiple datasets, spanning near to far OOD samples, remains challenging.

In this paper, we present a novel OOD detection method using Kolmogorov-Arnold Networks (KANs) (Liu et al., 2024b). KANs are neural networks with a unique architecture that enhances interpretability, improves the accuracy-to-parameter ratio, and mitigates catastrophic forgetting compared to multilayer perceptrons (MLPs). Our approach takes advantage of KANs' distinctive property of local neuroplasticity—a characteristic absent in traditional MLPs due to their reliance on shared, non-trainable activation functions. In contrast, KANs demonstrate local plasticity owing to their spline-based architecture. This characteristic ensures that learning a new task impacts only the regions of the network activated by the training data, thereby preserving the integrity of distant and unrelated regions.

As illustrated in Figure 1, our method compares the activation patterns of two identically initialized KANs: one trained on In-Distribution (InD) data and the other left untrained. OOD samples will predominantly trigger the regions of the trained network that were not adapted during the learning phase, thus the samples will produce a response closer to the untrained network.

Conversely, InD samples will mostly engage the neurons that have been trained, resulting in a noticeable difference in the activation between the two models.

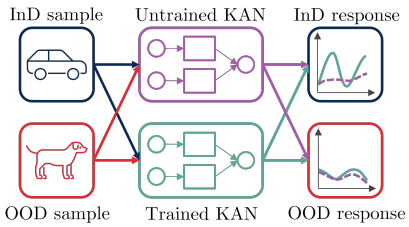

Figure 1: Overview of the proposed method: the detector compares the activation function response of a trained KAN model with its untrained counterpart. A difference in the response indicates the sample is InD, a similar response suggests it is OOD.

We tested our method on seven different benchmarks from two different domains: the OpenOOD CIFAR-10, CIFAR-100, ImageNet-200 full-spectrum (FS), and ImageNet-1K FS (Yang et al., 2022) for the image domain, and the Ethnicity, Age, and Synthetic OOD benchmarks for the tabular medical data domain (Azizmalayeri et al., 2023). Our experiments demonstrate that the KAN detector outperforms current State-Of-The-Art (SOTA) techniques across all seven benchmarks on the overall average AUROC that considers both near and far OOD. Additionally, in contrast to many other SOTA methods, our approach's performance does not vary significantly based on the number of training samples. This indicates that leveraging KANs leads to highly effective OOD detection, underscoring the potential of this novel architecture in developing more robust machine learning systems capable of operating reliably in diverse and unpredictable environments.

2 KAN-BASED OOD DETECTION

This section begins by providing a short background on KANs and their working principle. Next, we delve into the core concept underlying our proposed method for OOD detection. Finally, we describe the primary limitation of the KAN detector and propose a strategy to enable its deployment in complex real-world scenarios.

2.1 BACKGROUND

KANs are neural network architectures based on the Kolmogorov-Arnold representation theorem. This theorem states that any continuous multivariate function can be represented as a sum of continuous functions of a single variable. Hence, KANs approximate high-dimensional functions using simpler, univariate components, effectively addressing the curse of dimensionality in machine learning.

In practice, KANs decompose multivariate functions into univariate B-spline functions with learnable coefficients. Let x_p be the p-th component (feature) of the input vector $\mathbf{x} \in \mathbb{R}^{n_{\text{in}}}$ and let y_q be the q-th component (feature) of the output vector $\mathbf{y} \in \mathbb{R}^{n_{\text{out}}}$. A KAN layer transforms $\mathbf{x}$ into $\mathbf{y}$ using a matrix of univariate functions $\Phi = \{\phi_{p,q}\}$, where each $\phi_{p,q}$ is parameterized by a B-spline. Each B-spline consists of a linear combination of $G + k$ B-spline basis functions with learnable coefficients $c_{p,q,i}$. The spline order is denoted as k (usually $k = 3$) and G is the grid size.

$$y_q = \sum_p \phi_{p,q}(x_p) \quad \text{with:} \quad \phi_{p,q}(x_p) = \sum_{i=0}^{G+k} c_{p,q,i} B_i(x_p). \tag{1}$$

KAN layers can be stacked to construct deeper networks, allowing for complex transformations across multiple stages. Performance is further enhanced by incorporating residual connections, which add flexibility to the spline functions through trainable weights and additional basis functions (Liu et al., 2024b).

Local neuroplasticity in KANs is facilitated by two key properties. First, each input feature x_p is processed independently by its own set of activation functions $\{\phi_{p,q} \mid \forall q\}$. Second, during

backpropagation, only the spline coefficients near sample x_p are modified, leaving the other areas of the activation function largely unchanged.

2.2 OOD DETECTION WITH KANS

We propose leveraging the local plasticity of KANs for OOD detection. The InD data seen during training only affects specific regions (spline grid coefficients) of the network. By determining whether a region contains InD data and inspecting which regions are activated by each sample, the KAN-based detector can distinguish between InD and OOD samples. This differentiation is achieved by comparing the output of the trained activation functions with their values prior to training. The step-by-step procedure is as follows:

- **Setup**: Initialize an untrained KAN and create a copy. Train one KAN with the InD dataset while keeping the other untrained.
- **Detection**: Perform a forward pass on both networks with the given sample $\mathbf{x}$, and save the output of the activation functions:

$$\phi_{p,q}^{\text{trained}}(x_p), \quad \phi_{p,q}^{\text{untrained}}(x_p) \quad \forall p, q \tag{2}$$

Compute the difference between the responses:

$$\Delta_{p,q}(x_p) = \left|\phi_{p,q}^{\text{trained}}(x_p) - \phi_{p,q}^{\text{untrained}}(x_p)\right|. \tag{3}$$

Analyze the difference matrix Δ. OOD samples will tend to have a higher ratio of the entries in the Δ matrix close to zero. To obtain a scalar InD score $S(\mathbf{x})$, we aggregate the differences using a scoring function F_{score} (detailed in Appendix A.1):

$$S(\mathbf{x}) = F_{\text{score}}(\Delta(\mathbf{x})). \tag{4}$$

To clarify our method's working principle, let us rewrite $\Delta_{p,q}(x_p)$ using Eq. 1:

$$\Delta_{p,q}(x_p) = \sum_i \left|c_{p,q,i}^{\text{trained}} - c_{p,q,i}^{\text{untrained}}\right| \cdot B_i(x_p). \tag{5}$$

The terms $\left|c_{p,q,i}^{\text{trained}} - c_{p,q,i}^{\text{untrained}}\right|$ define the locations within the network where InD information is stored, while $B_i(x_p)$ serves as a mask and specify the regions activated by the sample $\mathbf{x}$. Consequently, multiplying these two terms provides a quantitative measure of the overlap between InD regions and the given sample. This overlap is subsequently utilized to compute the InD score.

Once the InD score is obtained, it is thresholded to classify the sample as InD or OOD:

$$D(\mathbf{x}) = \begin{cases} \text{InD}, & \text{if } S(\mathbf{x}) \geq \lambda \\ \text{OOD}, & \text{if } S(\mathbf{x}) < \lambda, \end{cases} \tag{6}$$

where λ is a predefined threshold. A test sample with a InD score less than λ is categorized as OOD. Otherwise, it is classified as InD.

Figure 2 illustrates the working principle of the proposed algorithm using a modified version of the toy example proposed by Liu et al. (2024b). The dataset is a one-dimensional regression task featuring five Gaussian peaks. We used two of these peaks as the training set and InD test set, while the remaining three peaks are the OOD test set. Here the KAN model is composed of a single layer with one input and one output, i.e. a single univariate function ϕ with 200 spline coefficients.

2.3 CAPTURING THE JOINT FEATURE DISTRIBUTION

Like MLPs, KANs are capable of processing multivariate inputs and producing multivariate outputs. However, differently from MLPs where activation functions receive a weighted sum of all inputs, in KANs, each activation function receives only a single input. While this characteristic allows the KAN detector to effectively capture the marginal distributions of input features, it also constrains its ability to model the joint distribution of features.

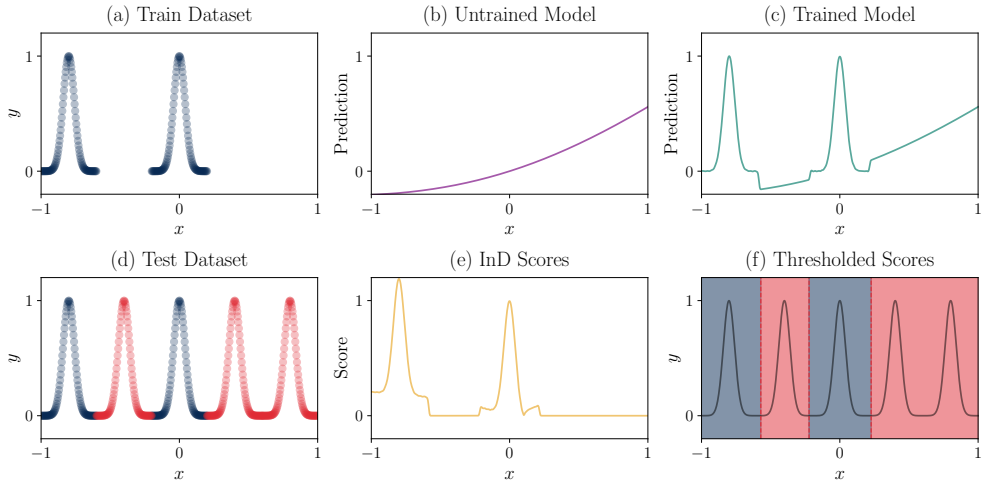

Figure 2: (a) Visualization of the training dataset, showing the relationship between inputs and targets. (b) Response of the untrained KAN model across the entire input range. (c) Response of the trained KAN model across the entire input range. (d) Test dataset illustrating inputs versus targets, created by combining the training dataset (InD) with three additional Gaussian peaks over the remaining input range (OOD). (e) InD score $S(\mathbf{x}) \forall \mathbf{x} \in [-1, 1]$ using the median as scoring function (F_{score}). (f) Final results after applying a threshold ($\lambda = 1e-3$) to the InD scores: blue regions indicate predicted InD areas and red regions indicate predicted OOD areas.

To overcome this limitation we propose to partition the InD dataset and train separate KAN models for each partition. In this way, the complex training distribution is decomposed into smaller parts that can be accurately described using only the marginal feature distribution. Various techniques can be employed to partition the dataset. A simple, yet effective approach is to split the dataset based on class labels. An alternative approach, which also works when class labels are absent, such as in regression tasks, is to apply a clustering algorithm like k-means (Lloyd, 1982). Formally, the dataset $\mathcal{D}$ is partitioned into $\mathcal{P}$ non-overlapping subsets $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_\mathcal{P}$. For each partition $\mathcal{D}_i$, we train a separate detector, denoted as KAN_i. While the partition $\mathcal{D}_i$ is different for each KAN_i, the training task is always the same (e.g., classification). During inference, we compute the InD score for a sample $\mathbf{x}$ by aggregating the InD score from each KAN model. Let $\Delta^i(\mathbf{x})$ be the difference matrix of KAN_i:

$$S(\mathbf{x}) = F_{\text{agg.}}(F_{\text{score}}(\Delta^1(\mathbf{x})), F_{\text{score}}(\Delta^2(\mathbf{x})), \ldots, F_{\text{score}}(\Delta^\mathcal{P}(\mathbf{x})), \tag{7}$$

where $F_{\text{agg.}}$ is a suitable aggregation function, such as the maximum function. Since the partitions are non-overlapping, for InD samples, there will be only one model that recognises the sample as InD (high InD score), while the other will flag it as OOD (low InD score).

Through this partitioning, our detector is now composed of multiple KAN models, and the strategy resembles ensemble methods. Furthermore, if all models are initialized with the same weights, the untrained KAN can be shared, reducing the number of forward passes at inference time.

To demonstrate the effectiveness of our proposed improvement method, we designed a specialized L-shaped dataset where the base KAN detector fails. This dataset consists of 2D points, with the training task being regression to a predefined constant. Figure 3 illustrates the results, showing the performance of the default KAN detector compared to the partitioning method.

3 Experiments

First, we describe the benchmarks, metrics, and implementation details used in our study. The results demonstrate the superior performance of our method, highlighting its key advantages. Finally, a comprehensive ablation study analyzes each hyperparameter and component, elucidating their impact on performance.

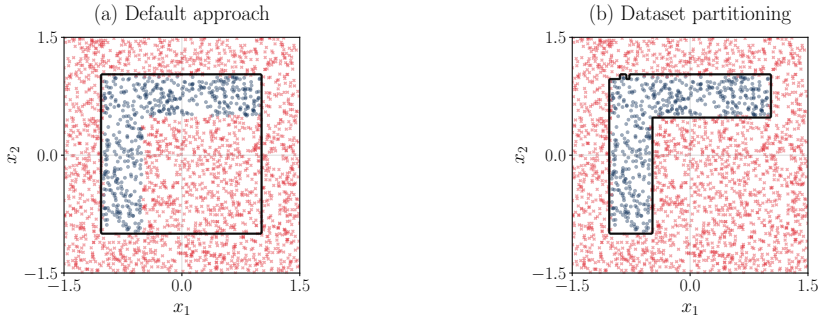

Figure 3: 2D L-shape toy dataset: The blue point cloud shows the training distribution and the red points are OOD test samples. The black contour represents the thresholded score function (median) separating InD from OOD samples. (a) Default KAN detector limited by the marginal distribution of x_1 and x_2. (b) Improved performance by partitioning the training dataset with KMeans clustering ($\mathcal{P} = 2$) and $F_{\text{agg.}} = \max$ as aggregation function.

3.1 Evaluation protocol

Setup. The evaluation of the proposed method is performed on seven different benchmarks from two different domains: OOD detection in images and tabular medical data.

For OOD detection in images, the experimental setup adheres to the OpenOOD (Yang et al., 2022) benchmark protocol. We evaluate the KAN detector on the CIFAR-10 benchmark, using CIFAR-10 (Krizhevsky et al., b) as the InD dataset. The OOD datasets are categorized into near OOD datasets (CIFAR-100 (Krizhevsky et al., a) and Tiny ImageNet (TIN) (Le & Yang, 2015)) and far OOD datasets (MNIST (Deng, 2012), SVHN (Netzer et al., 2011), Textures (Cimpoi et al., 2014), and Places365 (Zhou et al., 2018)). The CIFAR-100 benchmark contains the same datasets as the CIFAR-10 benchmark except for the CIFAR-10 and CIFAR-100 datasets which have an inverted role (CIFAR-100 as training data and CIFAR-10 as OOD dataset). To evaluate the scalability of our method, we also tested it on the ImageNet-200 FS and ImageNet-1K FS benchmarks. Compared to CIFAR-10 and CIFAR-100, these benchmarks features five to twenty times more training images, each with a size seven times larger. The full-spectrum version increases the detection challenge and, at the same time, makes it closer to real-world applications by enriching the InD test set with covariate-shifted InD samples (Yang et al., 2023). The datasets used in this benchmark are: ImageNet-200 or ImageNet-1K (Deng et al., 2009) as training set, ImageNet-V2 (Recht et al., 2019), ImageNet-C (Hendrycks & Dietterich, 2019), ImageNet-R (Hendrycks et al., 2021) as covariate-shifted InD test set, SSB-hard (Vaze et al., 2022), NINCO (Bitterwolf et al., 2023) as near OOD, and iNaturalist (Van Horn et al., 2018), Textures, OpenImage-O (Wang et al., 2022) as far OOD.

For OOD detection in tabular medical data, we follow the benchmark proposed by Azizmalayeri et al. (2023). We consider the benchmarks derived from the eICU dataset (Pollard et al., 2018), which contains clinical data of tens of thousands of Intensive Care Unit (ICU) patients in several hospitals. In the near OOD benchmarks, the eICU dataset is divided into InD and OOD according to some features such as ethnicity (Caucasian as InD) or age (older than 70 as InD). The feature used for splitting the dataset is then removed. In the synthetic OOD benchmark, the OOD data is generated by scaling a single feature from the InD set by a factor $\mathcal{F}$. For each factor, the experiment is repeated 100 times with different features, to minimize the impact of the chosen feature. By varying the scaling factor, the generated samples range from near to far OOD.

In contrast to training-time regularization methods (e.g., MOS (Huang & Li, 2021), CIDER (Ming et al., 2023)), our detector operates in a post-hoc manner and can be seamlessly integrated with any pre-trained classifier, regardless of model architecture, training procedures, or types of OOD data. The backbone is used to perform the classification or regression task and in the case of post-hoc methods it is trained independently from the detector. The OOD detector only uses the latent features of the backbone for InD/OOD classification. The considered OpenOOD benchmarks employ a pre-trained ResNet backbone (He et al., 2015) for feature extraction, while the tabular medical benchmarks use an FT-Transformer backbone (Gorishniy et al., 2021).

Given that the benchmarks we considered are all based on classification tasks and require a pre-trained backbone network, we conducted additional experiments on regression-based datasets, applying the detector directly to the data without a feature extractor. The results, presented in Appendix A.2, demonstrate that our method also performs well in these scenarios.

Metrics. In all benchmarks, the primary metric used to evaluate the OOD detection performance is the Area Under the Receiver Operating Characteristic curve (AUROC). This threshold-free metric provides a robust assessment of the model's ability to distinguish between InD and OOD samples.

In our evaluation, we focus on the average AUROC across all test datasets, including both near and far OOD (for more details on how the overall average is computed see Appendix A.3). This approach is motivated by the desire to develop a method that performs well across diverse datasets, as real-world applications often encounter unknown types of OOD samples. We acknowledge that achieving consistent performance across multiple datasets is challenging, as many methods excel on specific datasets but struggle to generalize.

Following the OpenOOD benchmark guidelines, we report the results averaged over three seeds, corresponding to three pre-trained backbones. This does not apply to the ImageNet-1K FS benchmark where only one pre-trained backbone is available. The results are averaged over five seeds for the tabular medical data benchmarks. Our approach also introduces some stochasticity due to the KAN initialization. To assess its impact on performance, we initialized the detector with five different seeds for each pre-trained backbone. The results indicate that the stochasticity due to the KAN initialization is lower than the one due to the backbone training (see Appendix A.4).

Implementation details. The detectors are trained using the InD dataset. During the evaluation phase, InD scores are calculated for all the test data. All hyperparameters are tuned using the validation set, according to the OpenOOD benchmark guidelines. The tabular medical data benchmarks follow a similar structure.

On all benchmarks, we used the median as the scoring function and the maximum as the aggregation function. The median is particularly effective due to its robustness to outliers, making it reliable for distinguishing between InD and OOD samples, as illustrated in Figure 4. These choices for both (scoring and aggregation) are further motivated in Appendix A.1.

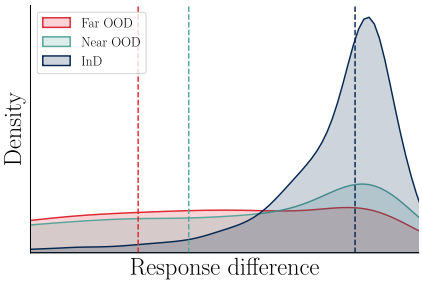

Figure 4: Distribution of activation's differences (Δ) for three different samples (InD, near and far OOD). The InD sample tends to produce bigger values in the Δ matrix compared to the OOD samples. Using the median as a scoring function (vertical dashed lines) effectively separates InD from OOD.

The latent features of the backbones exhibited a highly skewed distribution. To address this skewness and achieve a more balanced distribution that fully utilizes the KAN's grid range, we applied histogram normalization.

We leverage information from multiple latent layers of the pre-trained backbone. As demonstrated by Liu et al. (2024a), this multi-layer integration enriches the feature representation, leading to improved detection accuracy. Specifically, the authors claim that the last layer contains predominantly semantic information while including the layers closer to the input allows the detector to capture also the covariate information.

3.2 Results

Table 1 presents the results of our experiments on the CIFAR-10 and CIFAR-100 benchmarks, comparing the KAN detector with several SOTA OOD detection methods (see Appendix A.5 for a list of all the considered baselines). On top of the numerous baselines provided by the benchmark we also compare our approach to the current best post-hoc method on the CIFAR leaderboard: the NAC (Liu et al., 2024a). The results show that the KAN detector outperforms all previous methods on both benchmarks, demonstrating the effectiveness of leveraging spline-based local activation functions for OOD detection. In each column of this and the following tables, we highlight in **bold** the best-performing method. Where multiple seeds of the backbone are available we also highlight

any other methods that do not show a statistically significant difference from the best-performing method. Statistical significance is assessed using Welch's t-test with $p < 0.05$.

Table 1: Comparison of OOD detection performance (AUROC) on CIFAR-10 and CIFAR-100 benchmarks.

Method	Near OOD		Far OOD				Avg Near	Avg Far	Avg Overall
	CIFAR	TIN	MNIST	SVHN	Textures	Places365			
CIFAR-10 Benchmark									
OpenMax	86.91±0.31	88.32±0.28	90.50±0.44	89.77±0.45	89.58±0.60	88.63±0.28	87.62±0.29	89.62±0.19	88.95±0.41
ODIN	82.18±1.87	83.55±1.84	**95.24**±1.96	84.58±0.77	86.94±2.26	85.07±1.24	82.87±1.85	87.96±0.61	86.26±1.73
MDS	83.59±2.27	84.81±2.53	90.10±2.41	91.18±0.47	92.69±1.06	84.90±2.54	84.20±2.40	89.72±1.36	87.88±2.05
MDSEns	61.29±0.23	59.57±0.53	**99.17**±0.41	66.56±0.58	77.40±0.28	52.47±0.15	60.43±0.26	73.90±0.27	69.41±0.40
RMDS	88.83±0.35	90.76±0.27	93.22±0.80	91.84±0.26	92.23±0.23	**91.51**±0.11	89.80±0.28	92.20±0.21	91.40±0.40
Gram	58.33±4.49	58.98±5.19	72.64±2.34	**91.52**±4.45	62.34±8.27	60.44±3.41	58.66±4.83	71.73±3.20	67.37±5.04
ReAct	85.93±0.83	88.29±0.44	**92.81**±3.03	89.12±3.19	89.38±1.49	**90.35**±0.78	87.11±0.61	90.42±1.41	89.31±1.96
VIM	87.75±0.28	89.62±0.33	94.76±0.38	94.50±0.48	**95.15**±0.34	89.49±0.39	88.68±0.28	93.48±0.24	91.88±0.37
KNN	**89.73**±0.14	**91.56**±0.26	94.26±0.38	92.67±0.30	93.16±0.24	**91.77**±0.23	**90.64**±0.20	92.96±0.14	92.19±0.27
ASH	74.11±1.55	76.44±0.61	83.16±4.66	73.46±6.41	77.45±2.39	79.89±3.69	75.27±1.04	78.49±2.58	77.42±3.76
SHE	80.31±0.69	82.76±0.43	**90.43**±4.76	86.38±1.32	81.57±1.21	82.89±1.22	81.54±0.51	85.32±1.43	84.06±2.16
GEN	87.21±0.36	89.20±0.25	93.83±2.14	91.97±0.66	90.14±0.76	89.46±0.65	88.20±0.30	91.35±0.69	90.30±1.02
NAC	**89.83**±0.29	**92.02**±0.20	94.86±1.37	96.06±0.47	**95.64**±0.45	**91.85**±0.28	**90.93**±0.23	94.60±0.50	**93.37**±0.64
KAN	**90.06**±0.47	**91.92**±0.52	**97.86**±0.73	**97.39**±0.42	**95.85**±0.28	**91.64**±0.91	**90.99**±0.50	**95.69**±0.22	**94.12**±0.59
CIFAR-100 Benchmark									
OpenMax	74.38±0.37	78.44±0.14	76.01±1.39	82.07±1.53	80.56±0.09	79.29±0.40	76.41±0.25	79.48±0.41	78.46±0.88
ODIN	78.18±0.14	81.63±0.08	83.79±1.31	74.54±0.76	79.33±1.08	79.45±0.26	79.90±0.11	79.28±0.21	**79.49**±0.77
MDS	55.87±0.22	61.50±0.28	67.47±0.81	70.68±6.40	76.26±0.69	63.15±0.49	58.69±0.09	69.39±1.39	65.82±2.66
MDSEns	43.85±0.31	48.78±0.19	**98.21**±0.78	53.76±1.63	69.75±1.14	42.27±0.73	46.31±0.24	66.00±0.69	59.44±0.93
RMDS	77.75±0.19	82.55±0.02	79.74±2.49	84.89±1.10	83.65±0.51	**83.40**±0.46	80.15±0.11	82.92±0.42	**82.00**±1.15
Gram	49.41±0.58	53.91±1.58	80.71±4.15	**95.55**±0.60	70.79±1.32	46.38±1.21	51.66±0.77	73.36±1.08	66.12±1.98
ReAct	78.65±0.05	82.88±0.08	78.37±1.59	83.01±0.97	80.15±0.46	80.03±0.11	80.77±0.05	80.39±0.49	**80.52**±0.79
VIM	72.21±0.41	77.76±0.16	81.89±1.02	83.14±3.71	85.91±0.78	75.85±0.37	74.98±0.13	81.70±0.62	**79.46**±1.62
KNN	77.02±0.25	**83.34**±0.16	82.36±1.52	84.15±1.09	83.66±0.83	79.43±0.47	80.18±0.15	82.40±0.17	**81.66**±0.87
ASH	76.48±0.30	79.92±0.20	77.23±0.46	85.60±1.40	80.72±0.70	78.76±0.16	78.20±0.15	80.58±0.66	**79.79**±0.69
SHE	78.15±0.03	79.74±0.36	76.76±1.07	80.97±3.98	73.64±1.28	76.30±0.51	78.95±0.18	76.92±1.16	77.59±1.78
GEN	**79.38**±0.04	**83.25**±0.13	78.29±2.05	81.41±1.50	78.74±0.81	80.28±0.27	**81.31**±0.08	79.68±0.75	**80.23**±1.10
NAC	72.02±0.69	79.86±0.23	93.26±1.34	92.60±1.14	**89.36**±0.54	73.06±0.63	75.94±0.41	**87.07**±0.30	**83.36**±0.84
KAN	72.97±0.17	81.37±0.22	92.29±1.85	**87.16**±4.46	**89.43**±0.39	77.42±0.35	77.17±0.17	**86.57**±0.70	**83.44**±1.99

The KAN detector ranks first on both ImageNet-200 FS and ImageNet-1K FS benchmarks as shown in Table 2 and it consistently ranks in the top three across all tabular medical data benchmarks as reported in Tables 3, 4 and 5.

In Appendix A.6 we report the results with all the baselines available in the benchmarks for the AUROC and FPR@95 metrics.

Moreover, our method demonstrates significant robustness to variations in the number of training samples. Table 6 analyses this phenomenon on the CIFAR-10 and CIFAR-100 benchmarks, comparing the performance of the three previously best-performing methods and our approach by evaluating all methods with different dataset sizes only. Unlike other methods that achieve peak performance only with an optimal number of training samples, our approach consistently performs well across different dataset sizes. The performance of VIM and KNN is closely tied to the size of the InD dataset, while NAC achieves its best results when only 2% of the training samples are used. In contrast, the KAN detector maintains high performance across a wide range of training dataset sizes, with only a minor decrease observed in the extreme case of five samples per class. Robustness to variations in training dataset sizes is crucial in real-world scenarios where the number of training samples may be insufficient to capture the underlying distribution's characteristics. Additionally, this property is advantageous when scaling to large datasets. We attribute the strong performance of our approach across all considered benchmarks also to this key characteristic.

3.3 Ablation Study

Parameter analysis. The main hyperparameters that regulate the performance of the proposed method are the number of partitions $\mathcal{P}$ and the grid size G. Table 7 illustrates the variations in AUROC performance as a function of the number of partitions obtained through k-means clustering. Increasing $\mathcal{P}$ enhances the detector's ability to capture the joint distribution of features, resulting

Table 2: Comparison of OOD detection performance (AUROC) on ImageNet-200 FS and ImageNet-1K FS benchmarks.

Method	Near OOD		Far OOD			Avg Near	Avg Far	Avg Overall
	SSB-hard	NINCO	iNaturalist	Textures	OpenImage-O			
ImageNet-200 FS Benchmark								
OpenMax	47.64±0.20	54.15±0.23	72.44±0.87	69.12±0.36	62.31±0.24	50.89±0.18	67.96±0.39	61.13±0.46
ODIN	44.31±0.02	52.36±0.08	70.19±0.92	67.10±0.34	61.48±0.31	48.33±0.05	66.25±0.42	59.09±0.46
MDS	48.59±0.88	56.65±0.94	68.25±1.51	73.84±0.75	61.90±0.57	52.62±0.90	68.00±0.87	61.85±0.98
MDSEns	34.22±0.44	41.58±0.17	43.63±0.48	67.54±0.35	48.38±0.36	37.90±0.20	53.18±0.39	47.07±0.38
RMDS	56.24±0.62	60.95±0.94	71.71±1.49	64.61±1.07	63.52±0.83	58.59±0.77	66.62±1.11	63.41±1.03
Gram	**59.12**±0.73	**63.35**±0.76	58.42±0.75	75.86±0.10	61.51±0.39	**61.23**±0.74	65.26±0.31	63.65±0.61
ReAct	47.25±0.57	53.84±0.55	69.45±3.94	71.45±2.04	62.30±2.32	50.55±0.19	67.73±2.76	60.86±2.27
VIM	45.34±0.72	57.09±1.03	71.34±1.68	**82.54**±0.73	65.70±0.94	51.22±0.86	73.19±1.10	64.40±1.08
KNN	44.05±0.42	54.51±0.62	71.53±1.32	81.88±0.19	62.12±0.79	49.28±0.51	71.84±0.72	62.82±0.77
ASH	50.96±0.93	58.51±0.60	77.96±1.58	79.39±0.61	**69.09**±0.71	54.74±0.74	75.48±0.95	67.18±0.96
SHE	52.82±0.65	56.64±0.69	72.20±2.65	74.27±0.63	64.95±1.25	54.73±0.67	70.47±1.39	64.18±1.41
GEN	48.33±0.27	54.85±0.42	68.94±0.63	66.58±0.47	60.87±0.28	51.59±0.34	65.46±0.44	59.91±0.43
NAC	45.42±0.11	53.80±0.08	65.83±1.22	74.41±0.35	60.79±0.23	49.61±0.06	67.01±0.53	60.05±0.58
KAN	**58.37**±0.47	61.10±0.53	**84.13**±0.35	**83.30**±0.35	**70.40**±0.26	**59.74**±0.46	**79.28**±0.18	**71.46**±0.40
ImageNet-1K FS Benchmark								
OpenMax	53.79	60.28	80.30	73.54	71.88	57.03	75.24	67.96
ODIN	54.22	60.59	77.43	76.04	73.40	57.41	75.62	68.34
MDS	39.22	52.83	54.06	86.26	60.75	46.02	67.02	58.62
MDSEns	37.13	47.80	53.32	73.39	53.24	42.47	59.98	52.98
RMDS	56.61	67.50	73.48	74.25	72.13	62.06	73.29	68.79
Gram	51.93	60.63	71.36	84.83	69.40	56.28	75.20	67.63
ReAct	55.34	64.51	87.93	81.08	79.34	59.93	82.78	73.64
VIM	45.88	59.12	72.22	93.09	75.01	52.50	80.10	69.06
KNN	43.78	59.86	67.79	90.29	69.98	51.82	76.02	66.34
ASH	54.66	66.38	89.23	89.53	81.47	60.52	86.75	76.25
SHE	**58.15**	64.27	84.71	87.48	76.92	61.21	83.04	74.31
GEN	52.95	62.73	78.47	71.82	72.62	57.84	74.31	67.72
NAC	52.48	66.49	88.92	92.77	80.76	59.48	87.48	76.28
KAN	55.88	**69.55**	**91.55**	**93.45**	**82.15**	**62.71**	**89.05**	**78.52**

Table 3: Tab. Med. Caucasian Eth. as InD (AUROC metric).

Method	eICU - Eth.
MDS	**58.5**±2.2
RMDS	51.6±1.5
KNN	55.8±1.9
VIM	57.3±2.3
SHE	50.5±1.7
KLM	51.6±2.1
OpenMax	48.7±0.8
KAN	**61.4**±3.1

Table 4: Tab. Med. >70 y.o. as InD (AUROC metric).

Method	eICU - Age
MDS	**50.8**±1.1
RMDS	48.3±0.7
KNN	49.6±0.2
VIM	48.8±0.1
SHE	**50.4**±0.7
KLM	**51.0**±0.7
OpenMax	48.1±0.5
KAN	**50.5**±0.5

Table 5: Tab. Med. Feature multiplication (AUROC metric).

Method	eICU - Synthetic OOD			Avg Overall
	$\mathcal{F}=10$	$\mathcal{F}=100$	$\mathcal{F}=1000$	
MDS	59.9±1.4	79.5±1.4	87.5±0.9	75.63±1.26
RMDS	51.5±1.3	57.8±7.4	64.0±13.0	57.77±8.67
KNN	57.3±1.4	75.4±2.2	86.5±1.3	73.07±1.68
VIM	57.9±1.6	77.6±1.3	**88.3**±0.7	74.60±1.26
SHE	55.7±1.3	71.2±2.9	80.4±1.6	69.10±2.05
KLM	54.1±0.8	63.1±1.1	72.1±4.2	63.10±2.55
OpenMax	51.0±0.7	56.1±2.7	71.4±3.2	59.50±2.45
KAN	**64.6**±2.2	**83.0**±2.6	**89.8**±1.8	**79.13**±2.22

Table 6: The effect of training dataset sizes on AUROC performance.

Method	CIFAR-10 benchmark				CIFAR-100 benchmark		
	100%	10%	1%	0.1%	100%	10%	1%
VIM	91.88±0.37	91.69±0.38	88.67±1.29	76.38±3.83	79.46±1.62	78.83±1.67	67.06±2.63
KNN	92.19±0.27	91.72±0.28	88.94±0.70	8.15±0.86	81.66±0.87	80.05±0.85	27.03±1.71
NAC	87.05±1.14	89.74±0.90	93.09±0.65	89.29±0.78	80.80±0.67	81.72±0.59	80.97±1.09
KAN	**94.12**±0.59	**93.95**±0.61	**93.90**±0.62	**93.21**±0.53	**83.44**±1.99	**83.11**±2.43	**81.44**±1.21

in higher AUROC values. However, there is an upper limit beyond which further increasing the number of partitions does not lead to performance improvements. The choice of k-means clustering over other methods is justified by its simplicity and excellent scaling performance. Additionally, empirical evidence, as reported in Appendix A.7, demonstrates that the choice of the clustering algorithm does not significantly affect detection performance.

According to the authors of KAN (Liu et al., 2024b), varying the grid size has a similar effect to varying the width and depth of a traditional MLP. A fine-grained grid (higher G) should improve the accuracy of the network. In the case of our detector, as reported in Table 7, increasing the density of the grid above a certain threshold does not result in higher OOD detection performance.

Table 7: KAN detector performance *w.r.t* different datasets partitions and grid sizes over CIFAR-10.

Partitions ($\mathcal{P}$)	**AUROC**	**Grid size** (G)	**AUROC**
$\mathcal{P} = 1$	46.08±15.58	$k = 5$	87.20±1.52
$\mathcal{P} = 5$	90.39±2.78	$k = 10$	91.72±0.54
$\mathcal{P} = 10$	**94.12**±0.59	$k = 50$	93.92±0.40
$\mathcal{P} = 20$	94.10±0.62	$k = 100$	**94.12**±0.59
$\mathcal{P} = 30$	94.05±0.53	$k = 200$	94.03±0.49

Splines' smoothing operation. KANs incorporate splines that perform an essential smoothing operation, which is crucial given the continuous nature of the input space. To demonstrate the superiority of the KAN detector, we implemented a baseline histogram method by replacing all the univariate functions $\phi_{p,q}$ in KAN with simple histograms that record the presence of InD samples in a binary manner. The histogram method achieves an overall AUROC of 85.29%, which is approximately 9% lower than that of the KAN detector, clearly demonstrating the superiority of the KAN's splines approach.

Partitioning alternatives. To capture the joint feature distribution, the partitioning method is not the only solution. Another approach is to augment the input features with new features that are combinations of the original ones. This can be efficiently achieved using techniques like Principal Component Analysis (PCA) or autoencoders. PCA provides features that are linear combinations of the original ones, while autoencoders generate features that are non-linear combinations. Although this technique worked well on a toy L-shaped dataset (Appendix A.8), it did not yield the desired results on high-dimensional feature spaces in other benchmarks. It resulted in a lower AUROC compared to the partitioning method.

Influence of the training task. Since KANs are differentiable, they can be trained similarly to conventional MLPs using backpropagation. In our approach, the KAN is trained with latent features extracted from the backbone as inputs, and the training task mirrors that of the backbone network, specifically multi-class classification. For more details on the used training parameters see Appendix A.9. Importantly, the training task does not need to directly relate to the OOD detection problem. Similar to the histogram baseline, our primary objective is to *register* all input samples within the correct spline coefficients. Any training task that adjusts the spline coefficients in the vicinity of the InD samples can yield a valid OOD detector.

We experimentally verified our hypothesis by training the KAN using a different loss function and an unrelated task, namely regression to a constant value. The results from this regression task demonstrate that the detector effectively distinguishes between InD and OOD samples. Compared to the KAN trained on the classification task, we observed an improvement of approximately 0.2% in AUROC performance on the image benchmarks. However, on the tabular data benchmarks, the performance decreased by approximately 3%. These findings indicate that while modifying the training task of the detector can still yield satisfactory performance, the extent of this effect appears to be benchmark-dependent.

4 Related Work

This section reviews recent advancements in OOD detection, provides an overview of the latest innovations to enhance KAN performance, and explores the diverse sectors where KANs have demonstrated successful applications.

4.1 Out-Of-Distribution detection

OOD detection focuses on identifying instances with semantic shifts, a special case of distributional shift. OOD detection methods can be broadly classified into the following categories (Yang et al., 2024). **Classification-based methods** use the output of classification models, such as softmax

scores, to distinguish between InD and OOD samples. Examples include Maximum Softmax Probability (MSP) (Hendrycks & Gimpel, 2017), which uses the softmax score of the predicted class as a confidence score, and ODIN (Liang et al., 2018), which applies temperature scaling and input perturbations to enhance the separability of InD and OOD samples. More recent methods that fall in this category are SCALE (Xu et al., 2024a), ASH (Djurisic et al., 2023), VIM (Wang et al., 2022), and KNN (Sun et al., 2022). Gradient-based methods also belong to this category. Examples include GradNorm (Huang et al., 2021) and NAC (Liu et al., 2024a), which use gradients calculated from the KL divergence between the model's output and a uniform probability distribution. **Density-based methods** model the probability distribution of the training data to identify deviations. This is often achieved using a Gaussian mixture model (Zong et al., 2018) or normalizing flows (Zisselman & Tamar, 2020; Jiang et al., 2022). **Reconstruction-based methods** typically use autoencoders to reconstruct input samples and measure the reconstruction error as a signal for OOD detection (Jiang et al., 2023; Zhou, 2022). **Distance-based methods** rely on distance metrics in the feature space to identify OOD samples. The Mahalanobis distance-based detector (Lee et al., 2018) first models the feature distribution with a class-conditional Gaussian distribution and then it derives the InD score using the Mahalanobis distance between the InD centroids and the input sample. fDBD (Liu & Qin, 2024) measures the distance between the latent feature of the sample and the class decision boundaries. Our method also falls into this category, as it computes the InD score by measuring the distance between the network's regions activated during training (InD regions) and those activated by the test sample.

4.2 Kolmogorov-Arnold Networks

The recently introduced KAN (Hou & Zhang, 2024) represents a significant advancement in neural network architectures, offering a potential alternative to traditional MLPs by not only enhancing accuracy but also leading to more interpretable models. As a result, numerous studies have tried to innovate and refine KANs further. For example, many articles replace the spline architecture with more efficient or accurate alternatives such as Chebyshev polynomials (SS et al., 2024), wavelet-based structures (Bozorgasl & Chen, 2024), sinusoidal functions (Reinhardt et al., 2024), and radial basis functions (Li, 2024). Others try to replicate advanced neural network architectures using KAN's characteristics. This includes convolutional neural networks (Bodner et al., 2024) and graph neural networks (Kiamari et al., 2024; Bresson et al., 2024; Zhang & Zhang, 2024), further demonstrating the versatility and potential of KANs. Applications of KANs have rapidly expanded across various domains, including time series analysis (Vaca-Rubio et al., 2024; Xu et al., 2024b), solving ordinary and partial differential equations (Koenig et al., 2024; Wang et al., 2024), hyperspectral image classification (Seydi, 2024; Jamali et al., 2024), and computer vision (Azam & Akhtar, 2024; Li et al., 2024; Cheon, 2024). Additionally, KANs have recently been applied to fields similar to OOD detection, such as abnormality detection (Huang et al., 2024) and AI-generated image detection (Anon & Emon, 2024). These studies leverage the superior accuracy and interpretability of KANs (Liu et al., 2024b) to uncover more complex patterns in the data. While their work focuses on developing robust models that demonstrate KANs' capacity to generalize effectively to unseen samples, they do not address the detection of these samples. In contrast, we present a novel OOD detection method that leverages the unique local plasticity property of KANs, applicable to any backbone architecture.

5 Conclusions

This paper introduces a novel approach to OOD detection using KANs, capitalizing on their unique local neuroplasticity property. Our method effectively differentiates between InD and OOD samples by comparing the activation patterns of a trained KAN against its untrained counterpart. The experimental results show that our KAN-based detector reaches SOTA performance across seven benchmarks from two different domains. Importantly, our experiments show that the previous methods suffer from a non-optimal InD dataset size, while our method is unaffected by these perturbations. This makes the KAN detector a robust and versatile method that can maintain high performance across diverse and unpredictable data distributions. Future work will further explore the effect of different training tasks on detection performance.

ACKNOWLEDGMENTS

This work is a result of the joint research project STADT:up (19A22006O). The project is supported by the German Federal Ministry for Economic Affairs and Climate Action (BMWK), based on a decision of the German Bundestag. The authors are solely responsible for the content of this publication.

The authors would like to express their sincere gratitude to Pavel Kolev, Niklas Hanselmann, and Shuxiao Ding for their invaluable assistance in proofreading and providing insightful feedback on this manuscript.

REPRODUCIBILITY STATEMENT

Our implementation adheres rigorously to the benchmark guidelines. Detailed information on the hardware and software utilized is provided in Appendix A.10, while the inference time performance of our method is discussed in Appendix A.11. The settings and hyperparameters for each benchmark are reported in Appendix A.12. Our code is publicly available at the following link: `https://github.com/alessandro-canevaro/KAN-OOD`.

REFERENCES

Taharim Rahman Anon and Jakaria Islam Emon. Detecting the undetectable: Combining kolmogorov-arnold networks and mlp for ai-generated image detection, 2024. URL https://arxiv.org/abs/2408.09371.

Basim Azam and Naveed Akhtar. Suitability of kans for computer vision: A preliminary investigation, 2024. URL https://arxiv.org/abs/2406.09087.

Mohammad Azizmalayeri, Ameen Abu-Hanna, and Giovanni Cina. Unmasking the chameleons: A benchmark for out-of-distribution detection in medical tabular data. *arXiv preprint arXiv:2309.16220*, 2023.

Abhijit Bendale and Terrance E. Boult. Towards open set deep networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1563–1572, 2016. doi: 10.1109/CVPR.2016.173.

Julian Bitterwolf, Maximilian Müller, and Matthias Hein. In or out? fixing imagenet out-of-distribution detection evaluation, 2023. URL https://arxiv.org/abs/2306.00826.

Alexander Dylan Bodner, Antonio Santiago Tepsich, Jack Natan Spolski, and Santiago Pourteau. Convolutional kolmogorov-arnold networks, 2024. URL https://arxiv.org/abs/2406.13155.

Zavareh Bozorgasl and Hao Chen. Wav-kan: Wavelet kolmogorov-arnold networks, 2024. URL https://arxiv.org/abs/2405.12832.

Roman Bresson, Giannis Nikolentzos, George Panagopoulos, Michail Chatzianastasis, Jun Pang, and Michalis Vazirgiannis. Kagnns: Kolmogorov-arnold networks meet graph learning, 2024. URL https://arxiv.org/abs/2406.18380.

Minjong Cheon. Kolmogorov-arnold network for satellite image classification in remote sensing, 2024. URL https://arxiv.org/abs/2406.00600.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

Paulo Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis. Wine Quality. UCI Machine Learning Repository, 2009. DOI: https://doi.org/10.24432/C56S3T.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. doi: 10.1109/MSP.2012.2211477.

Andrija Djurisic, Nebojsa Bozanic, Arjun Ashok, and Rosanne Liu. Extremely simple activation shaping for out-of-distribution detection. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=ndYXTEL6cZz.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pp. 226–231. AAAI Press, 1996.

Jerome H. Friedman. Multivariate Adaptive Regression Splines. *The Annals of Statistics*, 19(1): 1 – 67, 1991. doi: 10.1214/aos/1176347963. URL https://doi.org/10.1214/aos/1176347963.

Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=i_Q1yrOegLY.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 1321–1330. JMLR.org, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. URL https://arxiv.org/abs/1512.03385.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 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. URL https://openreview.net/forum?id=Hkg4TI9xl.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. *ICCV*, 2021.

Dan Hendrycks, Steven Basart, Mantas Mazeika, Andy Zou, Joe Kwon, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. Scaling out-of-distribution detection for real-world settings. *ICML*, 2022.

Yuntian Hou and Di Zhang. A comprehensive survey on kolmogorov arnold networks (kan), 2024. URL https://arxiv.org/abs/2407.11075.

Rui Huang and Yixuan Li. Mos: Towards scaling out-of-distribution detection for large semantic space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

Rui Huang, Andrew Geng, and Yixuan Li. On the importance of gradients for detecting distributional shifts in the wild. In *Advances in Neural Information Processing Systems*, 2021.

Zhaojing Huang, Jiashuo Cui, Leping Yu, Luis Fernando Herbozo Contreras, and Omid Kavehei. Abnormality detection in time-series bio-signals using kolmogorov-arnold networks for resource-constrained devices. *medRxiv*, 2024. doi: 10.1101/2024.06.04.24308428. URL https://www.medrxiv.org/content/early/2024/06/04/2024.06.04.24308428.

Ali Jamali, Swalpa Kumar Roy, Danfeng Hong, Bing Lu, and Pedram Ghamisi. How to learn more? exploring kolmogorov-arnold networks for hyperspectral image classification, 2024. URL https://arxiv.org/abs/2406.15719.

Dihong Jiang, Sun Sun, and Yaoliang Yu. Revisiting flow generative models for out-of-distribution detection. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=6y2KBh-0Fd9.

Wenyu Jiang, Yuxin Ge, Hao Cheng, Mingcai Chen, Shuai Feng, and Chongjun Wang. Read: Aggregating reconstruction error into out-of-distribution detection, 2023. URL https://arxiv.org/abs/2206.07459.

R. Kelley Pace and Ronald Barry. Sparse spatial autoregressions. *Statistics And Probability Letters*, 33(3):291–297, 1997. ISSN 0167-7152. doi: https://doi.org/10.1016/S0167-7152(96)00140-X. URL https://www.sciencedirect.com/science/article/pii/S016771529600140X.

Mehrdad Kiamari, Mohammad Kiamari, and Bhaskar Krishnamachari. Gkan: Graph kolmogorov-arnold networks, 2024. URL https://arxiv.org/abs/2406.06470.

Benjamin C. Koenig, Suyong Kim, and Sili Deng. Kan-odes: Kolmogorov-arnold network ordinary differential equations for learning dynamical systems and hidden physics, 2024. URL https://arxiv.org/abs/2407.04192.

S. Kong and D. Ramanan. Opengan: Open-set recognition via open data generation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 793–802, Los Alamitos, CA, USA, oct 2021. IEEE Computer Society. doi: 10.1109/ICCV48922.2021.00085. URL https://doi.ieeecomputersociety.org/10.1109/ICCV48922.2021.00085.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-100 (canadian institute for advanced research). a. URL http://www.cs.toronto.edu/~kriz/cifar.html.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). b. URL http://www.cs.toronto.edu/~kriz/cifar.html.

Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

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*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/abdeb6f575ac5c6676b747bca8d09cc2-Paper.pdf.

Chenxin Li, Xinyu Liu, Wuyang Li, Cheng Wang, Hengyu Liu, Yifan Liu, Zhen Chen, and Yixuan Yuan. U-kan makes strong backbone for medical image segmentation and generation, 2024. URL https://arxiv.org/abs/2406.02918.

Ziyao Li. Kolmogorov-arnold networks are radial basis function networks, 2024. URL https://arxiv.org/abs/2405.06721.

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.

Dong C. Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45(1):503–528, Aug 1989. ISSN 1436-4646. doi: 10.1007/BF01589116. URL https://doi.org/10.1007/BF01589116.

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

Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 21464–21475. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/f5496252609c43eb8a3d147ab9b9c006-Paper.pdf.

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

Yibing Liu, Chris XING TIAN, Haoliang Li, Lei Ma, and Shiqi Wang. Neuron activation coverage: Rethinking out-of-distribution detection and generalization. In *The Twelfth International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=SNGXbZtK6Q.

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y. Hou, and Max Tegmark. Kan: Kolmogorov-arnold networks, 2024b. URL https://arxiv.org/abs/2404.19756.

S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2): 129–137, 1982. doi: 10.1109/TIT.1982.1056489.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Yifei Ming, Yiyou Sun, Ousmane Dia, and Yixuan Li. How to exploit hyperspherical embeddings for out-of-distribution detection? In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=aEFaE0W5pAd.

Fionn Murtagh and Pierre Legendre. Ward's hierarchical agglomerative clustering method: Which algorithms implement ward's criterion? *Journal of Classification*, 31(3):274–295, Oct 2014. ISSN 1432-1343. doi: 10.1007/s00357-014-9161-z. URL https://doi.org/10.1007/s00357-014-9161-z.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. URL http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf.

Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In T. Dietterich, S. Becker, and Z. Ghahramani (eds.), *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2001. URL https://proceedings.neurips.cc/paper_files/paper/2001/file/801272ee79cfde7fa5960571fee36b9b-Paper.pdf.

Tom J. Pollard, Alistair E. W. Johnson, Jesse D. Raffa, Leo A. Celi, Roger G. Mark, and Omar Badawi. The eicu collaborative research database, a freely available multi-center database for critical care research. *Scientific Data*, 5(1):180178, Sep 2018. ISSN 2052-4463. doi: 10.1038/sdata.2018.178. URL https://doi.org/10.1038/sdata.2018.178.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5389–5400. PMLR, 2019. URL http://dblp.uni-trier.de/db/conf/icml/icml2019.html#RechtRSS19.

Eric A. F. Reinhardt, P. R. Dinesh, and Sergei Gleyzer. Sinekan: Kolmogorov-arnold networks using sinusoidal activation functions, 2024. URL https://arxiv.org/abs/2407.04149.

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. URL https://arxiv.org/abs/2106.09022.

Vinita Rohilla, Ms Sanika Singh kumar, Sudeshna Chakraborty, and Ms. Sanika Singh. Data clustering using bisecting k-means. In *2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, pp. 80–83, 2019. doi: 10.1109/ICCCIS48478.2019.8974537.

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.

Seyd Teymoor Seydi. Unveiling the power of wavelets: A wavelet-based kolmogorov-arnold network for hyperspectral image classification, 2024. URL https://arxiv.org/abs/2406.07869.

Yue Song, Nicu Sebe, and Wei Wang. Rankfeat: Rank-1 feature removal for out-of-distribution detection. In *NeurIPS*, 2022.

Sidharth SS, Keerthana AR, Gokul R, and Anas KP. Chebyshev polynomial-based kolmogorov-arnold networks: An efficient architecture for nonlinear function approximation, 2024. URL https://arxiv.org/abs/2405.07200.

Yiyou Sun and Yixuan Li. Dice: Leveraging sparsification for out-of-distribution detection. In *European Conference on Computer Vision*, 2022.

Yiyou Sun, Chuan Guo, and Yixuan Li. React: Out-of-distribution detection with rectified activations. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=IBVBtz_sRSm.

Yiyou Sun, Yifei Ming, Xiaojin Zhu, and Yixuan Li. Out-of-distribution detection with deep nearest neighbors. *ICML*, 2022.

Cristian J. Vaca-Rubio, Luis Blanco, Roberto Pereira, and Màrius Caus. Kolmogorov-arnold networks (kans) for time series analysis, 2024. URL https://arxiv.org/abs/2405.08790.

Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The iNaturalist Species Classification and Detection Dataset . In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8769–8778, Los Alamitos, CA, USA, June 2018. IEEE Computer Society. doi: 10.1109/CVPR.2018.00914. URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2018.00914.

Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Open-set recognition: A good closed-set classifier is all you need. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=5hLP5JY9S2d.

Haoqi Wang, Zhizhong Li, Litong Feng, and Wayne Zhang. Vim: Out-of-distribution with virtual-logit matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Yizheng Wang, Jia Sun, Jinshuai Bai, Cosmin Anitescu, Mohammad Sadegh Eshaghi, Xiaoying Zhuang, Timon Rabczuk, and Yinghua Liu. Kolmogorov arnold informed neural network: A physics-informed deep learning framework for solving forward and inverse problems based on kolmogorov arnold networks, 2024. URL https://arxiv.org/abs/2406.11045.

Kai Xu, Rongyu Chen, Gianni Franchi, and Angela Yao. Scaling for training time and post-hoc out-of-distribution detection enhancement. In *The Twelfth International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=RDSTjtnqCg.

Kunpeng Xu, Lifei Chen, and Shengrui Wang. Kolmogorov-arnold networks for time series: Bridging predictive power and interpretability, 2024b. URL https://arxiv.org/abs/2406.02496.

Jingkang Yang, Pengyun Wang, Dejian Zou, Zitang Zhou, Kunyuan Ding, WENXUAN PENG, Haoqi Wang, Guangyao Chen, Bo Li, Yiyou Sun, Xuefeng Du, Kaiyang Zhou, Wayne Zhang, Dan Hendrycks, Yixuan Li, and Ziwei Liu. Openood: Benchmarking generalized out-of-distribution detection. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 32598–32611. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/d201587e3a84fc4761eadc743e9b3f35-Paper-Datasets_and_Benchmarks.pdf.

Jingkang Yang, Kaiyang Zhou, and Ziwei Liu. Full-spectrum out-of-distribution detection. *International Journal of Computer Vision*, 131(10):2607–2622, Oct 2023. ISSN 1573-1405. doi: 10.1007/s11263-023-01811-z. URL https://doi.org/10.1007/s11263-023-01811-z.

Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey, 2024. URL https://arxiv.org/abs/2110.11334.

Fan Zhang and Xin Zhang. Graphkan: Enhancing feature extraction with graph kolmogorov arnold networks, 2024. URL https://arxiv.org/abs/2406.13597.

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

Jinsong Zhang, Qiang Fu, Xu Chen, Lun Du, Zelin Li, Gang Wang, xiaoguang Liu, Shi Han, and Dongmei Zhang. Out-of-distribution detection based on in-distribution data patterns memorization with modern hopfield energy. In *The Eleventh International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=KkazG4lgKL.

Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Birch: an efficient data clustering method for very large databases. *SIGMOD Rec.*, 25(2):103–114, June 1996. ISSN 0163-5808. doi: 10.1145/235968.233324. URL https://doi.org/10.1145/235968.233324.

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*, 40(6):1452–1464, 2018. doi: 10.1109/TPAMI.2017.2723009.

Yibo Zhou. Rethinking reconstruction autoencoder-based out-of-distribution detection. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7369–7377, 2022. doi: 10.1109/CVPR52688.2022.00723.

Ev Zisselman and Aviv Tamar. Deep residual flow for out of distribution detection, 2020. URL https://arxiv.org/abs/2001.05419.

Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=BJJLHbb0-.

A APPENDIX

A.1 SCORING AND AGGREGATION FUNCTIONS

The trainable coefficients of the detector networks are initialized randomly. As a result, it may occasionally occur that some of these coefficients are initialized to the exact values they would attain post-training. Consequently, the training procedure does not modify these coefficients, as they are already optimal. This phenomenon can lead to false positives in detection, as both the trained and untrained networks might exhibit the same response to an InD sample. This issue can be mitigated using a scoring function that is more robust to outliers, such as the median. This hypothesis is experimentally validated in Table 8.

The use of the maximum function for aggregation allows us to select the detector closest to the test sample, which intuitively possesses the best information for decision-making. This approach is experimentally verified in Table 9.

Table 8: AUROC performance variation on the CIFAR-10 benchmark for different scoring F_{score} functions.

Scoring F_{score}	**AUROC**
min	90.84±0.35
mean	92.04±0.37
median	**94.12**±0.59
max	54.99±12.02

Table 9: AUROC performance variation on the CIFAR-10 benchmark for different aggregation $F_{\text{agg.}}$ functions.

Aggregation $F_{\text{agg.}}$	**AUROC**
min	91.55±0.97
mean	91.99±2.30
median	90.31±3.28
max	**94.12**±0.59

A.2 DETECTION ON REGRESSION-BASED DATASETS

As the standard benchmarks used in OOD detection mainly focus on classification tasks, we test our method on three additional regression-based datasets: the California Housing dataset (Kelley Pace & Barry, 1997), the Wine Quality dataset (Cortez et al., 2009), and the Friedman synthetic dataset (Friedman, 1991). To generate the InD and OOD partitions and ensure that the OOD samples are semantically different from the InD ones, we thresholded the regression (output) value. The KAN is then directly applied to the raw dataset features, highlighting that the method is not only effective on regression-based tasks but also in the absence of a feature extractor backbone network. This is not possible for other methods such as NAC that require the gradients of the backbone network for detection. Thus, as a baseline detector method, we used KNN, which, according to the results in Section 3.2, is one of the best approaches across all benchmarks. Table 10 reports the detection results in terms of AUROC on the three datasets, showing that the KAN detector outperforms the KNN baseline on all of them.

Table 10: Detection (AUROC) results for regression-based datasets.

Method	**California Housing**	**Wine Quality**	**Friedman**
KNN	68.73	68.69	67.30
KAN	**70.53**	**71.32**	**69.42**

On the California Housing and Wine Quality datasets, we used only one partition ($\mathcal{P}$) for the KAN detector because a value greater than one did not improve performance. This indicates that either the partitioning method does not work well on regression-based datasets, possibly due to poor internal separability of data clusters, or these datasets do not require the detector to capture the joint feature distribution to effectively separate InD and OOD samples. To investigate this observation further, we also tested our method on the Friedman dataset. Here, the regression output is generated by the following non-linear function of the inputs:

$$y(\mathbf{x}) = 10 \cdot \sin(\pi \cdot x_0 \cdot x_1) + 20 \cdot (x_2 - 0.5)^2 + 10 \cdot x_3 + 5 \cdot x_4 + \mathcal{N}(0, \sigma). \tag{8}$$

Given this non-linearity, it is ensured that the samples are not separable using only the marginal feature distribution. In this case, peak performance is achieved with a minimum of four partitions, as shown in Table 11. This shows that even for regression-based datasets our method can capture the joint feature distribution.

Table 11: AUROC performance as a function of the number of partitions ($\mathcal{P}$) in the Friedman dataset.

Partitions ($\mathcal{P}$)	**1**	**2**	**3**	**4**
AUROC	52.11	64.15	63.83	**69.42**

A.3 Average overall metric

Many benchmarks (including the OpenOOD CIFAR-10 and CIFAR-100) assess OOD detection performance on multiple OOD datasets. However, they lack an overall average that gives a holistic overview of the methods' performance. In our experiments, we additionally evaluate our method on the following *overall* metric:

$$\mu_{\text{overall}} = \frac{1}{N}\sum_{i=1}^{N}\mu_i, \quad \sigma_{\text{overall}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\sigma_i^2} \tag{9}$$

where μ_{i}, σ_{i} are the mean and standard deviation of dataset i calculated over multiple seeds.

A.4 Effect of KAN stochasticity

All benchmarks average results over multiple seeds to address the inherent randomness associated with weight initialization in the backbone model. Our method introduces an additional layer of randomness due to the KAN initialization process. To illustrate that the variability introduced by our detector is significantly lower than that stemming from the backbone initialization, we conducted the following experiment.

We repeated the CIFAR-10 benchmark using five distinct KAN initialization seeds ($N = 5$). For each KAN initialization seed i, we recorded the mean and standard deviation (μ_i, σ_i) of the experiment conducted on the three pre-trained backbones specified in the benchmark. The results are summarized in Table 12.

Table 12: CIFAR-10 benchmark results across different KAN initializations.

KAN seed (i)	**1**	**2**	**3**	**4**	**5**
AUROC ($\mu_i \pm \sigma_i$)	94.12$\pm$0.59	94.02$\pm$0.58	94.11$\pm$0.52	94.17$\pm$0.57	94.06$\pm$0.39

We compute the overall standard deviation attributable to the backbone initialization (σ_{b}) and that of our detector (σ_{d}) as follows:

$$\sigma_{\text{b}} = \frac{1}{N}\sum_{i=1}^{N}\sigma_i = 0.53, \quad \sigma_{\text{d}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\sigma_i - \mu_{\text{b}})^2} = 0.05 \text{ with } \mu_{\text{b}} = \frac{1}{N}\sum_{i=1}^{N}\mu_i = 94.10 \tag{10}$$

The calculated standard deviations σ_{b} and σ_{d} differ by approximately an order of magnitude, indicating that the randomness introduced by our detector has a negligible effect on the overall results.

A.5 Baselines methods

The baselines used in the banchmarks are: OpenMax (Bendale & Boult, 2016), MSP (Hendrycks & Gimpel, 2017), TempScale (Guo et al., 2017), ODIN (Liang et al., 2018), MDS (Lee et al., 2018),

MDSEns (Lee et al., 2018), RMDS (Ren et al., 2021), Gram (Sastry & Oore, 2020), EBO (Liu et al., 2020), OpenGAN (Kong & Ramanan, 2021), GradNorm (Huang et al., 2021), ReAct (Sun et al., 2021), MLS (Hendrycks et al., 2022), KLM (Hendrycks et al., 2022), VIM (Wang et al., 2022), KNN (Sun et al., 2022), DICE (Sun & Li, 2022), RankFeat (Song et al., 2022), ASH (Djurisic et al., 2023), SHE (Zhang et al., 2023b), GEN (Liu et al., 2023), and NAC (Liu et al., 2024a).

A.6 Full benchmark results

In Table 17, we present the AUROC results for all officially available baselines on the CIFAR-10 and CIFAR-100 benchmarks. Table 22 provides the results for the same set of baselines and benchmarks using the FPR@95 metric. Similarly, Tables 18 and 23 report the AUROC and FPR@95 results respectively for all the available baselines on the ImageNet-200 FS and ImageNet-1K FS benchmarks. The same metrics are reported for all the available baselines for the tabular medical benchmarks in Tables 19, 20, 21 for AUROC and 24, 25, 26 for FPR@95.

A.7 Influence of clustering method

This experiment analyzes the effect of different clustering methods on detection performance. We considered five popular clustering approaches as alternatives to k-means: spectral (Ng et al., 2001), agglomerative (Murtagh & Legendre, 2014), bisecting k-means (Rohilla et al., 2019), BIRCH (Zhang et al., 1996), and DBSCAN (Ester et al., 1996). Table 13 presents the experimental results for the CIFAR-10 benchmark.

Table 13: Detection performance with different clustering algorithms on the CIFAR-10 benchmark.

	k-means	**spectral**	**agglomerative**	**bisecting k-means**	**BIRCH**	**DBSCAN**
AUROC	94.12±0.59	94.11±0.59	94.12±0.59	94.10±0.57	94.12±0.59	89.82±4.64

The results show that the choice of clustering method has a negligible impact on detection performance, except for DBSCAN, which yields an approximate 4% drop. One reason for this behavior is that DBSCAN is the only algorithm among those considered that does not necessarily assign a cluster to all samples. In our implementation, these unclustered samples are grouped into an additional cluster. However, the samples in this extra cluster do not share common characteristics and can belong to different and distant regions of the input space. Although we are not focused on obtaining semantically meaningful clusters, we aim to divide the InD samples into smaller regions that can be effectively processed by a KAN. The leftover samples cluster in DBSCAN has a counterproductive effect, as it can span a wide region of the input space, making it difficult for the KAN to handle effectively.

A.8 Capturing the joint feature distribution

An alternative approach to the partitioning method for capturing the joint feature distribution is to expand the input features with additional values. We applied this technique to the 2D L-shaped toy dataset, as illustrated in Figure 5. In this scenario, the two input features are concatenated with the latent features derived from a variational autoencoder trained on the two original features, resulting in an augmented input space of size $2 + 64$.

This method demonstrates promising results, comparable to those achieved with the partitioning method. However, its applicability to high-dimensional input spaces remains uncertain. We hypothesize that the number of required features would become excessively large, leading to computational inefficiencies.

A.9 Training parameters

Finding the optimal training hyperparameters for KANs can initially be challenging, as they may not follow the same intuitions as MLPs and other networks (Liu et al., 2024b).

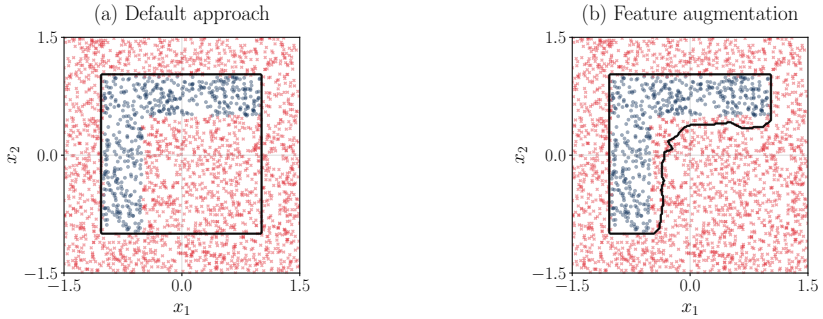

Figure 5: 2D L-shape toy dataset: The blue point cloud shows the training distribution and the red points are OOD test samples. The black contour represents the thresholded score function (median) separating InD from OOD samples. (a) Default KAN detector limited by the marginal distribution of x_1 and x_2. (b) Improved performance by concatenating the input features with the latent features of a variational autoencoder.

In our experiments, we use a learning rate of 0.1 and limit the training to a single epoch. With these settings, the trained KAN achieved a classification accuracy on the CIFAR-10 dataset of approximately 94.50%, comparable to the one of the ResNet-18 of 95.06%.

To enhance memory efficiency, we employed the AdamW optimizer (Loshchilov & Hutter, 2019) instead of the LBFGS optimizer (Liu & Nocedal, 1989) originally suggested by the KAN authors.

A.10 Software and hardware

All experiments are performed on a single NVIDIA GeForce GTX 1080Ti GPU. For testing larger models and accelerating the hyperparameter optimization, we leveraged a cloud computing platform with an NVIDIA A100 GPU.

We used Python version 3.10 together with PyTorch 2.3.1 as the deep learning framework and leveraged CUDA 11.8 for GPU acceleration.

A.11 Inference time and scalability analysis

Inference time. Table 14 reports the inference time of a single sample for various methods. The measurements are averaged over 1000 samples, using 100 extra samples as GPU warmup. The results show a positive correlation between the inference time and the overall AUROC performance.

Although the KAN method is currently the slowest among the tested ones, it is important to emphasize that the KAN architecture has just been developed recently. In just a few months since its release, its performance has been steadily improving thanks to many architecture refinements (e.g., replacing splines with Gaussian radial basis functions improves forward speed by approximately a factor of 3.3 (Li, 2024)). We believe that in the future, KANs will achieve efficiency comparable to MLPs. Furthermore, it is worth considering that inference time is not always a critical concern in various applications, particularly in medical contexts. In such scenarios, the enhanced detection performance offered by our method positions it as a more advantageous choice compared to faster alternatives.

Table 14: Inference time of single sample compared to the overall AUROC on CIFAR10.

Method	Inference time (ms)	Overall AUROC
VIM	0.271	91.88
KNN	0.175	92.19
NAC	0.681	93.37
KAN	2.216	94.12

Setup time. In Table 15, we report the setup time of our detector for different training dataset sizes and various numbers of partitions ($\mathcal{P}$), using the KNN method as a reference baseline.

Table 15: Setup time in seconds for different training dataset sizes and number of partitions.

Method	10K	100K	1M
KAN - $\mathcal{P} = 1$	2.84	17.49	165.86
KAN - $\mathcal{P} = 10$	2.82	17.44	166.45
KAN - $\mathcal{P} = 100$	3.19	18.24	172.66
KNN	3.27	15.51	141.75

The results indicate that the biggest factor influencing setup time (which includes inference on the backbone model, the partitioning method and the training of the KANs) is the dataset size; however, our method shows comparable speeds to the KNN baseline. On the other hand, varying the number of partitions seems to have a smaller influence, likely due to GPU parallelization.

Further considerations. The number of parameters in the KAN network is determined by the product of three factors: the number of inputs, the number of outputs, and the grid size. The grid size depends on the specific benchmark, and our experiments indicate that it does not correlate with the benchmark's complexity. For instance, on CIFAR-10 the optimal value is 100, while on CIFAR-100 it is 50. Scalability to larger images, or more generally to large input spaces, is typically not an issue as our detector is applied to the latent space of the backbone model, which is usually much smaller than the inputs. For example, ImageNet-200 has an input space of roughly 150k dimensions, but it is compressed by the backbone into a latent space of just 512 features. Lastly, our preliminary results presented in Section 3.3 demonstrate that changing the training task of the KAN detector can lead to similar performance. This can can be used to reduce the number of outputs required by the KAN model, further improving scalability. For example, in the ImageNet-1K FS benchmark, we employed class-based partitioning, resulting in 1000 clusters and, consequently, 1000 models. However, we reduced the number of outputs for each model from 1000 to 10 classes by randomly grouping labels together. This adjustment is motivated by the fact that with 1000 clusters, the problem tackled by each model is greatly reduced, and thus the model's capacity can also be reduced. As a result, the training time per sample of our model is slightly lower than that for the ImageNet-200 FS benchmark: approximately 1.6ms per sample compared to 1.9ms per sample for ImageNet-200 FS. This indicates that our method remains efficient and robust even with a large number of clusters.

A.12 HYPERPARAMETERS

Table 16 reports all hyperparameters and settings for the five benchmarks. The search space of each hyperparameter is as follows: [10, 200] for the grid size, [1, 200] for the partitions, [0.0001, 0.1] for the learning rate, [1, 100] for the epochs, and [1, 100] for the histogram bins.

Table 16: Hyperparameters.

Parameter	OpenOOD				TabMed		
	CIFAR10	CIFAR100	ImageNet-200 FS	ImageNet-1K FS	Ethnicity	Age	Synthetic
Grid size	100	50	10	50	100	50	50
Partitions	10	100	200	1000	30	30	10
Train Size	100%	100%	100%	100%	100%	100%	100%
Learning rate	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Epochs	1	1	1	1	1	1	1
Partitioning method	k-means	class-based	class-based	class-based	k-means	k-means	k-means
Histogram norm.	yes	yes	yes	yes	yes	no	no
Histogram bins	5	10	5	5	40	-	-
Backbone	ResNet-18	ResNet-18	ResNet-18	ResNet-50	FT-Transformer	FT-Transformer	FT-Transformer
Pre-Trained backbone	yes	yes	yes	yes	no	no	no

Table 17: AUROC performance on CIFAR-10 and CIFAR-100 benchmarks.

Method	Near OOD		Far OOD				Avg Near	Avg Far	Avg Overall
	CIFAR	TIN	MNIST	SVHN	Textures	Places365			
CIFAR-10 Benchmark									
OpenMax	86.91±0.31	88.32±0.28	90.50±0.44	89.77±0.45	89.58±0.60	88.63±0.28	87.62±0.29	89.62±0.19	88.95±0.41
MSP	87.19±0.33	88.87±0.19	92.63±1.57	91.46±0.40	89.89±0.71	88.92±0.47	88.03±0.25	90.73±0.43	89.83±0.76
TempScale	87.17±0.40	89.00±0.23	93.11±1.77	91.66±0.52	90.01±0.74	89.11±0.52	88.09±0.31	90.97±0.52	90.01±0.86
ODIN	82.18±1.87	83.55±1.84	**95.24**±1.96	84.58±0.77	86.94±2.26	85.07±1.24	82.87±1.85	87.96±0.61	86.26±1.73
MDS	83.59±2.27	84.81±2.53	90.10±2.41	91.18±0.47	92.69±1.06	84.90±2.54	84.20±2.40	89.72±1.36	87.88±2.05
MDSEns	61.29±0.23	59.57±0.53	**99.17**±0.41	66.56±0.58	77.40±0.28	52.47±0.15	60.43±0.26	73.90±0.27	69.41±0.40
RMDS	88.83±0.35	90.76±0.27	93.22±0.80	91.84±0.26	92.23±0.23	**91.51**±0.11	89.80±0.28	92.20±0.21	91.40±0.40
Gram	58.33±4.49	58.98±5.19	72.64±2.34	**91.52**±4.45	62.34±8.27	60.44±3.41	58.66±4.83	71.73±3.20	67.37±5.04
EBO	86.36±0.58	88.80±0.36	**94.32**±2.53	91.79±0.98	89.47±0.70	89.25±0.78	87.58±0.46	91.21±0.92	90.00±1.22
OpenGAN	52.81±7.69	54.62±7.68	**56.14**±24.08	**52.81**±27.60	**56.14**±18.26	53.34±5.79	53.71±7.68	54.61±15.51	**54.31**±17.45
GradNorm	54.43±1.59	55.37±0.41	63.72±7.37	53.91±6.36	52.07±4.09	60.50±5.33	54.90±0.98	57.55±3.22	56.67±4.88
ReAct	85.93±0.83	88.29±0.44	**92.81**±3.03	89.12±3.19	89.38±1.49	**90.35**±0.78	87.11±0.61	90.42±1.41	89.31±1.96
MLS	86.31±0.59	88.72±0.36	**94.15**±2.48	91.69±0.94	89.41±0.71	89.14±0.76	87.52±0.47	91.10±0.89	89.90±1.20
KLM	77.89±0.75	80.49±0.85	85.00±2.04	84.99±1.18	82.35±0.33	78.37±0.33	79.19±0.80	82.68±0.21	81.52±1.08
VIM	87.75±0.28	89.62±0.33	94.76±0.38	94.50±0.48	**95.15**±0.34	89.49±0.39	88.68±0.28	93.48±0.24	91.88±0.37
KNN	**89.73**±0.14	**91.56**±0.26	94.26±0.38	92.67±0.30	93.16±0.24	**91.77**±0.23	**90.64**±0.20	92.96±0.14	92.19±0.27
DICE	77.01±0.88	79.67±0.87	**90.37**±5.97	90.02±1.77	81.86±2.35	74.67±4.98	78.34±0.79	84.23±1.89	82.27±3.43
RankFeat	77.98±2.24	80.94±2.80	75.87±5.22	68.15±7.44	73.46±6.49	**85.99**±3.04	79.46±2.52	75.87±5.06	77.07±4.95
ASH	74.11±1.55	76.44±0.61	83.16±4.66	73.46±6.41	77.45±2.39	79.89±3.69	75.27±1.04	78.49±2.58	77.42±3.76
SHE	80.31±0.69	82.76±0.43	**90.43**±4.76	86.38±1.32	81.57±1.21	82.89±1.22	81.54±0.51	85.32±1.43	84.06±2.16
GEN	87.21±0.36	89.20±0.25	93.83±2.14	91.97±0.66	90.14±0.76	89.46±0.65	88.20±0.30	91.35±0.69	90.30±1.02
NAC	**89.83**±0.29	**92.02**±0.20	94.86±1.37	96.06±0.47	**95.64**±0.45	**91.85**±0.28	**90.93**±0.23	94.60±0.50	**93.37**±0.64
KAN	**90.06**±0.47	**91.92**±0.52	**97.86**±0.73	**97.39**±0.42	**95.85**±0.28	**91.64**±0.91	**90.99**±0.50	**95.69**±0.22	**94.12**±0.59
CIFAR-100 Benchmark									
OpenMax	74.38±0.37	78.44±0.14	76.01±1.39	82.07±1.53	80.56±0.09	79.29±0.40	76.41±0.25	79.48±0.41	78.46±0.88
MSP	78.47±0.07	82.07±0.17	76.08±1.86	78.42±0.89	77.32±0.71	79.22±0.29	80.27±0.11	77.76±0.44	78.60±0.90
TempScale	79.02±0.06	82.79±0.09	77.27±1.85	79.79±1.05	78.11±0.72	79.80±0.25	80.90±0.07	78.74±0.51	**79.46**±0.92
ODIN	78.18±0.14	81.63±0.08	83.79±1.31	74.54±0.76	79.33±1.08	79.45±0.26	79.90±0.11	79.28±0.21	**79.49**±0.77
MDS	55.87±0.22	61.50±0.28	67.47±0.81	70.68±6.40	76.26±0.69	63.15±0.49	58.69±0.09	69.39±1.39	65.82±2.66
MDSEns	43.85±0.31	48.78±0.19	**98.21**±0.78	53.76±1.63	69.75±1.14	42.27±0.73	46.31±0.24	66.00±0.69	59.44±0.93
RMDS	77.75±0.19	82.55±0.02	79.74±2.49	84.89±1.10	83.65±0.51	**83.40**±0.46	80.15±0.11	82.92±0.42	**82.00**±1.15
Gram	49.41±0.58	53.91±1.58	80.71±4.15	**95.55**±0.60	70.79±1.32	46.38±1.21	51.66±0.77	73.36±1.08	66.12±1.98
EBO	79.05±0.11	82.76±0.08	79.18±1.37	82.03±1.74	78.35±0.83	79.52±0.23	80.91±0.08	79.77±0.61	**80.15**±0.97
OpenGAN	63.23±2.44	68.74±2.29	**68.14**±18.78	68.40±2.15	65.84±3.43	**69.13**±7.08	65.98±1.26	67.88±7.16	**67.25**±8.47
GradNorm	70.32±0.20	69.95±0.79	65.35±1.12	76.95±4.73	64.58±0.13	69.69±0.17	70.13±0.47	69.14±1.05	69.47±2.01
ReAct	78.65±0.05	82.88±0.08	78.37±1.59	83.01±0.97	80.15±0.46	80.03±0.11	80.77±0.05	80.39±0.49	**80.52**±0.79
MLS	**79.21**±0.10	82.90±0.05	78.91±1.47	81.65±1.49	78.39±0.84	79.75±0.24	81.05±0.07	79.67±0.57	**80.14**±0.93
KLM	73.91±0.25	79.22±0.28	74.15±2.59	79.34±0.44	75.77±0.45	75.70±0.24	76.56±0.25	76.24±0.52	76.35±1.10
VIM	72.21±0.41	77.76±0.16	81.89±1.02	83.14±3.71	85.91±0.78	75.85±0.37	74.98±0.13	81.70±0.62	**79.46**±1.62
KNN	77.02±0.25	**83.34**±0.16	82.36±1.52	84.15±1.09	83.66±0.83	79.43±0.47	80.18±0.15	82.40±0.17	**81.66**±0.87
DICE	78.04±0.32	80.72±0.30	79.86±1.89	84.22±2.00	77.63±0.34	78.33±0.66	79.38±0.23	80.01±0.18	**79.80**±1.18
RankFeat	58.04±2.36	65.72±0.22	63.03±3.86	72.14±1.39	69.40±3.08	63.82±1.83	61.88±1.28	67.10±1.42	65.36±2.43
ASH	76.48±0.30	79.92±0.20	77.23±0.46	85.60±1.40	80.72±0.70	78.76±0.16	78.20±0.15	80.58±0.66	**79.79**±0.69
SHE	78.15±0.03	79.74±0.36	76.76±1.07	80.97±3.98	73.64±1.28	76.30±0.51	78.95±0.18	76.92±1.16	77.59±1.78
GEN	**79.38**±0.04	**83.25**±0.13	78.29±2.05	81.41±1.50	78.74±0.81	80.28±0.27	**81.31**±0.08	79.68±0.75	**80.23**±1.10
NAC	72.02±0.69	79.86±0.23	93.26±1.34	92.60±1.14	**89.36**±0.54	73.06±0.63	75.94±0.41	**87.07**±0.30	**83.36**±0.84
KAN	72.97±0.17	81.37±0.22	92.29±1.85	**87.16**±4.46	**89.43**±0.39	77.42±0.35	77.17±0.17	**86.57**±0.70	**83.44**±1.99

Table 18: AUROC performance on ImageNet-200 FS and ImageNet-1K FS benchmarks.

Method	Near OOD		Far OOD			Avg Near	Avg Far	Avg Overall
	SSB-hard	NINCO	iNaturalist	Textures	OpenImage-O			
ImageNet-200 FS Benchmark								
OpenMax	47.64±0.20	54.15±0.23	72.44±0.87	69.12±0.36	62.31±0.24	50.89±0.18	67.96±0.39	61.13±0.46
MSP	50.94±0.25	57.76±0.46	70.42±0.67	65.11±0.43	62.80±0.24	54.35±0.35	66.11±0.37	61.41±0.44
TempScale	50.05±0.28	56.86±0.47	70.18±0.76	65.65±0.43	62.29±0.26	53.46±0.37	66.04±0.41	61.01±0.48
ODIN	44.31±0.02	52.36±0.08	70.19±0.92	67.10±0.34	61.48±0.31	48.33±0.05	66.25±0.42	59.09±0.46
MDS	48.59±0.88	56.65±0.94	68.25±1.51	73.84±0.75	61.90±0.57	52.62±0.90	68.00±0.87	61.85±0.98
MDSEns	34.22±0.44	41.58±0.17	43.63±0.48	67.54±0.35	48.38±0.36	37.90±0.20	53.18±0.39	47.07±0.38
RMDS	56.24±0.62	60.95±0.94	71.71±1.49	64.61±1.07	63.52±0.83	58.59±0.77	66.62±1.11	63.41±1.03
Gram	**59.12**±0.73	**63.35**±0.76	58.42±0.75	75.86±0.10	61.51±0.39	**61.23**±0.74	65.26±0.31	63.65±0.61
EBO	47.56±0.29	53.45±0.32	65.91±1.41	68.03±0.48	59.83±0.59	50.51±0.30	64.59±0.76	58.96±0.74
OpenGAN	41.58±5.51	46.51±2.97	61.78±8.75	59.02±4.68	58.53±3.57	44.04±4.21	59.78±4.65	53.48±5.48
GradNorm	53.50±1.03	55.32±0.88	70.38±2.68	73.12±0.59	63.52±1.33	54.41±0.94	69.01±1.39	63.17±1.49
ReAct	47.25±0.57	53.84±0.55	69.45±3.94	71.45±2.04	62.30±2.32	50.55±0.19	67.73±2.76	60.86±2.27
MLS	48.18±0.28	54.43±0.32	67.94±1.20	67.12±0.45	60.66±0.44	51.30±0.30	65.24±0.61	59.67±0.64
KLM	54.03±0.71	60.52±0.13	75.37±0.91	64.12±0.65	66.20±0.29	57.28±0.41	68.56±0.52	64.05±0.61
VIM	45.34±0.72	57.09±1.03	71.34±1.68	**82.54**±0.73	65.70±0.94	51.22±0.86	73.19±1.10	64.40±1.08
KNN	44.05±0.42	54.51±0.62	71.53±1.32	81.88±0.19	62.12±0.79	49.28±0.51	71.84±0.72	62.82±0.77
DICE	48.11±0.51	54.31±0.55	66.53±2.14	72.51±0.66	61.73±0.94	51.21±0.53	66.92±1.12	60.64±1.14
RankFeat	46.41±0.23	43.03±2.29	22.74±4.50	20.60±3.08	40.74±4.41	44.72±1.07	28.03±3.96	34.70±3.30
ASH	50.96±0.93	58.51±0.60	77.96±1.58	79.39±0.61	**69.09**±0.71	54.74±0.74	75.48±0.95	67.18±0.96
SHE	52.82±0.65	56.64±0.69	72.20±2.65	74.27±0.63	64.95±1.25	54.73±0.67	70.47±1.39	64.18±1.41
GEN	48.33±0.27	54.85±0.42	68.94±0.63	66.58±0.47	60.87±0.28	51.59±0.34	65.46±0.44	59.91±0.43
NAC	45.42±0.11	53.80±0.08	65.83±1.22	74.41±0.35	60.79±0.23	49.61±0.06	67.01±0.53	60.05±0.58
KAN	**58.37**±0.47	61.10±0.53	**84.13**±0.35	**83.30**±0.35	**70.40**±0.26	**59.74**±0.46	**79.28**±0.18	**71.46**±0.40
ImageNet-1K FS Benchmark								
OpenMax	53.79	60.28	80.30	73.54	71.88	57.03	75.24	67.96
MSP	56.66	64.93	76.35	69.33	71.28	60.79	72.32	67.71
TempScale	55.71	64.60	77.33	70.84	72.32	60.16	73.50	68.16
ODIN	54.22	60.59	77.43	76.04	73.40	57.41	75.62	68.34
MDS	39.22	52.83	54.06	86.26	60.75	46.02	67.02	58.62
MDSEns	37.13	47.80	53.32	73.39	53.24	42.47	59.98	52.98
RMDS	56.61	67.50	73.48	74.25	72.13	62.06	73.29	68.79
Gram	51.93	60.63	71.36	84.83	69.40	56.28	75.20	67.63
EBO	52.93	60.28	74.01	73.89	72.22	56.61	73.37	66.67
GradNorm	**61.33**	64.06	87.62	85.99	76.85	62.70	83.49	75.17
ReAct	55.34	64.51	87.93	81.08	79.34	59.93	82.78	73.64
MLS	53.56	61.43	75.61	73.42	72.66	57.49	73.90	67.34
KLM	56.87	67.26	80.88	70.73	74.74	62.06	75.45	70.10
VIM	45.88	59.12	72.22	93.09	75.01	52.50	80.10	69.06
KNN	43.78	59.86	67.79	90.29	69.98	51.82	76.02	66.34
DICE	54.01	60.29	82.52	83.89	76.42	57.15	80.94	71.43
RankFeat	50.30	40.37	34.34	66.29	44.98	45.34	48.54	47.26
ASH	54.66	66.38	89.23	89.53	81.47	60.52	86.75	76.25
SHE	58.15	64.27	84.71	87.48	76.92	61.21	83.04	74.31
GEN	52.95	62.73	78.47	71.82	72.62	57.84	74.31	67.72
NAC	52.48	66.49	88.92	92.77	80.76	59.48	87.48	76.28
KAN	55.88	**69.55**	**91.55**	**93.45**	**82.15**	**62.71**	**89.05**	**78.52**

Table 19: Tab. Med. Caucasian Eth. as InD (AUROC metric).

Method	eICU - Eth.
MDS	**58.5**±2.2
RMDS	51.6±1.5
KNN	55.8±1.9
VIM	57.3±2.3
SHE	50.5±1.7
KLM	51.6±2.1
OpenMax	48.7±0.8
MSP	48.4±1.0
MLS	49.1±1.1
TempScale	48.4±1.0
ODIN	48.4±1.0
EBO	49.1±1.1
GRAM	50.8±2.7
GradNorm	50.4±3.4
ReAct	48.1±1.1
DICE	49.9±2.8
ASH	52.6±2.7
KAN	**61.4**±3.1

Table 20: Tab. Med. > 70 y.o. as InD (AUROC metric).

Method	eICU - Age
MDS	**50.8**±1.1
RMDS	48.3±0.7
KNN	49.6±0.2
VIM	48.8±0.1
SHE	**50.4**±0.7
KLM	**51.0**±0.7
OpenMax	48.1±0.5
MSP	48.1±0.5
MLS	48.0±0.5
TempScale	48.1±0.5
ODIN	48.1±0.5
EBO	48.0±0.5
GRAM	48.6±0.6
GradNorm	48.7±0.8
ReAct	48.2±0.7
DICE	48.1±0.6
ASH	48.6±1.3
KAN	**50.5**±0.5

Table 21: Tab. Med. Feature multiplication (AUROC metric).

Method	eICU - Synthetic OOD			Avg Overall
	$\mathcal{F} = 10$	$\mathcal{F} = 100$	$\mathcal{F} = 1000$	
MDS	59.9±1.4	79.5±1.4	87.5±0.9	75.63±1.26
RMDS	51.5±1.3	57.8±7.4	64.0±13.0	57.77±8.67
KNN	57.3±1.4	75.4±2.2	86.5±1.3	73.07±1.68
VIM	57.9±1.6	77.6±1.3	**88.3**±0.7	74.60±1.26
SHE	55.7±1.3	71.2±2.9	80.4±1.6	69.10±2.05
KLM	54.1±0.8	63.1±1.1	72.1±4.2	63.10±2.55
OpenMax	51.0±0.7	56.1±2.7	71.4±3.2	59.50±2.45
MSP	50.9±0.6	55.8±2.7	71.3±3.1	59.33±2.40
MLS	50.9±0.7	55.8±3.1	70.8±3.1	59.17±2.56
TempScale	50.9±0.6	55.8±2.7	71.3±3.1	59.33±2.40
ODIN	50.9±0.6	55.9±2.7	71.4±3.1	59.40±2.40
EBO	50.9±0.7	55.5±3.1	70.2±3.2	58.87±2.60
GRAM	48.9±0.1	50.1±0.8	62.5±3.0	53.83±1.79
GradNorm	52.0±0.8	59.5±2.9	73.6±2.9	61.70±2.41
ReAct	50.9±0.8	55.7±3.4	69.9±3.2	58.83±2.73
DICE	52.2±1.5	62.8±2.5	75.6±1.4	63.53±1.87
ASH	50.1±1.4	54.6±2.1	69.7±1.3	58.13±1.64
KAN	**64.6**±2.2	**83.0**±2.6	**89.8**±1.8	**79.13**±2.22

Table 22: FPR@95 performance on CIFAR-10 and CIFAR-100 benchmarks.

Method	Near OOD		Far OOD				Avg Near	Avg Far	Avg Overall
	CIFAR	TIN	MNIST	SVHN	Textures	Places365			
CIFAR-10 Benchmark									
OpenMax	48.06±3.25	39.18±1.44	23.33±4.67	25.40±1.47	31.50±4.05	38.52±2.27	43.62±2.27	29.69±1.21	34.33±3.11
MSP	53.08±4.86	43.27±3.00	23.64±5.81	25.82±1.64	34.96±4.64	42.47±3.81	48.17±3.92	31.72±1.84	37.21±4.19
TempScale	55.81±5.07	46.11±3.63	23.53±7.05	26.97±2.65	38.16±5.89	45.27±4.50	50.96±4.32	33.48±2.39	39.31±5.01
ODIN	77.00±5.74	75.38±6.42	**23.83**±12.34	68.61±0.52	67.70±11.06	70.36±6.96	76.19±6.08	57.62±4.24	63.81±8.14
MDS	52.81±3.62	46.99±4.36	27.30±3.55	25.96±2.52	27.94±4.20	47.67±4.54	49.90±3.98	32.22±3.40	38.11±3.86
MDSEns	91.87±0.10	92.66±0.42	**1.30**±0.51	74.34±1.04	76.07±0.17	94.16±0.33	92.26±0.20	61.47±0.48	71.73±0.53
RMDS	43.86±3.49	33.91±1.39	21.49±2.32	23.46±1.48	25.25±0.53	31.20±0.28	38.89±2.39	25.35±0.73	29.86±1.92
Gram	91.68±2.24	90.06±1.59	70.30±8.96	**33.91**±17.35	94.64±2.71	90.49±1.93	90.87±1.91	72.34±6.73	78.51±8.16
EBO	66.60±4.46	56.08±4.83	**24.99**±12.93	35.12±6.11	51.82±6.11	54.85±6.52	61.34±4.63	41.69±5.32	48.24±7.39
OpenGAN	94.84±3.83	94.11±4.21	79.54±19.71	**75.27**±26.93	83.95±14.89	95.32±4.45	94.48±4.01	83.52±11.63	87.17±15.21
GradNorm	94.54±1.11	94.89±0.60	85.41±4.85	91.65±2.42	98.09±0.49	92.46±2.28	94.72±0.82	91.90±2.23	92.84±2.46
ReAct	67.40±7.34	59.71±7.31	**33.77**±18.00	**50.23**±15.98	51.42±11.42	44.20±3.35	63.56±7.33	44.90±8.37	**51.12**±11.75
MLS	66.59±4.44	56.06±4.82	**25.06**±12.87	35.09±6.09	51.73±6.13	54.84±6.51	61.32±4.62	41.68±5.27	48.23±7.37
KLM	90.55±5.83	85.18±7.60	76.22±12.09	59.47±7.06	81.95±9.95	95.58±2.12	87.86±6.37	78.31±4.84	81.49±8.08
VIM	49.19±3.15	40.49±1.55	18.36±1.42	**19.29**±0.41	21.14±1.83	41.43±2.17	44.84±2.31	25.05±0.52	31.65±1.94
KNN	37.64±0.31	30.37±0.65	20.05±1.36	22.60±1.26	24.06±0.55	30.38±0.63	34.01±0.38	24.27±0.40	27.52±0.88
DICE	73.71±7.67	66.37±7.68	30.83±10.54	36.61±4.74	62.42±4.79	77.19±12.60	70.04±7.64	51.76±4.42	57.85±8.50
RankFeat	65.32±3.48	56.44±5.76	61.86±12.78	64.49±7.38	59.71±9.79	**43.70**±7.39	60.88±4.60	57.44±7.99	58.59±8.30
ASH	87.31±2.06	86.25±1.58	70.00±10.56	83.64±6.48	84.59±1.74	77.89±7.28	86.78±1.82	79.03±4.22	81.61±6.00
SHE	81.00±3.42	78.30±3.52	**42.22**±20.59	62.74±4.01	84.60±5.30	76.36±5.32	79.65±3.47	66.48±5.98	70.87±9.31
GEN	58.75±3.97	48.59±2.34	23.00±7.75	28.14±2.59	40.74±6.61	47.03±3.22	53.67±3.14	34.73±1.58	41.04±4.87
NAC	**35.15**±0.40	**26.55**±0.18	15.13±2.62	**14.33**±1.26	**17.05**±0.60	**26.73**±0.81	**30.85**±0.19	**18.31**±0.93	**22.49**±1.27
KAN	**40.81**±2.91	34.17±1.85	11.63±3.11	**14.13**±2.38	21.30±0.98	32.49±2.21	37.49±2.36	**19.89**±1.19	**25.75**±2.35
CIFAR-100 Benchmark									
OpenMax	**60.17**±0.97	52.99±0.51	53.82±4.74	53.20±1.78	56.12±1.91	**54.85**±1.42	56.58±0.73	54.50±0.68	**55.19**±2.33
MSP	**58.91**±0.93	**50.70**±0.34	57.23±4.68	59.07±2.53	61.88±1.28	56.62±0.87	**54.80**±0.33	58.70±1.06	**57.40**±2.30
TempScale	**58.72**±0.81	**50.26**±0.16	56.05±4.61	57.71±2.68	61.56±1.43	56.46±0.94	**54.49**±0.48	57.94±1.14	**56.79**±2.31
ODIN	60.64±0.56	55.19±0.57	45.94±3.29	67.41±3.88	62.37±2.96	59.71±0.92	57.91±0.51	58.86±0.79	58.54±2.45
MDS	88.00±0.49	79.05±1.22	71.72±2.94	67.21±6.09	70.49±2.48	79.61±0.34	83.53±0.60	72.26±1.56	76.01±2.99
MDSEns	95.94±0.16	95.82±0.12	**2.83**±0.86	82.57±2.58	84.94±0.83	96.61±0.17	95.88±0.04	66.74±1.04	76.45±1.17
RMDS	61.37±0.24	**49.56**±0.90	52.05±6.28	51.65±3.68	53.99±1.06	**53.57**±0.43	55.46±0.41	52.81±0.63	**53.70**±3.03
Gram	92.71±0.64	91.85±0.86	53.53±7.45	**20.06**±1.96	89.51±2.54	94.67±0.60	92.28±0.29	64.44±2.37	73.72±3.35
EBO	**59.21**±0.75	52.03±0.50	52.62±3.83	53.62±3.14	62.35±2.06	57.75±0.86	**55.62**±0.61	56.59±1.38	**56.26**±2.25
OpenGAN	78.83±3.94	74.21±1.25	63.09±23.25	70.35±2.06	74.77±1.78	**73.75**±8.32	76.52±2.59	70.49±7.38	**72.50**±10.28
GradNorm	84.30±0.36	86.85±0.62	86.97±1.44	69.90±7.94	92.51±0.61	85.32±0.44	85.58±0.46	83.68±1.92	84.31±3.32
ReAct	61.30±0.43	51.47±0.47	56.04±5.66	50.41±2.02	55.04±0.82	55.30±0.41	56.39±0.34	54.20±1.56	**54.93**±2.50
MLS	**59.11**±0.64	51.83±0.70	52.95±3.82	53.90±3.04	62.39±2.13	57.68±0.91	**55.47**±0.66	56.73±1.33	**56.31**±2.24
KLM	84.77±2.95	71.07±0.59	73.09±6.67	50.30±7.04	81.80±5.80	81.40±1.58	77.92±1.31	71.65±2.01	73.74±4.82
VIM	70.59±0.43	54.66±0.42	48.32±1.07	46.22±5.46	46.86±2.29	61.57±0.77	62.63±0.27	50.74±1.00	**54.70**±2.49
KNN	72.80±0.44	**49.65**±0.37	48.58±4.67	51.75±3.12	53.56±2.32	60.70±1.03	61.22±0.14	53.65±0.28	**56.17**±2.53
DICE	**60.98**±1.10	54.93±0.53	51.79±3.67	49.58±3.32	64.23±1.65	59.39±1.25	57.95±0.53	56.25±0.60	**56.82**±2.25
RankFeat	82.78±1.56	78.40±0.95	75.01±5.83	58.49±2.30	66.87±3.80	77.42±1.96	80.59±1.10	69.45±1.01	73.16±3.19
ASH	68.06±0.44	63.35±0.90	66.58±3.88	46.00±2.67	61.27±2.74	62.95±0.99	65.71±0.24	59.20±2.46	61.37±2.30
SHE	60.41±0.51	57.74±0.73	58.78±2.70	59.15±7.61	73.29±3.22	65.24±0.98	59.07±0.25	64.12±2.70	62.44±3.59
GEN	**58.87**±0.69	**49.98**±0.05	53.92±5.71	55.45±2.76	61.23±1.40	56.25±1.01	**54.42**±0.33	56.71±1.59	**55.95**±2.70
NAC	81.37±1.27	61.59±0.78	21.97±6.62	**24.39**±4.46	**40.65**±1.94	73.57±1.16	71.48±1.03	**40.14**±1.86	**50.59**±3.44
KAN	82.28±1.76	55.74±1.94	39.61±7.67	57.13±12.74	**45.76**±2.87	65.33±0.61	69.01±1.85	51.96±3.31	**57.64**±6.28

Table 23: FPR@95 performance on ImageNet-200 FS and ImageNet-1K FS benchmarks.

Method	Near OOD		Far OOD			Avg Near	Avg Far	Avg Overall
	SSB-hard	NINCO	iNaturalist	Textures	OpenImage-O			
ImageNet-200 FS Benchmark								
OpenMax	91.55±0.05	85.00±0.26	68.19±0.77	76.72±0.33	77.39±0.28	88.27±0.14	74.10±0.46	79.77±0.41
MSP	89.08±0.03	**79.30**±0.42	67.24±0.64	79.79±0.36	74.22±0.10	**84.19**±0.21	73.75±0.26	77.93±0.38
TempScale	89.33±0.04	**79.51**±0.37	**66.61**±0.68	79.67±0.39	74.10±0.22	**84.42**±0.18	73.46±0.24	77.84±0.40
ODIN	91.71±0.09	86.66±0.39	**65.50**±1.53	79.13±0.77	76.03±0.28	89.18±0.16	73.55±0.63	79.81±0.80
MDS	93.90±0.16	88.52±0.16	74.46±1.37	74.10±1.07	83.68±0.52	91.21±0.07	77.41±0.78	82.93±0.82
MDSEns	96.28±0.11	95.83±0.15	91.39±0.26	84.83±0.16	93.52±0.13	96.06±0.13	89.91±0.10	92.37±0.17
RMDS	89.44±0.17	**79.23**±0.48	**65.62**±1.16	76.73±0.68	74.82±0.09	**84.33**±0.31	72.39±0.64	77.17±0.64
Gram	**86.37**±0.25	87.59±1.45	86.24±0.49	82.76±0.49	87.23±0.84	86.98±0.77	85.41±0.28	86.04±0.82
EBO	90.71±0.15	83.61±0.39	70.53±1.68	79.46±0.88	77.14±0.64	87.16±0.27	75.71±0.83	80.29±0.91
OpenGAN	95.90±0.21	94.47±1.39	81.85±4.09	84.76±1.87	86.71±0.61	95.18±0.60	84.44±0.80	88.74±2.12
GradNorm	91.16±0.41	91.68±0.31	78.68±2.25	82.11±1.95	84.81±0.53	91.42±0.32	81.86±0.44	85.69±1.37
ReAct	91.22±0.50	84.61±0.72	**67.52**±2.93	72.82±1.56	74.81±0.18	87.91±0.61	71.72±1.54	**78.20**±1.54
MLS	90.68±0.17	83.27±0.42	69.50±1.61	79.39±0.85	76.55±0.43	86.98±0.28	75.15±0.78	79.88±0.86
KLM	91.97±0.85	85.33±1.20	66.87±1.12	80.11±0.48	77.94±0.45	88.65±0.36	74.97±0.54	80.44±0.88
VIM	91.61±0.15	82.35±0.44	68.15±0.74	**58.50**±0.85	74.54±0.51	86.98±0.28	**67.06**±0.63	**75.03**±0.59
KNN	91.73±0.15	81.23±0.35	69.10±0.74	69.06±0.22	74.43±0.66	86.48±0.25	70.86±0.50	77.11±0.49
DICE	90.94±0.09	84.24±0.51	72.10±1.81	78.84±0.87	77.79±0.68	87.59±0.29	76.24±0.78	80.78±0.98
RankFeat	95.78±0.10	96.98±0.18	99.16±0.31	99.75±0.28	98.16±0.35	96.38±0.14	99.02±0.30	97.97±0.26
ASH	90.29±0.42	84.21±0.45	**63.16**±1.71	65.99±0.56	**71.69**±0.41	87.25±0.40	**66.95**±0.64	**75.07**±0.87
SHE	91.16±0.13	86.49±0.62	71.48±2.66	78.98±1.32	79.50±0.69	88.82±0.36	76.65±0.88	81.52±1.39
GEN	89.59±0.05	**80.12**±0.33	**66.47**±0.74	79.30±0.45	73.96±0.26	84.85±0.14	73.24±0.40	77.89±0.43
NAC	92.75±0.35	88.83±0.12	72.57±1.49	69.08±0.63	79.55±0.72	90.79±0.16	73.73±0.66	80.56±0.81
KAN	94.67±1.20	87.35±1.49	**64.89**±3.67	68.75±1.79	78.84±1.55	91.01±1.35	70.83±0.25	**78.90**±2.13
ImageNet-1K FS Benchmark								
OpenMax	86.76	76.41	49.98	62.28	60.05	81.59	57.44	67.10
MSP	84.47	72.37	61.75	75.27	67.29	78.42	68.10	72.23
TempScale	**84.23**	71.55	58.16	72.87	64.35	77.89	65.13	70.23
ODIN	86.01	80.40	57.10	67.34	65.45	83.21	63.29	71.26
MDS	95.41	86.89	83.38	55.95	82.16	91.15	73.83	80.76
MDSEns	96.22	93.56	87.47	78.56	92.72	94.89	86.25	89.71
RMDS	87.09	**70.28**	55.68	67.83	61.30	78.69	61.61	68.44
Gram	90.87	86.24	72.69	64.65	79.05	88.55	72.13	78.70
EBO	86.19	76.15	55.18	66.05	60.55	81.17	60.59	68.82
GradNorm	84.44	85.45	45.63	55.84	76.76	84.95	59.41	69.62
ReAct	86.90	72.97	40.64	53.47	55.89	79.94	50.00	61.97
MLS	85.98	75.43	54.59	66.34	60.39	80.71	60.44	68.55
KLM	90.19	74.14	58.60	68.69	66.22	82.17	64.50	71.57
VIM	88.85	77.85	55.59	33.64	57.23	83.35	48.82	62.63
KNN	90.50	74.77	62.46	42.59	65.03	82.64	56.69	67.07
DICE	86.93	79.87	54.89	63.84	66.53	83.40	61.75	70.41
RankFeat	93.17	96.21	96.43	83.49	93.59	94.69	91.17	92.58
ASH	84.49	70.74	**36.82**	38.34	**52.33**	**77.61**	42.50	**56.54**
SHE	85.10	80.47	50.97	52.17	69.38	82.78	57.50	67.62
GEN	85.72	72.38	50.61	66.40	57.67	79.05	58.23	66.56
NAC	89.54	76.53	36.88	**29.19**	58.48	83.03	**41.52**	58.12
KAN	89.64	74.57	42.67	37.79	67.66	82.11	49.37	62.47

Table 24: Tab. Med. Caucasian Eth. as InD (FPR@95 metric).

Method	eICU - Eth.
MDS	**89.7**±4.3
RMDS	**93.1**±1.6
KNN	**91.8**±2.0
VIM	**90.7**±4.3
SHE	**93.2**±1.8
KLM	**92.8**±1.4
OpenMax	**93.8**±1.9
MSP	**93.5**±1.6
MLS	**93.4**±1.6
TempScale	**93.5**±1.6
ODIN	**93.5**±1.6
EBO	**93.3**±1.6
GRAM	**93.6**±2.3
GradNorm	**92.0**±3.2
ReAct	**94.5**±2.3
DICE	**93.0**±2.2
ASH	**92.6**±2.1
KAN	**91.6**±3.4

Table 25: Tab. Med. > 70 y.o. as InD (FPR@95 metric).

Method	eICU - Age
MDS	**94.3**±0.5
RMDS	95.6±0.1
KNN	**94.7**±0.2
VIM	**94.6**±0.0
SHE	95.1±0.3
KLM	95.2±0.3
OpenMax	95.2±0.1
MSP	95.3±0.2
MLS	95.3±0.1
TempScale	95.3±0.2
ODIN	95.3±0.2
EBO	95.2±0.1
GRAM	95.0±0.3
GradNorm	95.3±0.2
ReAct	95.2±0.2
DICE	**94.9**±0.1
ASH	95.3±0.1
KAN	97.3±0.9

Table 26: Tab. Med. Feature multiplication (FPR@95 metric).

Method	eICU - Synthetic OOD			Avg Overall
	$\mathcal{F} = 10$	$\mathcal{F} = 100$	$\mathcal{F} = 1000$	
MDS	**81.6**±1.3	**48.1**±4.1	**28.0**±4.9	**52.57**±3.76
RMDS	91.9±2.3	78.7±10.4	64.2±17.7	78.27±11.93
KNN	83.6±1.8	**53.4**±5.3	**29.8**±5.6	**55.60**±4.57
VIM	**81.9**±2.8	**48.7**±3.5	**26.6**±3.0	**52.40**±3.11
SHE	88.2±2.5	63.0±6.6	41.6±4.4	64.27±4.80
KLM	91.0±1.8	79.5±4.4	60.9±9.7	77.13±6.24
OpenMax	91.2±1.9	80.7±4.8	60.6±9.2	77.50±6.09
MSP	91.0±1.4	79.2±2.5	58.8±5.3	76.33±3.48
MLS	91.2±1.0	80.4±3.7	61.0±7.7	77.53±4.97
TempScale	91.0±1.4	79.2±2.5	58.8±5.3	76.33±3.48
ODIN	90.9±1.4	79.0±2.4	58.4±5.1	76.10±3.35
EBO	91.6±0.8	82.0±5.1	63.7±9.0	79.10±5.99
GRAM	94.1±0.6	89.8±4.7	79.9±8.3	87.93±5.52
GradNorm	91.3±1.4	78.6±4.5	58.3±6.7	76.07±4.73
ReAct	91.8±0.7	82.9±3.9	65.5±6.7	80.07±4.49
DICE	88.3±1.1	66.9±2.3	45.1±2.7	66.77±2.14
ASH	91.6±0.7	81.0±4.5	61.0±6.9	77.87±4.77
KAN	**77.2**±4.0	**49.3**±5.5	33.7±5.2	**53.42**±4.94