OUT OF DISTRIBUTION DETECTION WITH NULL SPACE ANALYSIS

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural network development and training often relies on highly curated datasets. The use of curated data does not adequately take into account the possibility of encountering unexpected inputs or outliers. However, encountering outliers is common when deploying systems in real world applications. Since outliers are samples not drawn from the data distribution of interest, the diversity of outliers precludes the ability to train with fully representative outlier data. One mechanism to create more robust neural network systems that can gracefully handle outliers or unexpected inputs is to leverage out-of-distribution (OOD) detection. In this paper we introduce Null Space Analysis (NuSA) as a way to manipulate the null space inherent in many neural networks to detect OOD data. NuSA functions by minimizing the null space projections for inlier data, so that when OOD data is encountered a much larger null space projection will occur. Experiments show an improvement over existing state-of-the-art algorithms using a fixed architecture and several OOD datasets.

1 INTRODUCTION

Artificial neural network (ANN) and deep learning-based approaches are increasingly being incorporated into real-world applications and systems. Unlike in curated benchmark datasets, unexpected data is commonly encountered in systems deployed for real-world application. For example, encountering a buried soda can when characterizing underground plant root systems or a black and white image in a color image dataset. Thus, the ability to accurately identify the unexpected data to ensure reliability of ANN-dependent systems is essential. Often detecting unexpected data in this way is referred to as outlier detection Bansal et al. (2016). Outlier detection can be geared towards subtle differences as in Szegedy et al. (2013); Fawzi et al. (2018) or can be entirely different classes/distributions. This second instance is commonly referred to as out-of-distribution (OOD) detection Salehi et al. (2021).

In practice, OOD is the process of detecting when data from unknown classes or datasets are shown to an algorithm alongside the expected data. Without mechanisms to detect unexpected data, ANNs will classify every sample (sometimes with high confidence Moosavi-Dezfooli et al. (2015)) to a class found in the training data. Imagine an ANN trained to classify between cats and dogs. If the network is presented with a bird it must choose between a cat or a dog, and may do so with high confidence.

There are several methods that take aim at the OOD task. However, many of these algorithms work directly on the network outputs, i.e. Liang et al. (2017) and Sun et al. (2021), or work on existing pre-trained networks as in Wei & Wang (2023). These methods take a post-hoc approach to OOD, that is, they introduce functions that are added to any existing network, after it has been trained, to allow for OOD detection. This paradigm also limits nearly all of these methods to only functioning within classification tasks, which is a small subset of the available use cases for competency aware learning. Another weakness of this type of algorithm is that there is no enhancement of the OOD detection performance. In other words, post-hoc methods only exploit existing differences between expected and unexpected data. This means that the performance of these methods may be limited by the inherent performance of the network being used.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

In this paper, we define a novel method of OOD detection called NuSA. As opposed to post-hoc methods our method is incorporated during the training process to encourage better performance. The contributions of this paper are summarized by:

- We propose a method to measure how much data is projected in an ANNs null space.
- We propose a novel calculation, internal to neural networks, that uses the null space to increase OOD detection performance.
- Experiments using CIFAR10, SVHN and others show the proposed method outperforms similar methods.

2 Related Work

Before delving into the methodology of NuSA first we will describe competing algorithms and other uses of the null space in ANNs.

2.1 Out of Distribution Detection

A variety of outlier detection methods have been developed in the literature. One common approach to outlier detection is to use distances to nearby points Breunig et al. (2000); Goldstein & Dengel (2012). An example of this is to use the distance to the k nearest neighbors as an outlier score Ramaswamy et al. (2000). The Angle-Based Outlier Detection Kriegel et al. (2008) which uses both the distance and angle between points to identify outliers is a slight variation on the standard distance based methods. Another class of outlier detection algorithms are one-class classifiers. A popular example of these algorithms is the One-Class Support Vector Machine Schölkopf et al. (2001). In this case the Support Vector Machine (SVM) is used to estimate a hyperplane that encompasses the training data and outliers can be detected based on their distance from the hyperplane. Using the major and minor principal components of known "good" data to generate a hyperplane for comparison has also been used Shyu et al. (2003). The Isolation Forest Liu et al. (2008) is another method that uses a random forest to find outliers, identifying them by analyzing the number of graph edges needed to isolate data points. Methods that identify outliers by having a large reconstruction error using a model fit with the training data have also been used. Recently, the most popular examples of these methods are auto-encoders Aggarwal (2015) or variational auto-encoders Kingma & Welling (2013). However, all of these methods are capable of only one thing, outlier detection.

More recently, out-of-distribution (OOD) detection has become the popular term for outlier detection. The difference is that many of the OOD methods incorporate the outlier detection into an existing classification model, typically an ANN of some kind. Methods such as Bitterwolf et al. (2020); Hein et al. (2019); Hendrycks et al. (2018) achieve good performance, however, they require some knowledge of OOD data during training. In general, these methods use one dataset to optimize classification and include a second dataset to use as outliers during training. Knowledge of both inliers and outliers allows the model to correctly classify the inliers while learning a space where the inliers are separated from the outliers. These methods can be used for unseen OOD data but they are not unsupervised OOD detectors. To this end methods that generate OOD data during training attempt to overcome this shortcoming. In Nakamura et al. (2024) pseudo-outliers are generated using Q-Gaussian distributions based on the existing training data thereby not requiring known OOD data prior to training. Methods such as Li et al. (2022) use high levels of data augmentation, along with a contrastive loss function between heavily and lightly augmented data during network training, to move away from the requirement of OOD data during training. Further examples of improving OOD detection performance via data augmentation are Ye et al. (2024) and Tan et al. (2024).

Unsupervised OOD methods are preferred to mimic the traditional outlier detection paradigm. Methods such as Zhao et al. (2024) or Sehwag et al. (2021) operate on similar principles to the previously defined distance based detection methods. Here the distribution of inliers is defined internally to an ANN at a specific layer, and data too far away from the distributions are considered outliers. In Pad (2020) the authors investigated using one-vs-all classifiers to provide a possibilistic output layer. Traditionally a probabilistic output is used. The possibilistic output layer allows the ANN to assign very low confidence scores along all output classes. This can simulate an unknown class label. If all

outputs are below a threshold the sample is identified as an outlier. In DeVries & Taylor (2018) the authors add a second output layer in parallel to the traditional ANN output layer. The additional output consists of a single neuron and the output of this neuron is minimized during training. Therefore, for known data the output is close to zero, due to a sigmoid function, and for unknown data the value will be larger. Combined with a modified training algorithm where the fast gradient sign method (FGSM) Goodfellow et al. (2014) is used, very good performance can be achieved. Additionally, methods such as Janjua et al. (2024), use the training labels to enforce tight within class projections to encourage inlier data to be compact so that outlier data is further from the class centers during tests.

We propose a method that leverages the null space of existing ANN architectures to detect data that is unfamiliar to a trained ANN. Null Space Analysis (NuSA) that is capable of encoding outlier detection directly into an ANN, thereby increasing the competency awareness of the ANN.

2.2 Null Space

In our NuSA approach, we leverage the null space associated with the weight matrix representation of a fully connected layer within an ANN. Using the existing architecture allows outlier detection to run in parallel to classification in the ANN without needing any modifications to the architecture. For review, the null space of a matrix is defined as:

$$\mathcal{N}(\boldsymbol{A}) = \left\{ \boldsymbol{z} \in \mathbb{R}^{K\times 1} \,|\, \boldsymbol{A}\boldsymbol{z} = \boldsymbol{0} \right\}, \tag{1}$$

where $\boldsymbol{A} \in \mathbb{R}^{M\times K}$ represents the linear map $f_{\boldsymbol{A}} : \mathbb{R}^K \rightarrow \mathbb{R}^M$ Hefferon (2012). In other words, the null space of matrix $\boldsymbol{A}$ defines the region of the input space that maps to zero for the given linear mapping. The row space, which is orthogonal to the null space, defines the subspace spanned by the rows of $\boldsymbol{A}$, i.e. the row space projection contains all of the information from $\boldsymbol{z}$ that contributes to the output of the mapping via $\boldsymbol{A}$ all possible outputs of the linear mapping. This data is then propagated on to successive layers in an ANN. For ease of reading we will refer to the left null space as the null space for the remainder of this article.

The null space will always contain the trivial solution, $\boldsymbol{A0} = \boldsymbol{0}$, however, we are interested in cases with non-trivial null spaces, that is, we assume $M < K$. The dimensionality of the null space is directly related to the architecture of the ANN. The rank nullity theorem,

$$K = \text{nullity}(\boldsymbol{A}) + \text{rank}(\boldsymbol{A}), \tag{2}$$

states the relationship between the weight matrix $\boldsymbol{A}$, its rank, and the dimensionality of the null space, (the nullity). The fully connected layers within an ANN are commonly used to reduce the dimensionality of the feature space prior to classification, indicating $M < K$. If $M < K$, then $\text{rank}(\boldsymbol{A}) \leq M$, implying that the $\text{nullity}(\boldsymbol{A}) > 0$. Therefore, the null space will be non-trivial. The null space can be non-trivial in other cases, like when the weight matrix $\boldsymbol{A}$ contains linearly dependent rows.

The null space has been used for many different tasks within the ANN literature over a variety of tasks. Several early works looked at the null space of the input matrix and described how the null space of the input matrix can either be accessed via nonlinear transformations within a neural network Goggin et al. (1992) or tracked during training to identify blind channels in the input Dong & Liu (1998). In Zhang et al. (2016) and Li et al. (2016) the authors use a modified linear discriminant analysis (LDA) that computes the null space as a distance metric to do human re-identification. Both of these methods use the null space but it is not with respect to the layers of the ANN, and instead focus on the data null space or on exterior projections. More related to the proposed method are methods that also inside ANNs for different purposes. In Abdelpakey & Shehata (2020) the authors use a method similar to Zhang et al. (2016), mentioned previously, to project input data from the same class onto a single point in the joint null/feature space.

For applications more similar to the proposed method, there is the incremental kernel null space discriminant analysis (IKNDA) Liu et al. (2017), in which the authors define a novelty detection algorithm based on a similar approach to Zhang et al. (2016) and Abdelpakey & Shehata (2020). However, this method is stand alone and not directly implemented within an ANN. In Idnani et al. (2023) the authors define a basis for a new failure mode based on the null space of the feature space within an ANN, the authors then proceed to define a training procedure to improve OOD

detection performance by adding new projection layers to an existing ANN architecture. Null Space Avoidance Medapati (2019) defines a precursor to NuSA that instead attempted to maximize the projection on the column space, via Given's rotation matrices during optimization.

In the proposed method we define a method that is capable of manipulating the null space, via a unique computation that does not add complexity to an existing ANN, that improves the ability to detect OOD data. The NuSA approach is a partial, but important, solution to the problem of competency awareness of ANNs. It is unlikely that there is one method alone that can alleviate this problem.

3 Null Space Analysis of Neural Networks

In this section, we delve into more specific detail on how the null space can be used within the existing architecture of an ANN for OOD detection.

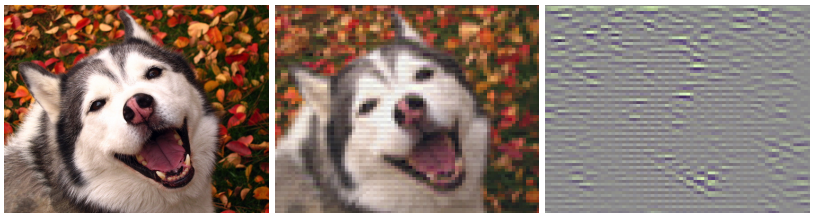


Figure 1: Left: Original image. Center: Null space information, data the network doesn't consider. Right: Row space information, data that proceeds through the layer and into the remainder of the neural network. Image from of ImageNet dataset Deng et al. (2009).

3.1 Origins of Null Space Analysis

For the analysis of this method we follow the Hammerstein interpretation of neural networks in which a layer is defined as a nonlinearity followed by the linear mapping Hammerstein (1930); Janczak (2005). Therefore, each fully connected layer of an ANN is modeled as the application of a nonlinear activation function to the input sample followed by a mapping via the layers' weight matrix. This sample is either the input data point at the first layer or the output of a previous layer for any hidden layers. Every weight matrix has an associated null space, although some may be trivial, i.e. they contain only the zero basis. However, any weight matrix that projects into a lower dimensional space has a non-empty null space. The overwhelming majority of all commonly used deep learning architectures consist of several subsequent layers that project into lower dimensional spaces. Therefore, there exists a series of non-empty null spaces in these networks corresponding to the individual layers' weight matrices.

The computed null space does not account for the nonlinearity, thus does not represent the complete layer. However, the Hammerstein model ensures any projection onto the null space is done with the nonlinearly transformed data. Therefore, when studying the null space projections, the output of the layer is accounted for and we can say that the computed null space describes the layer output.

As data is passed through a network, at each layer with a non-trivial null-space, information from the input will be lost to the null space. The information lost to the null space is therefore invisible to downstream layers of the network. Therefore, as part of the standard network training, the information lost is not useful, or is not as useful as other information kept, to the task the network is performing, i.e. classification. To visualize this consider the task of classifying the type of vehicle in an image, the color of the vehicle is not as important to the classification as the shape of the vehicle, therefore information about the color could be thrown away and classification can still be done.

For clarity, the null space information concept is first illustrated using a simple one-layer ANN with K input features and M outputs with $K > M$. It is then illustrated for multi-layer networks. Assume $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$, with $\boldsymbol{x}_n \in \mathbb{R}^{K\times 1}$, is a collection of samples drawn from the joint distribution of the classes of interest, that is, $\boldsymbol{x}_n \sim p_{\mathcal{D}}$ and that a network, $f_{\boldsymbol{X}}$ has been trained to approximate the true mapping, $\boldsymbol{y}_n = f(\boldsymbol{x}_n)$. In the one-layer case, $\hat{\boldsymbol{y}}_n = f_{\boldsymbol{X}}(\boldsymbol{x}_n) = \boldsymbol{W}\boldsymbol{x}_n$. Since

$K > M$, there is a non-trivial null space, $\mathcal{N}(\boldsymbol{W})$, of dimension $K-M$. Let $\boldsymbol{x}_n = \boldsymbol{x}_n^{(rs)} + \boldsymbol{x}_n^{(ns)}$ with $\boldsymbol{x}_n^{(rs)}$ and $\boldsymbol{x}_n^{(ns)}$ being the portions of the input that project onto the row and null spaces respectively. As $\boldsymbol{x}_n^{(ns)} \in \mathcal{N}(\boldsymbol{W})$ we have $\boldsymbol{W}\boldsymbol{x}_n^{(ns)} = 0$, therefore we can conclude:

$$\begin{aligned} \hat{\boldsymbol{y}}_n = \boldsymbol{W}\boldsymbol{x}_n &= \boldsymbol{W}\left(\boldsymbol{x}_n^{(rs)} + \boldsymbol{x}_n^{(ns)}\right) = \boldsymbol{W}\left(\boldsymbol{x}_n^{(rs)}\right) + \boldsymbol{W}\left(\boldsymbol{x}_n^{(ns)}\right) \\ &= \boldsymbol{W}\left(\boldsymbol{x}_n^{(rs)}\right), \end{aligned} \tag{3}$$

which illustrates that $\boldsymbol{x}_n^{(ns)}$ was lost to the null space.

In a multi-layer network, there are many layers of matrix-vector multiplications. Each layer can be expressed as

$$\boldsymbol{x}_{n,h} = \sigma_h(\boldsymbol{W}_h\boldsymbol{x}_{n,h-1}),$$

where $\sigma_h \forall h = 1, 2, \ldots, N_H$ are nonlinear activation functions and $\boldsymbol{x}_{n,h}$ is the n^{th} input to the $(h+1)^{\text{th}}$ layer of the network (i.e. $\boldsymbol{x}_{n,0}$ is the input to the first layer and $\boldsymbol{x}_{n,1}$ is the input to the second layer). In other words,

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = \sigma_{N_H}(\boldsymbol{W}_{N_H}\sigma_{N_H-1}(\cdots(\boldsymbol{W}_2\sigma_1(\boldsymbol{W}_1\boldsymbol{x}_n)))). \tag{4}$$

Therefore, in a large network there is a continual loss of information to the null space throughout a network.

With this example the information lost throughout a network can easily be confirmed, but how is this information loss related to OOD detection? Generally speaking, in a classification task, a network will learn to focus on only the information necessary for classification, the discarded information is thrown into the null spaces created at each layer that reduces dimensionality. Upon training the weights are fixed meaning the network has learned what type of information is important for classification and what is not as useful. Previous methods such as Sehwag et al. (2021) have focused on using the data the network considers important. However, much more of the data is actually thrown away than is kept. Therefore, there may be more information in the null space projections, which could allow better identification of the OOD data. An example of the information thrown away in the null space of a small network layer is shown in figure 1.

3.2 Calculating NuSA

The goal of NuSA is to leverage the null space to identify outliers. NuSA minimizes the magnitude of the projection of all training data onto the null space of a fully connected layer. This manipulates the weight matrix so that the fully connected layer loses very little information during calculations. During tests, the magnitude of the null space projections can be used to indicate if a sample is an outlier.

The first step in computing NuSA is to compute the null space. Here, we compute the null space using a singular value decomposition (SVD), defined as

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T. \tag{5}$$

Where the columns of $\boldsymbol{U}$ and $\boldsymbol{V}$ are the left and right singular vectors of $\boldsymbol{A}$, respectively. $\boldsymbol{\Sigma}$ is a diagonal matrix of the singular values of $\boldsymbol{A}$, the number of non-zero singular values indicates the rank of matrix $\boldsymbol{A}$. The null space is spanned by the columns of $\boldsymbol{U}$ that correspond to the zero valued entries in the diagonal of $\boldsymbol{\Sigma}$.

To compute NuSA, we must then select an appropriate fully connected layer to use. Many modern CNNs only have a single fully connected layer at the output, this is typically where NuSA will be applied. It is important to note that NuSA can only be applied to layers that reduce dimensionality as discussed in Section 2.2. NuSA operates by computing the magnitudes of the input feature, $\boldsymbol{x}$, and the projection of the input feature into the null space, $\mathcal{N}(\boldsymbol{W}_h)$. For OOD we seek to minimize this ratio, thus NuSA is defined as:

$$\mathcal{N}_u\mathcal{S}\mathcal{A} = \frac{\|\mathcal{N}(\boldsymbol{W})\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2}. \tag{6}$$

From this equation we can then demonstrate the impact NuSA has during training visually. figure 2 illustrates how this loss function can manipulate the weights during training. Since NuSA sets out

to minimize the null space projection, and we cannot change the input vector $\boldsymbol{x}$ the only option is to move the row/null spaces.

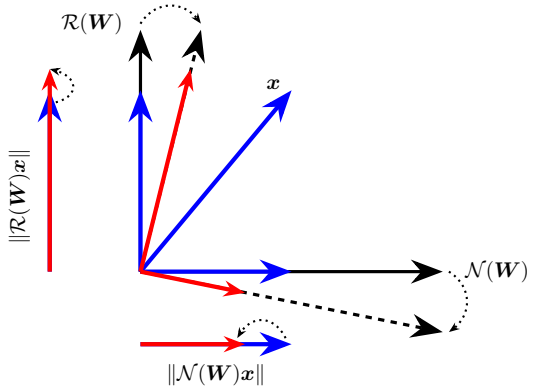


Figure 2: Diagram showing how comparing the length of the input data and the length of the null space projection can be effectively used to control how much of the data is lost to the null space. NuSA does not directly impact the sample and instead only impacts the row and null spaces. Therefore, in space the sample $\boldsymbol{x}$ does not change but the axis ($\mathcal{R}(\boldsymbol{W})$ and $\mathcal{N}(\boldsymbol{W})$) move around the input sample.

3.3 Application of NuSA

With equation 6 we have defined a method to compute the magnitude of the null space projections, in this section we describe how to implement this within an ANN to find outliers. We can encode the outlier detection into a network by directly incorporating the NuSA calculation into the loss function during training. To do this, equation 6 is added to the loss function as

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{x}) + \lambda \frac{\|\mathcal{N}(\boldsymbol{W})\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2}, \tag{7}$$

where λ is a tuning parameter that controls the trade-off between the traditional loss function $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{x})$ and the proposed NuSA term. During optimization $\mathcal{N}(\boldsymbol{W})$ is computed each time NuSA is evaluated. This allows the optimization to always use the correct null space at the expense of computing an SVD as part of every function evaluation. An important note is that this method performs outlier detection in an unsupervised fashion.

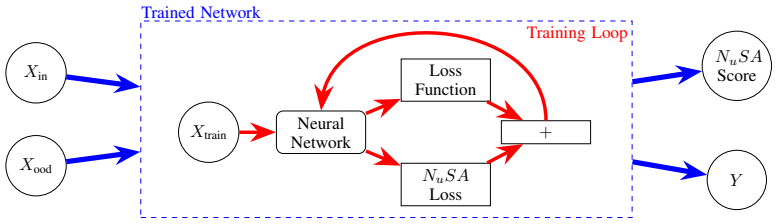


Figure 3: NuSA training and testing pipeline. The red arrows show the training flow, this follows the standard backprop training with the addition of the NuSA loss function. The blue arrows show the testing flow, where testing data, X_{in}, and OOD data, X_{ood} are passed into the trained network and the network returns the classification output, Y, along with the NuSA Score.

Once training has been completed the NuSA scores can be used to identify OOD data. The ANN will function normally except that there is an output. In the NuSA ANN model the classifier model still exists but the NuSA score can be seen as the "none of the above" class. The NuSA score is small when the testing data is similar to the training data and large when a disparity exists. Therefore, a threshold is applied to the NuSA score and large values are eliminated as outliers. To summarize, the pipeline for using NuSA is shown in figure 3.

4 EXPERIMENTS

We investigate the performance of NuSA using five small image datasets, CIFAR10Krizhevsky (2012), SVHNNetzer et al. (2011), iSUNXu et al. (2015), LSUNYu et al. (2015), and tiny Image net Deng et al. (2009). For datasets not originally 32×32, the images were resized to the correct resolution as in previous worksDeVries & Taylor (2018); Liang et al. (2017).[1] Of these datasets both CIFAR10 and SVHN are used as the known training datasets with the remaining four datasets used as out-of-distribution data. All images are normalized to have mean $[0.4914, 0.4822, 0.4465]$ and standard deviation $[0.2470, 0.2435, 0.2616]$. For CIFAR10 training, the data augmentations RandomCrop, crop size of 32 and buffer of 4, and random horizontal flips are used. For SVHN training, only the RandomResizedCrop is used with crop size of (32×32), scaling between 0.9 and 1.1, and ratio between 0.95 and 1.05.

For consistency three WideResNet models are trained for each method, with random initializations. Common parameters across each experiment are the maximum number of epochs, 300, weight decay, 0.01, learning rate, 0.01. An early stopper is enabled during training that stops if the validation accuracy does not increase between epochs for 15 consecutive epochs or if the best validation accuracy does not improve after 40 consecutive epochs; the minimum number of epochs allowed is 150. The early stopper is a slightly modified version of the early stopper available online here.[2] For optimization, Adam Kingma & Ba (2014) is used with cosine annealing applied to the learning rate. For comparison we use Regularized ArcFace (RegArcFace) Kim & Lee (2025), Rectified activation functions (ReAct) Sun et al. (2021), Self-supervised outlier detection (SSD) Sehwag et al. (2021), Out-of-distribution image detection (Odin) Liang et al. (2017), and a baseline method from Hendrycks & Gimpel (2016). For all comparison methods we set tuning parameters to default values. For NuSA, λ is the only tunable parameter and it is set to 1.0.

Upon the completion of the training routine, the known test data is concatenated individually with each of the remaining four datasets. To test OOD performance, we create a binary labeling scheme where the ood data is labeled as ones, and the known test data is labeled with zeros to generate the comparison metrics.

Table 1: False positive rate at 95% detection rate for all experiments the known dataset is listed at the top of each column set.

	Trained with CIFAR10				Trained with SVHN			
	SVHN	iSUN	LSUN	INet	CIFAR10	iSUN	LSUN	INet
NuSA	14.31	10.62	13.19	6.46	18.64	12.82	12.28	14.95
RegArcFace	31.68	47.08	47.86	49.19	23.39	12.40	16.84	14.95
ReAct	52.22	43.43	36.73	39.11	50.48	47.05	46.09	46.03
SSD	32.31	31.84	30.60	46.00	39.67	29.67	30.40	32.61
Odin	65.74	59.55	54.57	76.62	61.70	58.33	57.81	57.30
Baseline	51.26	44.80	41.01	88.14	20.73	18.50	18.93	21.97

Table 2: Area under the ROC curve for all experiments the known dataset is listed at the top of each column set.

	Trained with CIFAR10				Trained with SVHN			
	SVHN	iSUN	LSUN	INet	CIFAR10	iSUN	LSUN	INet
NuSA	0.97	0.98	0.97	0.99	0.96	0.97	0.97	0.97
RegArcFace	0.77	0.73	0.73	0.72	0.95	0.96	0.95	0.96
ReAct	0.89	0.92	0.93	0.93	0.90	0.91	0.92	0.92
SSD	0.92	0.93	0.94	0.90	0.94	0.95	0.95	0.95
Odin	0.86	0.89	0.90	0.84	0.87	0.89	0.89	0.89
Baseline	0.91	0.93	0.94	0.85	0.96	0.97	0.97	0.96

An extensive summary of all experiments is shown in Tables 1, 2, and 3. The statistics shown are defined as:

[1] Datasets available via link at: `https://github.com/ShiyuLiang/odin-pytorch`

[2] `https://github.com/Bjarten/early-stopping-pytorch`

Table 3: Detection Error, defined above, for all experiments the known dataset is listed at the top of each column set.

	Trained with CIFAR10				Trained with SVHN			
	SVHN	iSUN	LSUN	INet	CIFAR10	iSUN	LSUN	INet
NuSA	8.70	7.21	7.79	5.68	9.20	7.33	7.30	8.03
RegArcFace	18.13	23.94	23.75	24.82	9.86	8.37	9.24	8.75
ReAct	19.01	16.03	14.24	13.65	16.99	15.67	15.53	15.38
SSD	15.13	14.71	13.80	17.06	12.26	10.05	10.23	10.51
Odin	21.72	18.70	17.09	21.92	19.82	18.45	18.28	17.96
Baseline	14.87	13.38	12.31	18.62	9.11	8.44	8.51	9.23

- FPR at 95% TPR - Percent of outliers incorrectly identified as inliers when 95% of the inliers are correctly identified as inliers.
- AUROC - Area under the OOD ROC curve.
- Detection Error - Minimum possible misclassification probability of out-of-distribution examples over all thresholds, δ. Defined as $\min_\delta 0.5P_{in}(f(x) \leq \delta) + 0.5P_{out}(f(x) > \delta)$.

From the tables we can conclude that NuSA is an improvement over the comparison methods over the datasets tested for OOD. The performance gain is best seen through the FPR at 95% statistic which measures performance at very low false alarm rates. Here the NuSA method presents a very significant improvement across the board when compared to the previous methods. The downside to the NuSA method is that it does have a small impact on the classification accuracy, however in our experimentation we saw less than a 1% reduction in accuracy. It is possible though that with different training methods, similar to what the Confidence Estimation uses, or other improvements that this classification performance could be regained.

4.1 Classification Accuracy vs NuSA Score

Beyond looking at OOD performance in a vacuum we also look at the classification accuracy in relation to the NuSA score. When the NuSA score is small we expect all the data to be inliers. Therefore, we compute NuSA scores for known data to validate our assumption. We expect that in the known test data the smallest NuSA scores should indicate the data that most resembles the training data; therefore the network should predict with the highest accuracy.

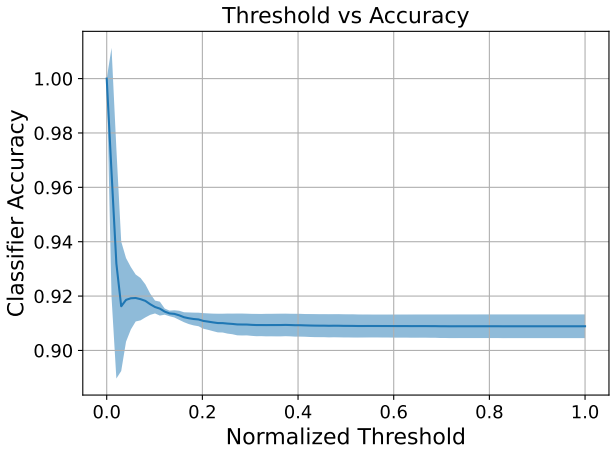


Figure 4: Mean classification accuracy vs the NuSA score. A smaller NuSA score indicates the data is more typical of the known training data and therefore better performance is expected for low NuSA scores. The shaded regions indicates the standard deviation of the five experiments.

We input blind test data to networks trained using CIFAR10 and plot the classification accuracy of samples that have NuSA scores below a selected threshold. The threshold is varied from the smallest

value of NuSA to the largest. For plotting purposes this scale has been modified to fall between zero and one. figure 4 shows the resulting Threshold vs Accuracy plot. The line is the mean of three network outputs and the shaded region corresponds to the standard deviation of the three outputs. As one would expect, as the threshold decreases the accuracy over the remaining data increases. This shows that smaller NuSA scores indicate data that is more representative of the training data as demonstrated by the overall higher accuracy.

4.2 Impacts on Speed/Efficiency

After investigating the performance of NuSA it is also important to observe any effects on the speed to train/process data. There will be a slowdown due to the SVD used to compute the null space during training/evaluation. To evaluate how impactful this is we set up an experiment where a wide resnet is trained for five epochs and the time it takes to complete the training is recorded. This experiment is repeated three times each for a baseline network without NuSA and for a network with NuSA. In this experiment the average run time for the base network was 148.73s with a standard deviation of 0.06s while NuSA's average run time is 149.53s with a standard deviation of 0.46s.[3] As expected NuSA does increase the training time, but only by approximately 3%.

5 Conclusion

In this paper, we introduce Null Space Analysis (NuSA), a novel technique for training neural networks to detect out-of-distribution (OOD) data. NuSA controls how the existing architecture utilizes the null space of the fully connected classification layer. Because the null space projections are inherently linked to the network's weights, NuSA modifies the learning process without requiring any changes to the underlying network architecture. Our experiments demonstrate that while NuSA incurs a minor penalty to standard classification accuracy, it significantly improves OOD detection performance—particularly in reducing the false positive rate. For future work, we plan to investigate the application of NuSA to convolutional and self-attention layers.

AI use statement

No generative AI tools were using for this work.

References

Revisiting One-vs-All Classifiers for Predictive Uncertainty and Out-of-Distribution Detection in Neural Networks, volume abs/2007.05134, 2020. URL https://arxiv.org/abs/2007.05134.

Mohamed H. Abdelpakey and Mohamed S. Shehata. Nullspacenet: Nullspace convoluional neural network with differentiable loss function. *CoRR*, abs/2004.12058, 2020. URL https://arxiv.org/abs/2004.12058.

Charu C. Aggarwal. *Outlier Analysis*. Springer, 2015.

R. Bansal, N. Gaur, and S. N. Singh. Outlier detection: Applications and techniques in data mining. In *2016 6th International Conference - Cloud System and Big Data Engineering (Confluence)*, pp. 373–377, 2016.

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

[3]Timing is computed on a Linux Desktop with an Intel i7-13700K, 32GB of RAM, and a single NVIDIA RTX 4080.

Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. Lof: Identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, SIGMOD '00, pp. 93–104, New York, NY, USA, 2000. Association for Computing Machinery. ISBN 1581132174. doi: 10.1145/342009.335388. URL https://doi.org/10.1145/342009.335388.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR09*, 2009.

Terrance DeVries and Graham W. Taylor. Learning confidence for out-of-distribution detection in neural networks, 2018. URL https://arxiv.org/abs/1802.04865.

G. Dong and Ruey-Wen Liu. An orthogonal learning rule for null-space tracking with implementation to blind two-channel identification. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 45(1):26–33, 1998. doi: 10.1109/81.660748.

Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Analysis of classifiers' robustness to adversarial perturbations. *Machine Learn*, 107(3):481–508, Mar 2018. ISSN 0885-6125.

Shelly D. D. Goggin, Karl E. Gustafson, and Kristina M. Johnson. Accessing the null space with nonlinear multilayer neural networks. In Dennis W. Ruck (ed.), *Science of Artificial Neural Networks*, volume 1710, pp. 308 – 316. International Society for Optics and Photonics, SPIE, 1992. doi: 10.1117/12.140097. URL https://doi.org/10.1117/12.140097.

Markus Goldstein and Andreas Dengel. Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. 09 2012.

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv 1412.6572*, 12 2014.

A. Hammerstein. Nichtlineare Integralgleichungen nebst Anwendungen. *Acta Mathematica*, 54 (none):117 – 176, 1930. doi: 10.1007/BF02547519. URL https://doi.org/10.1007/BF02547519.

Jim Hefferon. *Linear Algebra*, chapter Maps Between Spaces, pp. 173–319. Textbook Equity, Feb 2012.

Matthias Hein, Maksym Andriushchenko, and Julian Bitterwolf. Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 41–50, 2019. doi: 10.1109/CVPR.2019.00013.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. 2016. doi: 10.48550/ARXIV.1610.02136. URL https://arxiv.org/abs/1610.02136.

Dan Hendrycks, Mantas Mazeika, and Thomas G. Dietterich. Deep anomaly detection with outlier exposure. *CoRR*, abs/1812.04606, 2018. URL http://arxiv.org/abs/1812.04606.

Daksh Idnani, Vivek Madan, Naman Goyal, David J. Schwab, and Ramakrishna Vedantam. Don't forget the nullspace! nullspace occupancy as a mechanism for out of distribution failure. In *International Conference on Learning Representations*, 2023. URL https://api.semanticscholar.org/CorpusID:259298625.

Andrzej Janczak. *Identification on Nonlinear Systems Using Neural Networks and Polynomial Models*, chapter Neural network Hammerstein models, pp. 77–116. Springer, Berlin, Heidelberg, 2005. ISBN 9783540231851.

Muhammad Ghufran Janjua, Kevin Kaiser, Thomas Stadelmayer, Stephan Schoenfeldt, and Vadim Issakov. Jedan: Joint euclidean distance and autoencoder network for robust out-of-distribution detection in radar-based hand gesture recognition. *IEEE Access*, 12:196364–196381, 2024. doi: 10.1109/ACCESS.2024.3520810.

Taewook Kim and Jong-Seok Lee. Gently sloped and extended classification margin for overconfidence relaxation of out-of-distribution samples. *IEEE Transactions on Neural Networks and Learning Systems*, 36(7):12854–12865, 2025. doi: 10.1109/TNNLS.2024.3496473.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2013. URL https://arxiv.org/abs/1312.6114.

Hans-Peter Kriegel, Matthias Schubert, and Arthur Zimek. Angle-based outlier detection in high-dimensional data. pp. 444–452, 08 2008. doi: 10.1145/1401890.1401946.

Alex Krizhevsky. Learning multiple layers of features from tiny images. *University of Toronto*, 05 2012.

Shuangqun Li, Xinchen Liu, Wu Liu, Huadong Ma, and Haitao Zhang. A discriminative null space based deep learning approach for person re-identification. In *2016 4th International Conference on Cloud Computing and Intelligence Systems (CCIS)*, pp. 480–484, 2016. doi: 10.1109/CCIS.2016.7790306.

Xuan Li, Christian Desrosiers, and Xue Liu. Symmetric contrastive loss for out-of-distribution skin lesion detection. In *2022 IEEE 19th International Symposium on Biomedical Imaging (ISBI)*, pp. 1–5, 2022. doi: 10.1109/ISBI52829.2022.9761434.

Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2017.

F. T. Liu, K. M. Ting, and Z. Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pp. 413–422, 2008.

Juncheng Liu, Zhouhui Lian, Yi Wang, and Jianguo Xiao. Incremental kernel null space discriminant analysis for novelty detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4123–4131, 2017. doi: 10.1109/CVPR.2017.439.

Srinivas Sourabhreddy Sattireddy Medapati. Competency aware machine learning using null space avoidance. M.s. thesis, University of Florida, 2019.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. *CoRR*, abs/1511.04599, 2015. URL http://arxiv.org/abs/1511.04599.

Ryo Nakamura, Ryu Tadokoro, Eisuke Yamagata, Yusuke Kondo, Kensho Hara, Hirokatsu Kataoka, and Nakamasa Inoue. Pseudo-outlier synthesis using q-gaussian distributions for out-of-distribution detection. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3120–3124, 2024. doi: 10.1109/ICASSP48485.2024.10448313.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Ng. Reading digits in natural images with unsupervised feature learning. *NIPS*, 01 2011.

Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. volume 29, pp. 427–438, 06 2000. doi: 10.1145/335191.335437.

Mohammadreza Salehi, Hossein Mirzaei, Dan Hendrycks, Yixuan Li, Mohammad Hossein Rohban, and Mohammad Sabokrou. A unified survey on anomaly, novelty, open-set, and out-of-distribution detection: Solutions and future challenges. *CoRR*, abs/2110.14051, 2021. URL https://arxiv.org/abs/2110.14051.

Bernhard Schölkopf, John Platt, John Shawe-Taylor, Alexander Smola, and Robert Williamson. Estimating support of a high-dimensional distribution. *Neural Computation*, 13:1443–1471, 07 2001. doi: 10.1162/089976601750264965.

Vikash Sehwag, Mung Chiang, and Prateek Mittal. Ssd: A unified framework for self-supervised outlier detection. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=v5gjXpmR8J.

Mei-Ling Shyu, Shu-Ching Chen, Kanoksri Sarinnapakorn, and Liwu Chang. A novel anomaly detection scheme based on principal component classifier. *Proceedings of International Conference on Data Mining*, 01 2003.

Yiyou Sun, Chuan Guo, and Yixuan Li. React: Out-of-distribution detection with rectified activations. In *Advances in Neural Information Processing Systems*, 2021.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *CoRR*, abs/1312.6199, 2013. URL http://arxiv.org/abs/1312.6199.

Qiaozhi Tan, Long Bai, Guankun Wang, Mobarakol Islam, and Hongliang Ren. Endoood: Uncertainty-aware out-of-distribution detection in capsule endoscopy diagnosis. In *2024 IEEE International Symposium on Biomedical Imaging (ISBI)*, pp. 1–5, 2024. doi: 10.1109/ISBI56570.2024.10635759.

Jie Wei and Guotai Wang. Fine-grained out-of-distribution detection of medical images using combination of feature uncertainty and mahalanobis distance. In *2023 IEEE 20th International Symposium on Biomedical Imaging (ISBI)*, pp. 1–5, 2023. doi: 10.1109/ISBI53787.2023.10230433.

Pingmei Xu, Krista A Ehinger, Yinda Zhang, Adam Finkelstein, Sanjeev R Kulkarni, and Jianxiong Xiao. Turkergaze: Crowdsourcing saliency with webcam based eye tracking. *arXiv preprint arXiv:1504.06755*, 2015.

Yinyu Ye, Shijing Chen, Dong Ni, and Ruobing Huang. Triaug: Out-of-distribution detection for imbalanced breast lesion in ultrasound. In *2024 IEEE International Symposium on Biomedical Imaging (ISBI)*, pp. 1–5, 2024. doi: 10.1109/ISBI56570.2024.10635468.

Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

Li Zhang, Tao Xiang, and Shaogang Gong. Learning a discriminative null space for person re-identification. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1239–1248, 2016. doi: 10.1109/CVPR.2016.139.

Zhilin Zhao, Longbing Cao, and Kun-Yu Lin. Out-of-distribution detection by cross-class vicinity distribution of in-distribution data. *IEEE Transactions on Neural Networks and Learning Systems*, 35(10):13777–13788, 2024. doi: 10.1109/TNNLS.2023.3271856.