Hyperdimensional Feature Fusion for Out-of-Distribution Detection

Samuel Wilson
Queensland University of Technology
2 George St, Brisbane, QLD 4000, Australia
s84.wilson@hdr.qut.edu.au

Tobias Fischer
Queensland University of Technology
2 George St, Brisbane, QLD 4000, Australia
tobias.fischer@qut.edu.au

Niko Sünderhauf
Queensland University of Technology
2 George St, Brisbane, QLD 4000, Australia
niko.suenderhauf@qut.edu.au

Feras Dayoub
University of Adelaide
North Terrace, Adelaide, SA 5005, Australia
feras.dayoub@adelaide.edu.au

Abstract

We introduce powerful ideas from Hyperdimensional Computing into the challenging field of Out-of-Distribution (OOD) detection. In contrast to most existing works that perform OOD detection based on only a single layer of a neural network, we use similarity-preserving semi-orthogonal projection matrices to project the feature maps from multiple *layers into a common vector space. By repeatedly applying the* bundling *operation* $\oplus$*, we create expressive class-specific descriptor vectors for all in-distribution classes. At test time, a simple and efficient cosine similarity calculation between descriptor vectors consistently identifies OOD samples with competitive performance to the current state-of-the-art whilst being significantly faster. We show that our method is orthogonal to recent state-of-the-art OOD detectors and can be combined with them to further improve upon the performance.*

1. Introduction

Deep Neural Networks can achieve excellent performance on many vision tasks when the distribution of training data closely matches the test data. However, this assumption is often violated during deployment: especially in embodied AI application domains such as robotics and autonomous systems, objects and scenes that were not part of the training data distribution will inevitably be encountered. When confronted with such out-of-distribution (OOD) samples, deep neural networks tend to fail silently, producing overconfident but erroneous predictions [17, 49] that can pose severe risks [55]. It is therefore critically important that OOD samples are identified effectively during deployment. Previous techniques for OOD detection used softmax probabilities [19, 29] or distances in logit space [18] to distinguish in-distribution (ID) from OOD samples, making the implicit assumption that the features of a *single* layer in a Deep Neural Network (DNN) contain sufficient information to identify OOD data. However, deeper layers of a DNN can be more sensitive to OOD samples than the softmax probabilities [44], which are often overconfident in the presence of OOD samples [17]. Recent work investigating the utility of multi-scale deep features from a DNN found that modelling ID classes from raw features is difficult without first reducing the number of dimensions [44, 28, 42] due to the curse of dimensionality [61].

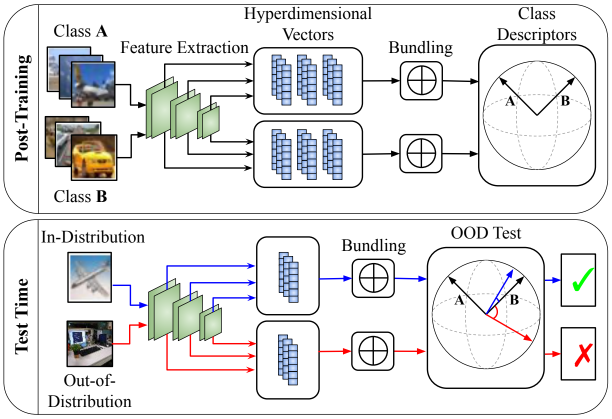

Figure 1. Illustration of our HDFF OOD detector. **Top:** After training, we extract the feature maps of in-distribution classes from multiple layers of the DNN and project them into a new hyperdimensional space with a similarity-preserving projection matrix. Using the bundling operation $\oplus$ we create class-specific hyperdimensional descriptor vectors. **Bottom:** During testing, we repeat feature extraction, projection and bundling to obtain a hyperdimensional image descriptor. OOD samples will produce descriptors with a large angular distance to all class descriptors (red vector).

In this paper, we introduce the powerful concepts of similarity-preserving projection, binding and bundling from Hyper Dimensional Computing (HDC) [53] to OOD detection. Figure 1 provides a high-level description of our proposed OOD detector. We project the feature maps from *multiple* layers of a network into a common high-dimensional vector space, using similarity-preserving semi-orthogonal projection matrices. This allows us to effectively *fuse* the information contained in multiple layers: a series of *bundling* operations $\oplus$ creates a class-specific high-dimensional vector representation for each of the in-distribution classes from the training dataset. During testing, we use the same principles of projection and bundling to obtain a single vector representation for a new input image. OOD samples can then be efficiently identified through cosine similarity operations with the class-specific representations.

Our Hyperdimensional Feature Fusion (HDFF) approach relies on the pseudo-random nature of its projection matrices and the similarity-preserving properties of the bundling operation $\oplus$ to effectively fuse information contained in the feature maps across multiple layers in a deep network. HDFF avoids the difficulties of estimating densities in high-dimensional spaces [44, 28], removes the limitation of relying on a single layer [19, 18], does not require to select informative layers based on example OOD data [22, 58, 31], and does not necessarily rely on expensive sampling techniques (such as ensembles [26] or Monte Carlo Dropout [11]) to identify OOD samples. HDFF can be applied to a trained network and does not require re-training or fine-tuning. Our HDFF detector competes with or exceeds state-of-the-art performance for OOD detection in conjunction with networks trained with standard cross-entropy loss or more sophisticated logit space shaping approaches [59].
In summary, our contributions are the following:

1. We propose a novel OOD detector based on the principles of Hyperdimensional Computing that is competitive with state-of-the-art OOD detectors whilst requiring significantly less training and inference time.
2. We show that fusing features from multiple layers is critical for general OOD detection performance, as different layers vary in sensitivity to different OOD data, but fusion results in OOD performance that is approximately on par or even better than the best single layer.
3. We show that the angles between HDFF vectors can be interpreted as visual similarity between inputs allowing for failure diagnosis.

We release a publicly available code repository to replicate our experiments at: `https://github.com/SamWilso/HDFF_Official`

2. Related Work

Out-of-distribution (OOD) detection has been addressed in the context of image classification [29, 31, 59, 19] and semantic segmentation [18, 9, 1, 2] with a diverse set of solutions to each of the problems. In this section, we highlight overarching concepts and most relevant contributions in three areas: i) Many approaches try to extrapolate the properties of the distribution of OOD data by utilising a small subset of the data to fine-tune on. ii) When this information is unavailable (OOD agnostic), distances or densities in the softmax probabilities and logit spaces are used. iii) Comparatively few bodies of work exist that use multiple feature maps for OOD detection.

OOD Fine-tuning Some prior work [48, 22, 58, 31, 29, 1] requires the availability of an OOD dataset to train or fine-tune the network or adjust the OOD detection process. However, such approaches are problematic. By definition, the OOD dataset has to always be incomplete: during deployment, samples that do not match the assumptions embedded in the OOD dataset can still be encountered, leaving such methods exposed to high OOD risk. In contrast, we do not require prior knowledge about the characteristics of OOD data, instead we directly model the in-distribution data and measure deviations from this set.

OOD Agnostic Most approaches that do not rely on the availability of OOD data during training or validation analyse the model outputs to identify OOD data during deployment. Early work [19] showed the maximum of a model's softmax output to be a good baseline. Expanding on this idea, [18] proposed using the maximum of the unnormalised output logits. Additional methods focus on extracting more information from the outputs [41, 54] or learning calibrated confidence estimates [7, 27]. Model averaging techniques such as Deep Ensembles [26], Monte Carlo (MC) Dropout [11, 18] and SWA-G [32, 59] often improve the OOD detection performance. Training networks with specialised losses that impose a beneficial structure on the feature space has been growing in popularity [34, 59, 35, 21, 8]. E.g. the recent 1D Subspaces [59] OOD estimator encourages the network to learn orthogonal features in the layer before the logits. The approaches discussed above attempt OOD detection based on only a *single* layer in the network. In contrast, we show that utilising *multiple* layers is beneficial for OOD detection.

Multi-Feature OOD Detection Newer approaches incorporate and analyse multi-scale feature maps extracted from a DNN to detect OOD samples. The use of feature maps and additional information extracted from a network has been shown to set a new state-of-the-art in both OOD segmentation [9, 1] and classification [10] when used as input to an auxiliary OOD detection network. The requirement on an auxiliary network necessarily entails training on OOD, or proxy-OOD, data that is strongly representative of the OOD samples likely to be drawn from target distribution. Related work with auxiliary networks in online performance monitoring demonstrates that this class of networks can be robust to failures of the primary network without prior knowledge of the exact failures that are likely to be encountered [46, 47]. Avoiding these problems altogether, recent works have tried to model in-distribution classes without an auxiliary network [44, 28, 42]. However, the curse of dimensionality requires these methods to rely on PCA reductions [44, 28] or random projections [42], losing information from the feature maps. By contrast, we use techniques from Hyperdimensional Computing to model our in-distribution data without the need for an auxiliary network. We seek to build upon evidence in the literature [44, 50, 30, 10] that multi-scale features benefit OOD detection performance.

3. Hyperdimensional Computing

Hyperdimensional Computing (HDC), also known as Vector Symbolic Architectures, is the field of computation with vectors in hyperdimensional, very large, spaces. HDC techniques leverage the large amount of expressive power in the hyperdimensional (HD) space to model associations between data points using redundant HD vector encodings. Redundant means that in spaces of 10^4 dimensionality it can be seen that two random vectors are almost guaranteed to be within 5 degrees of orthogonal [52] known as *quasi-orthogonality*. The associations between data points are represented as vectors, and the set of the representative vectors is known as the *associative memory* [14]. Construction of the associative memory is done through the structured combination of HD vectors using a set of standard operations. For our application we are primarily interested in the *bundling* $\oplus$ and *binding* $\otimes$ operation for combining feature vectors, and the *encoding* operation for projecting low dimensional representations into our target HD space.

Bundling The bundling $\oplus$ operation is used to store a representation of multiple input vectors that retains similarity to all of the input vectors [43]. Concretely, given random vectors $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c}$, the vector $\mathbf{a}$ will be similar to the bundles $\mathbf{a} \oplus \mathbf{b}$, $\mathbf{a} \oplus \mathbf{c}$ and $\mathbf{a} \oplus \mathbf{b} \oplus \mathbf{c}$, although in the final case it will be less similar as the bundled vector needs to be similar to all 3 input vectors. Typically, the bundling operation is implemented as element-wise addition with some form of normalisation to the required space for the architecture [52]. Normalisation steps that are commonly seen are normalisation to a magnitude of one [15, 12] or cutting/thresholding to a range of desired values [13]. Without normalisation, the bundling operation is commutative and associative; when normalisation steps are added the associative aspect of the operation is approximate, $(\mathbf{a} \oplus \mathbf{b}) \oplus \mathbf{c} \approx \mathbf{a} \oplus (\mathbf{b} \oplus \mathbf{c})$ [52].

Binding The binding $\otimes$ operation is used to combine a set of vectors into one representation that is dissimilar to all of the input vectors [43]. Given the quasi-orthogonality property of HD spaces, this entails that in all likelihood, the binding operation will generate a vector orthogonal to all of the input vectors in the cosine similarity space. The second core trait of the binding operation is that it approximately preserves the similarity of two vectors before and after binding if they are bound to the same target vector. More precisely, for a set of vectors in the same hyperdimensional space $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c}$, that $sim(\mathbf{a}, \mathbf{b}) \approx sim(\mathbf{a} \otimes \mathbf{c}, \mathbf{b} \otimes \mathbf{c})$.

Encoding The format of the original data will not always be in the hyperdimensional space that is required. Encoding addresses this by projecting each data point from the original space into the new HD space. The selection of data encoding into a HD space depends on the type of the original data [45, 20]. An important principle of encoding is that distances in the input are preserved in the output [20], much like the binding operation. Examples of encoding include multiplication with a projection matrix [36], fractional binding [24] that preserves real numbered differences, or the encoding proposed in [20] that preserves similarities of vectors over time and spatial coordinates.

We provided only a brief introduction to HDC, for more in-depth discussions and comparisons between known architectures we encourage the reader to consider [52, 39, 23].

4. Hyperdimensional Feature Fusion

We propose Hyperdimensional Feature Fusion (HDFF), a novel OOD detection method that applies the HDC concepts of Encoding and Bundling to the features from *multiple* layers in a deep neural network, without requiring re-training or any prior knowledge of the OOD data. Our core idea is to project the feature maps from multiple layers into a common vector space, using similarity-preserving semi-orthogonal projection matrices. Through a series of bundling operations, we create a class-specific vector-shaped representation for each of the classes in the training dataset. During deployment, we repeat the projection and bundling steps for a new input image, and use the cosine similarity to the class representatives to identify OOD samples. We provide pseudo code to assist with describing our HDFF detector in Algorithm 1.

Preliminaries From a pretrained network f, we can extract feature maps $\mathbf{m}_l$ from different layers $l \in \{1, \ldots, L\}$ in the the network. We apply a pooling operation across the spatial dimensions to reduce the tensor-shaped (height × width × channels) feature maps to vector representations $\mathbf{v}_l$ of length (channels) for each layer; we ablate the effects of different pooling in the Supplementary Material. Since the feature maps of different layers have a different number of channels c_l, we need to project the vectors $\mathbf{v}_l$ into a common m-dimensional vector space $\mathbb{R}^m$ in order to combine them. Conventional sizes for HD spaces range around 10^3 - 10^4 [40, 36, 20], meaning that typically our HD space will be much larger than our original space $m >> c$.

Feature Encoding Any $m \times c$ matrix $\mathbf{P}$ projects a c-dimensional vector $\mathbf{v}$ into a m-dimensional space by left-multiplying: $\mathbf{P}\mathbf{v}$. However, preserving the cosine similarity of the projected vectors is a crucial consideration for our

Algorithm 1 Computation of the model-wise class descriptors and ensemble descriptor vectors from training set.

```
Inputs: Set of images from the training set $x^{\{1,\ldots,I\}}$ with corresponding labels $j^{\{1,\ldots,I\}}$. Set of models in an ensemble $f_{\{1,\ldots,E\}}$ that produces a set of feature maps $\mathbf{m}_{\{1,\ldots,L\}}$ given an input $x^{(i)}$.
Outputs: Set of class descriptor vectors per input model $d^{(e)}_{\{1,\ldots,c\}}$ and a set of ensemble class descriptor vectors $d^*$
for e $\in$ {1,...,E} do
    for i $\in$ {1,...,I} do
        $m^{(i)}_{\{1,\ldots,L\}} \leftarrow f_e(x^{(i)})$
        for l $\in$ {1,...,L} do
            $v^{(i)}_l \leftarrow Pool(m^{(i)}_l)$
            $h^{(i)}_l \leftarrow P_l \cdot v^{(i)}_l$
        end for
        $y^{(i)} \leftarrow h^{(i)}_1 \oplus h^{(i)}_2 \oplus ... \oplus h^{(i)}_L$
        $d^{(e)}_{j^{(i)}} \leftarrow d^{(e)}_{j^{(i)}} \oplus y^{(i)}$
    end for
end for
$d^* \leftarrow (d^{(1)} \otimes z^{(1)}) \oplus ... \oplus (d^{(E)} \otimes z^{(E)})$
```

application in OOD detection. We therefore impose that the projection matrices $\mathbf{P}_l$ preserve the inner product of any two vectors $\mathbf{a}$ and $\mathbf{b}$ in their original and their projected vector spaces. Formally, this means that

$$(\mathbf{P} \cdot \mathbf{a})^\mathsf{T} \cdot (\mathbf{P} \cdot \mathbf{b}) = \mathbf{a}^\mathsf{T} \cdot \mathbf{b} \quad (1)$$

$$\mathbf{a}^\mathsf{T} \cdot \mathbf{P}^\mathsf{T} \cdot \mathbf{P} \cdot \mathbf{b} = \mathbf{a}^\mathsf{T} \cdot \mathbf{b} \quad (2)$$

Above condition is satisfied by any matrix that fulfils the requirement $\mathbf{P}^\mathsf{T}\mathbf{P} = \mathbf{I}$, which is the defining property of a semi-orthogonal matrix. Using the approach of [51], we therefore create a unique, pseudo-random, semi-orthogonal projection matrix $\mathbf{P}_l$ for each of the considered layers l. These project the feature vectors $\mathbf{v}_l$ into a common m-dimensional vector space:

$$\mathbf{h}_l = \mathbf{P}_l \mathbf{v}_l \quad \text{so that} \quad \mathbf{h}_l \in \mathbb{R}^m \quad \forall l \in \{1, \ldots, L\} \quad (3)$$

Feature Bundling Following the previous steps, we obtain a set of L high-dimensional vectors $\mathbf{h}^{(i)}_l$ for an input image $\mathbf{x}^{(i)}$. Since all $\mathbf{h}^{(i)}_l$ are elements of the same vector space, we can use the bundling operation $\oplus$ to combine them into a single vector $\mathbf{y}^{(i)}$ that serves as an expressive descriptor for the input image $\mathbf{x}^{(i)}$:

$$\mathbf{y}^{(i)} = \bigoplus_{l=1}^{L} \mathbf{h}^{(i)}_l = \bigoplus_{l=1}^{L} \mathbf{P}_l \cdot \mathbf{v}^{(i)}_l \quad (4)$$

As explained in Section 3, the resulting vector $\mathbf{y}^{(i)}$ will be cosine-similar to all contributing vectors $\mathbf{h}^{(i)}_l$, but dissimilar to vectors from $\mathbb{R}^m$ that were not part of the bundle. Essentially, $\mathbf{y}^{(i)}$ provides a summary of the feature vectors $\mathbf{v}^{(i)}_l$ of the entire network for a single image $\mathbf{x}^{(i)}$.

By bundling the descriptors of all images from the training set belonging to class c, we obtain a class-specific descriptor $\mathbf{d}_c$:

$$\mathbf{d}_c = \bigoplus_{j \in \mathbb{I}_c} \mathbf{y}^{(j)} \quad (5)$$

where $\mathbb{I}_c$ denotes the set of indices of the training images belonging to class c.

As discussed in Section 3, the bundling operation $\oplus$ can be implemented in various ways. We implement $\oplus$ to be an element-wise sum, without truncation.

Out-of-Distribution Detection During testing or deployment, an image $\mathbf{x}$ can be identified as OOD by obtaining its image descriptor $\mathbf{y}$ according to (4), and calculating the cosine similarity to each of the class-specific descriptors $\mathbf{d}_c$. Let θ be the angle to the class descriptor $\mathbf{d}_c$ that is most similar to $\mathbf{y}$

$$\theta = \min_c \cos^{-1}\left(\frac{\mathbf{y}^\mathsf{T}\mathbf{d}_c}{\|\mathbf{y}\| \cdot \|\mathbf{d}_c\|}\right), \; c \in \{1, \ldots, C\}, \quad (6)$$

The input $\mathbf{x}$ is then treated as OOD if θ is bigger than a threshold: $\theta > \theta^*$.

Ensembling While HDFF does not rely on ensembling, we briefly show that our method is amenable to ensembling to further boost performance (however at the cost of added compute). When using a set of pretrained networks f_e in an ensemble $e \in \{1, \ldots, E\}$ we collect inputs from all models and fuse them into singular image and class descriptors $\mathbf{y}_*$ and $\mathbf{d}^*$ respectively. For each model $f_{\{1,\ldots,E\}}$ the same process using equations (4) and (5) is used to compute the set of class descriptors for each model in the ensemble $\mathbf{d}^{\{1,\ldots,E\}}$. To ensure that each class descriptor is sufficiently distinct from all other descriptors, a set of random hyperdimensional vectors $\mathbf{z}^{(e)}$ are generated and bound $\otimes$ to the class descriptor. By bundling the bound class descriptors, we obtain the ensemble class descriptor $\mathbf{d}^*$:

$$\mathbf{d}^* = \bigoplus_{e=1}^{E} \mathbf{d}^{(e)} \otimes \mathbf{z}^{(e)} \quad (7)$$

When new input samples arrive $\mathbf{y}_*$ is computed according to (7) by substituting $\mathbf{d}$ for $\mathbf{y}$. OOD detection is done according to (6) using $\mathbf{d}^*$ and $\mathbf{y}_*$.

5. Experiments

We conduct a series of experiments to demonstrate the efficacy of Hyperdimensional Feature Fusion for OOD detection. We compare HDFF to the current state-of-the-art in the typical far-OOD settings, where the distributions of the of ID and OOD are very dissimilar (e.g. CIFAR10 → SVHN), and the more challenging near-OOD where the ID and OOD datasets are drawn from similar distributions (e.g. CIFAR10 → CIFAR100). Further, we report the results of some critical ablation studies, in particular, we identify which layers are most sensitive to OOD data and how sensitive HDFF is to the decision parameter θ^* during deployment.

5.1. Experimental Setup

Datasets For our comparisons to existing OOD detectors we use a wide array of popular datasets composed from multiple recent state-of-the-art OOD detectors. We construct our evaluation suite as the combination of datasets from [29, 50, 30]. For our comparison to existing OOD detectors, we use the popular dataset splits for near- and far-OOD detection using CIFAR10 and CIFAR100 [25] as the ID datasets. For our ID sets, we use the 50,000 training examples for training and computation of the class bundles, whilst the 10,000 testing images are used as our unseen ID data. For the near-OOD configurations, the test set of whichever CIFAR dataset is not being used for training will be used as the OOD set. For the far-OOD detection settings, we use a suite of benchmarks: iSUN, TinyImageNet [5] (cropped and resized: TINc and TINr), LSUN [57] (cropped and resized: LSUNc and LSUNr)[1], SVHN [38], MNIST [6], KMNIST [4], FashionMNIST [56], and Textures [3]. In the interest of brevity, we only show the settings where CIFAR10 is the ID set, the settings of CIFAR100 as ID are provided in the Supplementary Material.

Evaluation Metrics We consider the standard metrics [59, 50] for our comparison to existing OOD literature. **AUROC:** The Area Under The Receiver Operating Characteristic curve corresponds to the area under the ROC curve with true positive rate (TPR) on the y-axis and false positive rate (FPR) on the x-axis. AUROC can be interpreted as the probability that an OOD sample will be given a higher score than an ID sample [18]. **FPR95:** The FPR95 metric reports the FPR at a critical threshold which achieves a minimum of 95% in TPR. **Detection Error:** Detection Error indicates the minimum misclassification probability with respect to the critical threshold. **F1:** F1-score corresponds to the harmonic mean of precision and recall. We make use of F1-score to evaluate the general binarisation performance in our ablations. In the interests of brevity, we only evaluate using the AUROC metric for our evaluations. We provide additional evaluations with FPR95 and Detection Error in the Supplementary Material.

Baselines We divide our selected baselines and SOTA comparisons into two evaluation streams, these being; **statistics** and **training**. The **statistics** stream contains methods that are generally applied post-hoc to a pretrained network, requiring no training and only minimal calibration to represent in-distribution data. The **training** stream contains methods that allow for the retraining of the base network or the training of an auxiliary monitoring network. The distinction between the two streams is important for a fair comparison as the methods in the **training** stream have a significantly longer calibration (training) time as well as biases towards the OOD data, either explicitly through proxy-OOD training or implicitly through assumptions about the formation of OOD data with a custom loss.

In the **statistics** stream we compare against: maximum softmax probability [19] (MSP), max logit [18] (ML), gramian matrices [50] (Gram) and energy-based model [31] without calibration on OOD data. For the **training** stream we compare against: NMD [10], Spectral Discrepancy trained with the 1D Subspaces methodology [59] (1DS), Deep Deterministic Uncertainty [37] (DDU) and MOOD [30].

Implementation We follow the evaluation procedure defined in [59, 29], implementing the standard WideResNet [60] network with a depth of 28 and a width of 10. In the **statistics** stream, this model is trained to convergence with standard cross-entropy loss. Custom loss objectives are restricted to the **training** stream.

As HDFF is a post-hoc statistics-based method, it is innately orthogonal to the methods contained within the **training** stream and can be combined with many of them in a post-hoc fashion. To demonstrate the robustness of HDFF we combine it with two other state-of-the-art detectors, 1D Subspaces [59] and NMD [10]. The *HDFF-1DS* method applies as described in Section 4 but the base model has been trained with the 1D Subspaces custom objective. The *HDFF-MLP* model trains an auxiliary MLP as a binary ID/OOD classifier using the generated HDFF image descriptor vectors $\mathbf{y}^{(i)}$ of perturbed (proxy-OOD) and normal (ID) images from the ID training set as input into the MLP alike NMD [10].

We re-implement the MSP, ML and Gram detectors for the WideResNet architecture, for all other methods, excluding NMD [10] and MOOD [30], we report the published results on the same architecture. At the time of writing, a publicly available implementation of NMD [10] is unavailable and as such we utilise their published results in the zero-shot OOD scenario on ResNet34 [16]. MOOD [30] is built upon a custom network architecture that enables early exits during inference, we report the published results on this custom architecture due to the unclear applicability of the method to other network architectures.

When applying HDFF and Gram [50] to the WideResNet model we attempt to faithfully recreate the hook locations of Gram from the original architecture. Specifically, features are recorded from the outputs of almost all of the following modules within the network: Conv2d, ReLU, BasicBlock, NetworkBlock and shortcut connections. In total, there are 76 features extracted per sample and hence the same number of semi-orthogonal projection matrices $\mathbf{P}$ are generated for

[1] Download links for OOD datasets can be found in the following repository: https://github.com/facebookresearch/odin

Statistics Stream - CIFAR10

OOD Dataset	HDFF (Ours)	HDFF-Ens (Ours)	Gram [50]	MSP [19]	ML [18]	Energy [31]
iSun	99.2	*99.3*	**99.9**	96.4	97.8	92.6
TINc	98.3	*98.4*	**99.4**	95.4	96.8	-
TINr	99.2	*99.4*	**99.8**	95.0	96.5	-
LSUNc	96.2	96.8	*98.1*	95.7	97.1	**98.4**
LSUNr	99.2	*99.4*	**99.9**	96.5	98.0	94.2
SVHN	*99.4*	**99.5**	*99.4*	96.0	97.2	91.0
MNIST	99.6	*99.7*	**99.97**	89.4	90.6	-
KMNIST	99.0	*99.1*	**99.98**	92.7	93.4	-
FMNIST	98.7	*99.1*	**99.8**	93.6	95.2	-
Textures	94.5	*94.8*	**98.2**	92.7	93.5	85.2
CIFAR100	75.4	75.8	79.4	**87.8**	*87.3*	-
Average	96.2	*96.5*	**97.6**	93.7	94.9	93.2

Table 1. OOD detection results for the against the methods contained belonging to the **statistics** stream. Comparison metric is AUROC, higher is better. **Best** results are shown in **blue and bold**, *second* best results are shown in *green and italics*. The ensemble in *HDFF-Ens* always consists of 5 models. HDFF and Gram are consistently the top two performers across the significant majority of the far-OOD detection settings.

Training Stream - CIFAR10

OOD Dataset	HDFF-MLP (Ours)	HDFF-1DS (Ours)	1DS [59]	NMD [10]	DDU [37]	MOOD [30]
iSun	**99.99**	*99.9*	-	*99.9*	-	93.0
TINc	**99.9**	*99.7*	98.1	99.2*	91.1*	-
TINr	**99.96**	*99.8*	98.5	-	91.1*	-
LSUNc	98.2	99.1	**99.4**	98.8	-	*99.2*
LSUNr	**99.99**	*99.9*	99.3	-	-	93.3
SVHN	84.8	*99.2*	-	**99.6**	97.9	96.5
MNIST	*99.4*	99.3	-	-	-	**99.8**
KMNIST	98.6	*99.3*	-	-	-	**99.9**
FMNIST	*99.6*	99.3	-	-	-	**99.9**
Textures	*97.4*	97.3	-	**98.9**	-	93.3
CIFAR100	69.9	*90.7*	-	90.1	**91.3**	-
Average	95.2	*98.5*	**98.8**	97.8	94.6	95.0

Table 2. OOD detection results for the against the methods contained belonging to the **training** stream. Comparison metric is AUROC, higher is better. **Best** results are shown in **blue and bold**, *second* best results are shown in *green and italics*. Published results that are unclear which variant of TIN they correspond to are identified with a *. The incorporation of HDFF into pre-existing pipelines leads to consistently improving or comparable results, demonstrating the robust nature of HDFF.

HDFF.

Unless otherwise stated, we operate in a hyperdimensional space of 10^4 dimensions, we ablate this hyperparameter in the Supplementary Material. Before projecting feature maps into the hyperdimensional space as per (5), we apply mean-centering by subtracting the layer-wise mean activations (obtained from the training set) from all $\mathbf{m}_l$. For pooling we apply max pooling over the spatial dimensions to reduce our feature maps $\mathbf{m}_l$ to a vector representation, we ablate the effect of this choice in the Supplementary Material.

5.2. Results and Discussion

Table 1 compares the results of our HDFF OOD detector on the AUROC metric to all of the methods in the **statistics** stream under both the near- and far-OOD settings. In the far-OOD setting HDFF and Gram are consistently the top performers with small performance differences of less than 1% AUROC between the two methods on most OOD dataset configurations. This finding indicates that the feature representations of the in-distribution data from both Gram and HDFF are powerful for the far-OOD detection task. On this note, we identify that the vector representation of HDFF is far more compact than the square matrix representations from Gram. The difference in representation complexity accounts for the performance differences, but also introduces large gaps in computational performance with gram requiring ≈4.5x longer per inference pass compared to HDFF as described in Section 5.3.

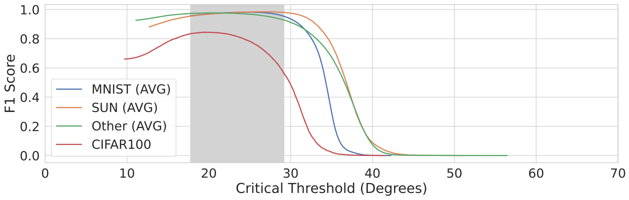

Figure 2. F1-score for binarisation at different critical values of angular distance to closest class descriptor. The model used is the 1D Subspaces trained WideResNet. The grey region corresponds to binarisations that would produce a result within 5% of the maximum F1-score achieved for all far-OOD datasets. To avoid clutter, far-OOD datasets have been grouped: i) MNIST (AVG) contains KMNIST, MNIST and FashionMNIST. ii) SUN (AVG) contains iSUN, LSUNr and LSUNc. iii) Other (AVG) contains all other far-OOD datasets. As expected, the near-OOD task (CIFAR100) leads to significantly lower thresholds compared to all far-OOD tasks.

We note that the MSP detector outperforms all other methods in the **statistics** stream in the near-OOD detection setting (CIFAR10 as in-distribution and CIFAR100 as OOD). Considering that HDFF is effectively detecting deviations in convolutional feature activations this would indicate that images with similar features are being grouped with in-distribution classes, this behaviour is discussed more in Section 5.6.

Table 2 compares the results of the methods in the **training** stream on the AUROC metric in the near- and far-OOD settings. The first finding from this table is the broadly powerful nature of the HDFF vector representation in combination with the MLP. Across the board, the majority of the top performing results are from the HDFF MLP detector demonstrating the power of the HDFF representation when combined with latest state-of-the-art detectors.

Secondly and more specifically, we note that when HDFF is applied to the 1D Subspaces trained model it improves upon the performance of the Spectral Discrepancy detector on 3 out of the 4 comparable benchmarks. We additionally note that the Spectral discrepancy detector requires 50 SWA-G [33] samples to achieve these performance levels whilst the HDFF detector requires only one inference pass, mandating a 5000% increase in computational time when using Spectral Discrepancy. These two findings combined reinforce the claim that HDFF is generally applicable to a wide range of models and training regimes.

5.3. Computational Efficiency

HDC techniques are commonly used for computation or learning in low-power situations [36, 20] and as such, we expect HDFF to introduce minimal computational overhead. For a full pass of the CIFAR10 test set, HDFF takes 7.0±0.9s compared to a standard inference pass at 6.0±0.7s, averaged over 5 independent runs. We note that this ≈17% increase is comparatively minor considering the large performance gains that HDFF boasts over the MSP and ML detectors. By comparison, the closest equivalent method to ours, Gram [50] takes 31.4±0.3s to complete a full inference pass over the CIFAR10 test set, resulting in an ≈4.5x longer inference time than HDFF. Additionally, we expect the computational efficiency of Gram to drop on larger networks due to the gram matrices scaling $O(n^2)$ in computation and memory requirements with the number of channels whereas HDFF scales linearly $O(n)$ with the number of channels.

Characteristic of belonging to the **statistics** stream, HDFF requires far less calibration or training than other methods belonging to the **training** stream. In particular, the closest comparative method, NMD, prescribes a training regime of 60-100 epochs for the MLP detector, whereas HDFF only requires a single epoch to collect the full in-distribution statistics, resulting in a minimum 60x decrease in computational time. Additionally, HDFF can be applied post-hoc to common networks, requiring no additional computation in the training or fine-tuning of the network. We further note that HDFF with only a single inference pass competes with or exceeds the Spectral Discrepancy detector [59] that requires 50 MC samples to be collected, necessitating a minimum inference increase of 5000%.

5.4. Critical Threshold Ablation

During deployment, it is often more useful if an OOD detector produces a binarisation of ID vs OOD rather than a raw OOD estimate; the grey regions in Figure 2 show a range of critical thresholds that will produce performance reasonably close to optimal in this setting. Specifically, the grey region in Figure 2 shows a region of confidence where any critical value would produce an F1-score within 5% of the maximum value achieved on each far-OOD dataset, an approximate standard of reasonable performance. The near-OOD detection task is plotted but does not contribute towards the definition of the grey region due to the severe differences between the near- and far-OOD tasks. As we can see, a large region of critical values around the range of 18-29 degrees will result in generally good performance for far-OOD detection.

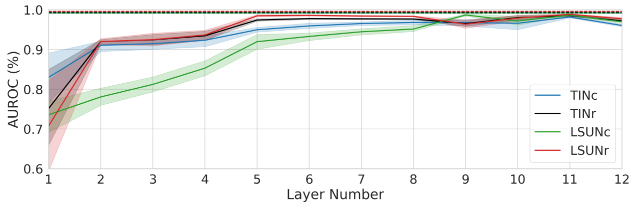

Figure 3. Comparison of effectiveness for OOD detection of each layer individually with the AUROC measure. CIFAR10 is the in-distribution dataset. Mean and 95% confidence interval over 5 randomly initialised models trained with the 1D Subspaces methodology. Dotted lines correspond to the results attained by the fusion of all 12 BasicBlocks. Performance of individual layers shows a trend toward later layers being more effective at detecting OOD data. The fusion of feature maps from across the network provides good general performance across all datasets and does not require calibration on OOD data to set.

5.5. Layer Sensitivivity To OOD Data

Congruent with other multi-layer OOD detectors [10, 50] we ablate the sensitivity of individual layers with respect to OOD samples from different target distributions. Figure 3 demonstrates the effectiveness of individual layers when they are used in isolation for OOD detection based on our HDFF OOD estimator. For the sake of readability, we only collect features from the outputs of the 12 BasicBlocks and only consider the TIN and LSUN datasets. In both ID settings earlier features in the network appear to be less reliable at detecting OOD samples; particularly the CIFAR10 setting has a clear upwards trend as the layer number increases. This observation lends weight towards suggestions in prior work that shallow layers in a DNN are unable to or at least are less effective at detecting OOD data [44].

The dotted lines, and related 95% confidence shaded area, for each dataset correspond to the fusion of information from all layers; we consider these lines the best that can be reasonably achieved within the bounds of the OOD task. In Figure 3 we observe that no single individual layer is able to detect the OOD data as well as the fusion of all feature maps. Critically, we note two points in favour of the fusion of features: (1) without OOD data the optimal layer combination for a given data set is unknown, and (2) the optimal layer combination is not consistent between OOD data configurations as shown by the drop in performance on LSUNc. To summarise, if there is no access to OOD data at training time to determine which layer(s) are the best, then the fusion of feature maps from across the network often provides the best performance or a close approximation. We provide extra ablations against other metrics and the CIFAR100 ID set in the Supplementary Material.

5.6. Distance of Features as Visual Similarity

As HDFF uses similarity preserving projections, the angular distance directly represents the differences between two

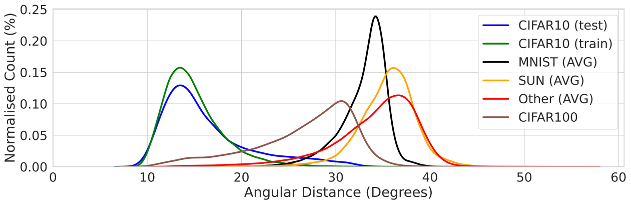

Figure 4. KDE estimate of separation between ID and OOD sets based on minimum angular distance to closest class representative in the CIFAR10 setting. To avoid clutter, far-OOD datasets have been grouped: i) MNIST (AVG) contains KMNIST, MNIST and FashionMNIST. ii) SUN (AVG) contains iSUN, LSUNr and LSUNc. iii) Other (AVG) contains all other far-OOD datasets. Overlap between the test and OOD distributions can be considered erroneous samples.

input sets of raw features. Since these features are extracted from a deep CNN, the angular distance between any two vectors is a proxy for their visual dissimilarity. This definition leads to intuitive understandings of how HDFF behaves and identifying failure cases; we discuss these points here.

Using our definition, since HDFF separates based on visual similarity, we can infer that failures in OOD detection are due either to ID samples being visually dissimilar to the training set or OOD samples are as similar, if not more so, to the training set than the ID samples. To aid in understanding this, Figure 4 visualises the differences in distributions of angular distance between the ID and OOD datasets in the CIFAR10 ID setting through a KDE estimate (for smoothing) over binned angular distances on the HDFF 1D Subspaces model. The area of overlap between the test set and any OOD set can be considered as erroneous detection. Inspecting Figure 4, we observe that, in the far-OOD settings, errors due to dissimilar ID samples are more likely to appear due to the distributional shift between the training and testing distributions, i.e. false positives. By contrast, in the near-OOD detection task we see that a significant number of errors are due to OOD samples appearing very similar to ID samples, i.e. false negatives.

As a more concrete example, Figure 5 demonstrates HDFF separating input samples based on the angle to the nearest class descriptor vector, in this case, the CIFAR10 truck class. Consistent with our previous assertions, we observe in Figure 5 that samples that are $< 15^\circ$ from the class descriptor vector appear visually very similar, with no far-OOD samples occupying this range. In the range of $15-30^\circ$ we observe that samples from all datasets still have vehicle-like appearance, but whether or not these accurately represent a truck is debatable; this region is still predominantly populated by ID and near-OOD samples. Once outside the 30° ring, we see that the significant majority of samples do not appear vehicle-like with the very few ID samples in this region having the truck visually obscured; this region is dominated by far-OOD samples.

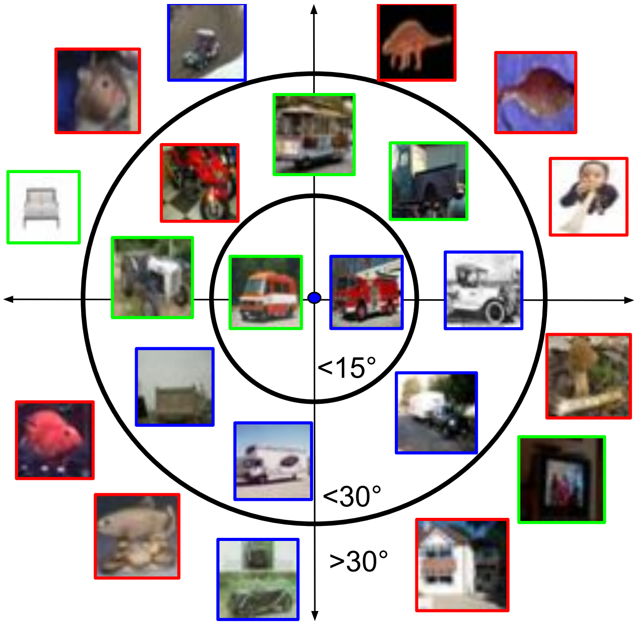

Figure 5. Sample images from in-distribution (CIFAR10, blue), near-OOD (CIFAR100, green) and far-OOD (TINc, red) datasets with their approximate distance to the nearest class descriptor vector, corresponding to the CIFAR10 ID truck class. The underlying model is the 1D Subspaces trained WideResNet. Distances to the class bundle (centre blue dot) can be approximately inferred from which circle the sample is encapsulated by. This figure reinforces the hypothesis that HDFF separates based on visual similarity to the in-distribution class as truck-like objects appear within the inner circle, vehicle-like objects in the middle region and heavily dissimilar object falling outside both of those regions.

6. Conclusion

This paper introduced powerful ideas from hyperdimensional computing into the important task of OOD detection. We investigate the sensitivity of individual layers to OOD data and find that the fusion of feature maps provides the best general performance with no requirements for OOD data to fine-tune on. We perform competitively with state-of-the-art OOD detection methods with the added benefit of significantly reducing the computational costs associated with the current state-of-the-art. We show the interpretation of cosine distance as a proxy for visual similarity allows for additional failure diagnosis capabilities over competing methods. In this paper, we utilised the simple but powerful element-wise addition for bundling, however, this is one of potentially many applications of HDC to DNNs, opening new future research directions.

Acknowledgements: The authors acknowledge continued support from the Queensland University of Technology (QUT) through the Centre for Robotics. TF was partially supported by funding from ARC Laureate Fellowship FL210100156 and Intel's Neuromorphic Computing Lab. We also thank Peer Neubert from Chemnitz University of Technology for greatly insightful discussions on the topic of Hyperdimensional Computing.

References

[1] Petra Bevandić, Ivan Krešo, Marin Oršić, and Siniša Šegvić. Simultaneous semantic segmentation and outlier detection in presence of domain shift. In *Pattern Recognition*, pages 33–47, 2019.

[2] Hermann Blum, Paul-Edouard Sarlin, Juan Nieto, Roland Siegwart, and Cesar Cadena. The fishyscapes benchmark: Measuring blind spots in semantic segmentation. *International Journal of Computer Vision*, 129(11):3119–3135, 2021.

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

[4] Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature. *CoRR*, abs/1812.01718, 2018.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[6] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[7] Terrance DeVries and Graham W Taylor. Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint arXiv:1802.04865*, 2018.

[8] Akshay Raj Dhamija, Manuel Günther, and Terrance Boult. Reducing network agnostophobia. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[9] Giancarlo Di Biase, Hermann Blum, Roland Siegwart, and Cesar Cadena. Pixel-wise anomaly detection in complex driving scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16918–16927, 2021.

[10] Xin Dong, Junfeng Guo, Ang Li, Wei-Te Ting, Cong Liu, and H.T. Kung. Neural mean discrepancy for efficient out-of-distribution detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19217–19227, 2022.

[11] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, 2016.

[12] Stephen I. Gallant and T. Wendy Okaywe. Representing Objects, Relations, and Sequences. *Neural Computation*, 25(8):2038–2078, 2013.

[13] R. Gayler. Multiplicative binding, representation operators & analogy. In *Advances in analogy research: Integr. of theory and data from the cogn., comp., and neural sciences*, 1998.

[14] Lulu Ge and Keshab K. Parhi. Classification using hyperdimensional computing: A review. *IEEE Circuits and Systems Magazine*, 20(2):30–47, 2020.

[15] Jan Gosmann and Chris Eliasmith. Vector-derived transformation binding: An improved binding operation for deep symbol-like processing in neural networks. *Neural Computation*, 31(5):849–869, 2019.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[17] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 41–50, 2019.

[18] Dan Hendrycks, Steven Basart, Mantas Mazeika, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. Scaling out-of-distribution detection for real-world settings. *arXiv preprint arXiv:1911.11132*, 2019.

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

[20] Alejandro Hernández-Cano, Cheng Zhuo, Xunzhao Yin, and Mohsen Imani. Real-time and robust hyperdimensional classification. In *Great Lakes Symposium on VLSI*, page 397–402, 2021.

[21] Yen-Chang Hsu, Yilin Shen, Hongxia Jin, and Zsolt Kira. Generalized ODIN: Detecting out-of-distribution image without learning from out-of-distribution data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10951–10960, 2020.

[22] Haiwen Huang, Zhihan Li, Lulu Wang, Sishuo Chen, Bin Dong, and Xinyu Zhou. Feature space singularity for out-of-distribution detection. In *Workshop on Artificial Intelligence Safety*, 2021.

[23] Pentti Kanerva. Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors. *Cognitive Computation*, 1(2):139–159, 2009.

[24] Brent Komer, Terrence Stewart, Aaron Voelker, and Chris Eliasmith. A neural representation of continuous space using fractional binding. In *CogSci*, pages 2038–2043, 2019.

[25] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[26] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, 2017.

[27] Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. In *Proceedings of the International Conference on Learning Representations*, 2018.

[28] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, 2018.

[29] Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *Proceedings of the International Conference on Learning Representations*, 2018.

[30] Ziqian Lin, Sreya Dutta Roy, and Yixuan Li. Mood: Multi-level out-of-distribution detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15313–15323, 2021.

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

[32] Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. In *Advances in Neural Information Processing Systems*, 2019.

[33] Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. In *Advances in Neural Information Processing Systems*, 2019.

[34] Dimity Miller, Niko Sunderhauf, Michael Milford, and Feras Dayoub. Class anchor clustering: A loss for distance-based open set recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3570–3578, 2021.

[35] Dimity Miller, Niko Sünderhauf, Michael Milford, and Feras Dayoub. Uncertainty for identifying open-set errors in visual object detection. *IEEE Robotics and Automation Letters*, 7(1):215–222, 2021.

[36] Justin Morris, Yilun Hao, Roshan Fernando, Mohsen Imani, Baris Aksanli, and Tajana Rosing. Locality-based encoder and model quantization for efficient hyper-dimensional computing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2021. to appear.

[37] Jishnu Mukhoti, Andreas Kirsch, Joost van Amersfoort, Philip H. S. Torr, and Yarin Gal. Deep deterministic uncertainty: A simple baseline. *arXiv preprint arXiv:2102.11582*, 2022.

[38] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Ng. Andrew. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[39] Peer Neubert, Stefan Schubert, and Peter Protzel. An introduction to hyperdimensional computing for robotics. *Künstliche Intelligenz*, 33(4):319–330, 2019.

[40] Peer Neubert, Stefan Schubert, Kenny Schlegel, and Peter Protzel. Vector semantic representations as descriptors for visual place recognition. In *Robotics: Science and Systems*, 2021.

[41] Philipp Oberdiek, Matthias Rottmann, and Gernot A. Fink. Detection and retrieval of out-of-distribution objects in semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 1331–1340, 2020.

[42] Nicolas Papernot and Patrick D. McDaniel. Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning. *arXiv preprint arXiv:1803.04765*, 2018.

[43] Tony Plate. A common framework for distributed representation schemes for compositional structure. In *Connectionist Systems for Knowledge Representation and Deduction*, pages 15–34, 1997.

[44] Janis Postels, Hermann Blum, Cesar Cadena, Roland Siegwart, Luc Van Gool, and Federico Tombari. Quantifying aleatoric and epistemic uncertainty using density estimation in latent space. *arXiv preprint arXiv:2012.03082*, 2020.

[45] Abbas Rahimi, Pentti Kanerva, and Jan M. Rabaey. A robust and energy-efficient classifier using brain-inspired hyperdimensional computing. In *International Symposium on Low Power Electronics and Design*, page 64–69, 2016.

[46] Quazi Marufur Rahman, Niko Sünderhauf, and Feras Dayoub. Online monitoring of object detection performance post-deployment. *arXiv preprint arXiv:2011.07750*, 2020.

[47] Quazi Marufur Rahman, Niko Sünderhauf, and Feras Dayoub. Did you miss the sign? A false negative alarm system for traffic sign detectors. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3748–3753, 2019.

[48] 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. *arXiv preprint arXiv:2106.09022*, 2021.

[49] Amir Rosenfeld, Richard S. Zemel, and John K. Tsotsos. The elephant in the room. *CoRR*, abs/1808.03305, 2018.

[50] Chandramouli Shama Sastry and Sageev Oore. Detecting out-of-distribution examples with Gram matrices. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 8491–8501. PMLR, 2020.

[51] Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *Proceedings of the International Conference on Learning Representations*, 2014.

[52] Kenny Schlegel, Peer Neubert, and Peter Protzel. A comparison of vector symbolic architectures. *Artificial Intelligence Review*, pages 1–33, 2021.

[53] Paul Smolensky. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, 46(1):159–216, 1990.

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

[55] Niko Sünderhauf, Oliver Brock, Walter Scheirer, Raia Hadsell, Dieter Fox, Jürgen Leitner, Ben Upcroft, Pieter Abbeel, Wolfram Burgard, Michael Milford, et al. The limits and potentials of deep learning for robotics. *The International Journal of Robotics Research*, 37(4-5):405–420, 2018.

[56] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[57] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, 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.

[58] Qing Yu and Kiyoharu Aizawa. Unsupervised out-of-distribution detection by maximum classifier discrepancy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9518–9526, 2019.

[59] Alireza Zaeemzadeh, Niccolo Bisagno, Zeno Sambugaro, Nicola Conci, Nazanin Rahnavard, and Mubarak Shah. Out-

of-distribution detection using union of 1-dimensional subspaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9452–9461, 2021.

[60] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference*, pages 87.1–87.12, 2016.

[61] Arthur Zimek, Erich Schubert, and Hans-Peter Kriegel. A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining*, 5(5):363–387, 2012.