Class Anchor Clustering: A Loss for Distance-based Open Set Recognition

Dimity Miller, Niko Sünderhauf, Michael Milford, Feras Dayoub
Australian Centre for Robotic Vision, Queensland University of Technology
Brisbane, Australia
{d24.miller, niko.suenderhauf, michael.milford, feras.dayoub}@qut.edu.au

Abstract

In open set recognition, deep neural networks encounter object classes that were unknown during training. Existing open set classifiers distinguish between known and unknown classes by measuring distance in a network's logit space, assuming that known classes cluster closer to the training data than unknown classes. However, this approach is applied post-hoc to networks trained with cross-entropy loss, which does not guarantee this clustering behaviour. To overcome this limitation, we introduce the Class Anchor Clustering (CAC) loss. CAC is a distance-based loss that explicitly trains known classes to form tight clusters around anchored class-dependent cluster centres in the logit space. We show that training with CAC achieves state-of-the-art open set performance for distance-based open set classifiers on the standard benchmark datasets, with a 2.4% performance increase in AUROC on the challenging TinyImageNet, without sacrificing classification accuracy. We also show that our anchored class centres achieve higher open set performance than learnt class centres, particularly on object-based datasets and large numbers of training classes.

1. Introduction

Many practical applications require the deployment of trained visual perception models under *open set* conditions, such as autonomous systems, driverless cars, and robotics. In open set conditions, a model encounters object classes that were not present during training (referred to as 'unknown' classes) [17]. Deep convolutional neural networks (CNNs) degrade in performance in open set conditions, as they can confidently misclassify unknown classes as known training classes [4, 8, 14]. This behaviour raises serious concerns about the safety of using CNNs in open set environments [1] – particularly on autonomous systems where perception failures may have severe consequences [4, 21].

Open set recognition was introduced to extend object recognition to an open set environment [17]. During testing, an open set classifier must classify known object classes and reject unknown object classes [17]. In this paper, we propose a new distance-based loss that achieves state-of-the-art performance for distance-based open set classification.

Many open set classifiers model the position of known training data in the final layer, or logit space, of a CNN [2, 25, 26]. Such approaches assume that known classes cluster tightly in the logit space, and all unknown classes will maintain a distance from these clusters. Figure 1a shows this ideal performance. Current methods apply this concept to networks trained with cross-entropy loss [2, 25, 26]. However, cross-entropy loss does not encourage (nor guarantee) the clustering behaviour these methods seek to exploit. We exhibit this in Figure 1b, where we train a CNN with cross-entropy loss to classify trains, buses, and bicycles (CIFAR100 classes). The resulting logit space of this CNN appears crowded with inflated class clusters, and it is challenging to distinguish the unknown classes (bear and possum) from these clusters.

In this work, we introduce the Class Anchor Clustering (CAC) loss to address this limitation in prior approaches. CAC is a distance-based loss that explicitly encourages the known training data to form tight clusters around anchored, class-specific centre points in the logit space. CAC is compatible with existing classification networks, with only slight modifications to the network architecture. Compared to the cross-entropy trained CNN, a logit space trained with CAC exhibits tight, separate class clusters and an improved distinction of these clusters from unknown classes (see Figure 1c).

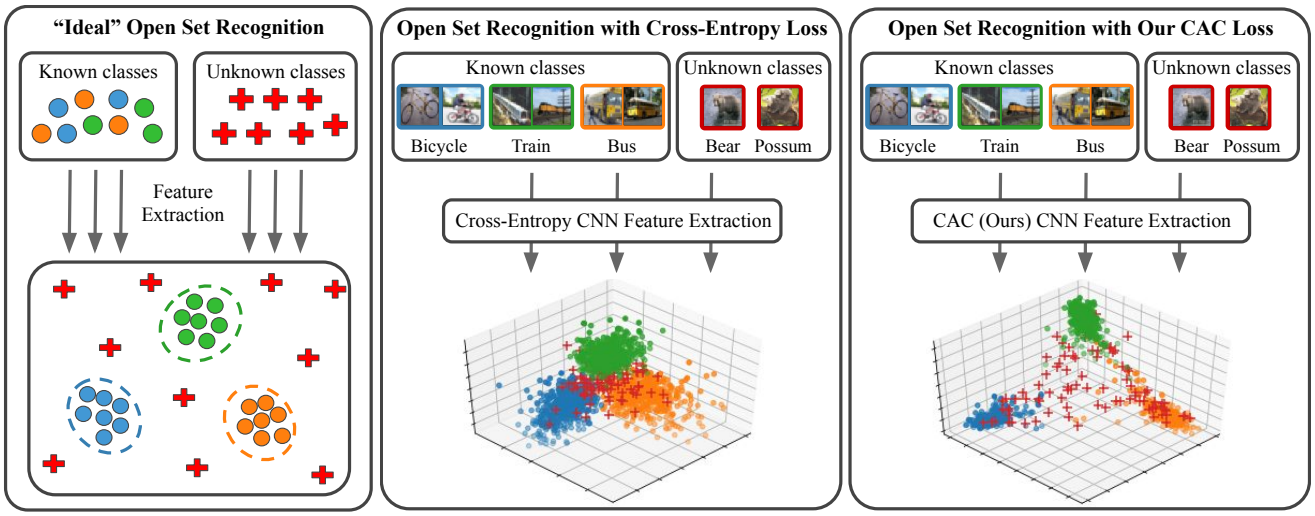
Our paper makes the following contributions:

1. We propose a new loss term for open set recognition that encourages known class training data to cluster tightly around class-specific centre points in logit space.

2. We show that training with this novel *Class Anchor Clustering* (CAC) loss achieves new state-of-the-art open set performance for distance-based open set classifiers, without sacrificing classification accuracy.

3. We introduce the concept of *anchored* class centres as an effective and scalable strategy for distance-based training. In contrast to class centres that are learnt during training, anchored centres are beneficial for datasets with high intra-class variation and large numbers of classes, such as TinyImageNet.

The authors acknowledge continued support from the Queensland University of Technology (QUT) through the Centre for Robotics. This research was conducted by the Australian Research Council Centre of Excellence for Robotic Vision (project number CE140100016).



(a) 'Ideal' open set recognition (b) Open set reocngition with cross-entropy loss (c) Open set recognition with our CAC loss

Figure 1: Left: An 'ideal' open set classifier will tightly cluster known classes in the feature space, and unknown classes will fall far away. Middle: A CNN trained on real image data from CIFAR100 with the standard cross-entropy loss shows a final logit space with inflated known class clusters which cannot be easily distinguished from the unknown classes. Right: A CNN trained with our proposed CAC loss (on the same CIFAR100 data) shows a final logit space with tight, separated known class clusters and improved distinction of unknown classes.

2. Related Work

Open set recognition Open set recognition is multiclass classification with the additional requirement of rejecting inputs from unknown classes [17]. This is formalised as the task of minimising open space risk, the portion of classification space labelled as 'known' that is far from the known training data, while maintaining generalisation and classification accuracy on the known classes [17]. Related areas, such as out-of-distribution and novelty detection, exist as relaxed forms of open set classification where known and unknown classes are from different distributions [8] or multiclass classification is not required [5]. For this work, we focus specifically on open set recognition.

OpenMax was one of the first CNN-based open set classifiers, using the network's final layer's logits, or logit space, as the classification space with open space risk [2]. OpenMax models each known class as a single cluster, and using a Weibull distribution, re-calibrates an input's softmax scores based on the distance to each cluster centre. OpenMax was the first 'distance-based' approach, using separation from the training data to minimise the open space risk of a CNN.

Several following works employed real or generated 'known unknown' data to augment the training dataset, either using the data to improve the feature representation for distance-based measures [7,26] or to bound the known classification space with an 'other' class [12, 18].

Other recent open set classifiers use a combined classifier and autoencoder network architecture [15, 25]. In [25], OpenMax is applied to the joint classifier logit space and auto-encoder latent space, with the additional reconstruction-learnt features improving the overall feature representation. In contrast, [15] uses the reconstruction error from a class-conditioned autoencoder-classifier to distinguish between known and unknown inputs. While [15] currently has state-of-the-art performance for open set image classification, it does not explicitly minimise open space risk. Others [6, 19] observed that reconstruction error alone is not suitable as a measure of class novelty, as inputs from unknown classes have been reconstructed with low error. The inclusion of a distance-based approach with reconstruction error has been shown to improve this performance [6].

In contrast to existing distance-based open set classifiers [2, 25, 26], which *assume* known classes will tightly cluster but train with cross-entropy loss, our work is the first to train with a distance loss when using distance for testing.

Distance Losses for Deep Neural Networks The field of metric learning uses distance loss functions to learn meaningful feature embeddings. Triplet loss is a popular distance loss that encourages inputs to minimise distance to a 'positive example and maximise distance to a single 'negative example. Tuplet loss was introduced as an extension of triplet loss that maximises an inputs distance to *multiple* 'negative

examples [20]. We adopt a modified version of Tuplet loss as one of two terms in our new CAC loss. We will show that Tuplet loss alone is not sufficient and that *both* terms of CAC are necessary for best performance.

Center Loss [23] was proposed to improve discriminative learning for facial recognition by encouraging clustering in a feature space. It is used in conjunction with cross-entropy loss and encourages an input to minimise distance to its ground truth class centre. The class centres are learnt simultaneously with the feature embedding during training. In contrast, we propose to use *anchored*, i.e. fixed, class centres. This makes training more stable and, as we will show, more scalable to larger and more complex datasets.

Recently, [11] demonstrated the utility of metric learning for open set classification, however only for fine-grained image classification. Such metric learning approaches compute distances between individual instances of the training data, and the sampling technique used to achieve this can have a significant effect on the convergence speed and stability of the training minimum [24]. As discussed in [16], this sampling typically makes metric learning computationally intractable on larger datasets, such as CIFAR10, CIFAR100, or ImageNet. Although recent work [16] adapted metric learning approaches for large-scale datasets, this technique degrades the classification accuracy of a standard cross-entropy network, making it unsuitable for open set classification on large-scale datasets.

3. Class Anchor Clustering (CAC) for Open Set Classification

We introduce the two core ideas of our paper that enable distance-based training for large-scale image open set classification: (1) the Class Anchor Clustering (CAC) loss that encourages training data to form tight, class-specific clusters. Tight clusters make it easier to distinguish between known and unknown class inputs during deployment. (2) the concept of using *anchored* class centres in the logit space to fix the cluster centre positions during training. In contrast to learnt class centres, anchored class centres stabilise the training and scale well to object-based datasets and to training with more known classes.

General Architecture CAC is compatible with existing classification networks, and requires only slight modifications to architecture and training procedure. Our proposed CAC-trained open set classifier has three main components:

1. A base network, f, that projects an input image $\mathbf{x}$ to a vector of class logits $\mathbf{z} = f(\mathbf{x})$. This network can be any existing classifier with an N-dimensional logit space, where N is the number of known classes.

2. A non-trainable parameter, $\mathbf{C}$, representing a set of class centre points $(\mathbf{c}_1, \ldots, \mathbf{c}_N)$, one for each of the N known classes.

3. A new layer, $e(\mathbf{z}, \mathbf{C})$, that calculates $\mathbf{d}$, a vector of Euclidean distances between a logit vector $\mathbf{z}$ and the set of class centres $\mathbf{C}$.

In summary, the output of our distance-based classifier is

$$\mathbf{d} = e(\mathbf{z}, \mathbf{C}) = (\|\mathbf{z} - \mathbf{c}_1\|_2, \ldots, \|\mathbf{z} - \mathbf{c}_N\|_2)^\mathsf{T} \quad (1)$$

where $\|\cdot\|_2$ denotes the Euclidean norm.

3.1. Training with a Distance-based Loss Function

During training, we wish to learn a logit space embedding $f(\mathbf{x})$ where known inputs form tight, class-specific clusters. We introduce our CAC loss and the concept of anchored class centres to encourage this clustering behaviour during training. During testing, the clustering enables us to use a distance-to-class-centre metric to a) reject unknown class inputs, and b) classify known class inputs.

3.1.1 Class Anchor Clustering Loss

We require a distance-based loss that a) encourages training inputs to minimise the distance to their ground-truth class centre, while b) maximising the distance to all other class centres to encourage discriminative learning.

To do this, we use a modified Tuplet loss term $\mathcal{L}_T$ [20] that forces an input $\mathbf{x}$ to maximise the difference in distance to the correct class centre $\mathbf{c}_y$ and all other class centres. Remembering that $\mathbf{d} = (d_1, \ldots, d_N)^\mathsf{T}$ is defined as in (1), we define this loss component as

$$\mathcal{L}_T(\mathbf{x}, y) = \log\left(1 + \sum_{j \neq y}^{N} e^{d_y - d_j}\right). \quad (2)$$

$\mathcal{L}_T$ differs from Tuplet loss [20] because it is based on class centres $\mathbf{C}$ rather than sampled class instances. Our modified Tuplet loss term is equivalent to cross-entropy loss applied to the distance vector $\mathbf{d}$, but used with a $\mathrm{softmin}$ function rather than $\mathrm{softmax}$ (see supplementary material for proof). The $\mathrm{softmin}$ function is the opposite of $\mathrm{softmax}$: it assigns a large value (≈ 1) to the *smallest* value of the input vector and is defined as:

$$\mathrm{softmin}(\mathbf{d})_i = \frac{e^{-\mathbf{d}_i}}{\sum_{k=1}^{N} e^{-\mathbf{d}_k}}. \quad (3)$$

While effective for discriminative learning, $\mathcal{L}_T$ (and cross-entropy loss) aim to maximise the *margin* between correct and incorrect inputs. To ensure an input is explicitly forced to lower its *absolute* distance to the correct class centre, we also penalise the Euclidean distance between the

training logit and the ground truth class centre. We refer to this as the Anchor loss term:

$$\mathcal{L}_A(\mathbf{x}, y) = d_y = \|f(\mathbf{x}) - \mathbf{c}_y\|_2. \quad (4)$$

We combine the Anchor and Tuplet loss terms to form our final distance-based loss, which we refer to as the Class Anchor Clustering (CAC) loss:

$$\mathcal{L}_{\text{CAC}}(\mathbf{x}, y) = \mathcal{L}_T(\mathbf{x}, y) + \lambda \mathcal{L}_A(\mathbf{x}, y). \quad (5)$$

A hyperparameter of our method is λ, which balances these two individual loss terms (explored in section 5.2.3). By combining the Anchor and Tuplet loss terms, our loss encourages training inputs to minimise the distance to their ground-truth anchored class centre, while maximising the distance to other anchored class centres.

3.1.2 Anchored Class Centres

We introduce anchored class centre points as a method of *anchoring*, i.e. fixing, cluster centres for each class in the logit space during training. By anchoring our class centres during training, we eliminate the need to learn another parameter (as done in previous approaches to distance losses, e.g. [23]). For each known class i, our network has an anchored class centre $\mathbf{c}_i$ in the logit space. Given an N-dimensional logit space for N known classes, we place the anchored centre for each known class at a point on its class coordinate axis. This anchored centre point is therefore equivalent to a scaled standard basis vector $\mathbf{e}_i$, or scaled one-hot vector, for each class. The magnitude of the anchored centre, α, is a hyperparameter of our method (explored in section 5.2.3). We summarise this below:

$$\mathbf{C} = (\mathbf{c}_1, \ldots, \mathbf{c}_N) = (\alpha \cdot \mathbf{e}_1, \ldots, \alpha \cdot \mathbf{e}_N) \quad (6)$$

$$\mathbf{e}_1 = (1, 0, \ldots, 0)^\mathsf{T}, \quad \mathbf{e}_N = (0, \ldots, 0, 1)^\mathsf{T}. \quad (7)$$

After completing training, the anchored class centre positions $\mathbf{C}$ are adjusted to the mean position of the correctly classified training data. This allows us to model the class cluster centres for complex datasets more accurately, where visual and semantic similarities between classes can cause slight divergence from the original anchored class centre positions.

3.2. Using Distance-based Measures during Testing

During testing, the trained network has to reject unknown class inputs and correctly classify known class inputs. Our CAC loss trains known inputs to have two distance-based properties: (1) a high $\mathrm{softmin}$ score for the correct known class anchor (as per the modified Tuplet loss term $\mathcal{L}_T$) and (2) a low absolute distance to the correct known class anchor (as per the Anchor loss term $\mathcal{L}_A$). Based on this, we calculate rejection scores $\boldsymbol{\gamma} = (\gamma_1, ..., \gamma_N)^\mathsf{T}$ that express the classifier's *dis*belief that the input $\mathbf{x}$ belongs to each of the N known classes. We calculate the rejection scores $\boldsymbol{\gamma}$ as the element-wise product ($\circ$) of the distance vector $\mathbf{d}$ and its inverted $\mathrm{softmin}$:

$$\boldsymbol{\gamma} = \mathbf{d} \circ (1 - \mathrm{softmin}(\mathbf{d})) \quad (8)$$

By weighting the absolute distance with the inverted $\mathrm{softmin}$ score, inputs must have both a low absolute distance and high $\mathrm{softmin}$ score to be assigned a low rejection score for a known class. If all values in $\boldsymbol{\gamma}$ are above a threshold θ, the input does not belong to any known class and is rejected as unknown. Otherwise, the class label corresponding to the smallest value in $\boldsymbol{\gamma}$ is assigned:

$$\text{decision} = \begin{cases} \text{rejected as unknown} & \text{if } \min(\boldsymbol{\gamma}) > \theta \\ \text{class } i = \operatorname{argmin} \boldsymbol{\gamma} & \text{if } \min(\boldsymbol{\gamma}) \leq \theta \end{cases} \quad (9)$$

Using this distance-based decision procedure minimises open space risk [17]: the further away an input $\mathbf{x}$ projects from the class-specific centres, the more likely it is to be rejected as unknown.

4. Experimental Setup

We follow the exact evaluation protocol defined in [12]. In this protocol, six standard classification datasets are adapted for open set recognition by randomly splitting the dataset classes into 'known or 'unknown classes. The classifier trains with only the 'known classes. Depending on the proportion of known and unknown classes, [17] defined the *openness* **O** of the classification task as

$$\mathbf{O} = 1 - \sqrt{\frac{2 \cdot N_{\text{train}}}{N_{\text{test}} + N_{\text{target}}}} \quad (10)$$

where N_{train} is the number of classes during training, N_{target} is the number of classes requiring classification during testing and N_{test} is the total number of classes during testing (known and unknown). In general, a higher openness indicates a more difficult open set problem setup, but other factors such as the visual similarity between known and unknown classes influence the difficulty as well. For each dataset, we evaluate performance over 5 trials with random known and unknown class splits.

4.1. Datasets

The details of each dataset in its open set configuration are summarised below.

MNIST [10]: grayscale 32×32 images of handwritten digits, 6 known and 4 unknown classes, **O** = 13.39%.

SVHN [13]: RGB 32×32 images of street view house digits, 6 known and 4 unknown classes, **O** = 13.39%.

CIFAR10 [9]: RGB 32×32 images of animals and objects, 6 known and 4 unknown classes, **O** = 13.39%.
CIFAR+10/+50: considers the 4 non-animal classes of CIFAR10 as known, and 10 or 50 randomly sampled animal classes from CIFAR100 [9] as unknown (**O** = 33.33% and 62.86%).
TinyImageNet [22]: RGB 64×64 images of animals and objects, 20 known and 180 unknown classes, **O** = 57.35%. TinyImageNet images can contain significant background information unrelated to the object class, a number of classes are very visually and semantically related (e.g. different breeds of dogs), and there is high visual variation within individual classes. Examples are provided in the supplementary material.

TinyImageNet is of particular importance and interest for our evaluation as it represents the most challenging dataset in this benchmark in several ways. It contains a limited number of only 500 training images per class, a comparatively large image size of 64×64, a high openness score of 57.35%, and the inclusion of visually and semantically very similar classes. Compared to the other benchmark datasets, this makes the evaluation performance on TinyImageNet the most relevant and indicative for performance in practical applications. As we will show, our approach achieves a new state-of-the-art result on TinyImageNet.

4.2. Metrics

As established by [12], we use two different metrics to assess the performance of an open set classifier.
Area Under the ROC Curve (AUROC) is a calibration-free measure of the open set performance of a classifier. The Receiver Operating Characteristic (ROC) curve represents the trade-off between true positive rate (unknown inputs correctly rejected as 'unknown') and the false positive rate (known inputs incorrectly rejected as 'unknown') when applying varying thresholds to a given score. We modify the threshold θ that is compared to our network's rejection scores γ as discussed in (9).
Classification Accuracy measures the classifier's accuracy when applied to only the known classes in the dataset, equivalent to closed set classification. An open set classifier should maintain the classification accuracy of a standard closed set classifier.

4.3. State-of-the-art Methods for Comparison

We compare to six existing state-of-the-art open set classifiers. The core details of each method and the metric used for open set identification are listed below:
SoftMax [8]: A standard classifier using the maximum class softmax score for open set identification.
Open Set Recognition with Counterfactual Images (OSRCI) [12]: A cross-entropy network is trained with generated 'unknown' samples and an 'unknown' class. The difference between the 'unknown' class and maximum known class softmax score is used for open set identification.
Class Conditioned Auto-Encoder (C2AE) [15]: An autoencoder is added to a cross-entropy trained classifier and trained in a class-conditioned approach to reconstruct images. The smallest reconstruction error for any class conditioning is used for open set identification.
OpenMax [2]: A cross-entropy trained classifier using the maximum distance-calibrated softmax score for open set identification.
Generative OpenMax (G-OpenMax) [26]: A generative network produces 'unknown' samples to augment the training dataset of a cross-entropy trained classifier. The maximum distance-calibrated softmax score is used for open set identification.
Classification-Reconstruction learning for Open Set Recognition (CROSR) [25]: An autoencoder and classifier are jointly trained. OpenMax is applied to the logits and autoencoder latent space and the maximum distance-calibrated softmax score is used for open set identification.

While the last three methods [2, 25, 26] use distance during *testing* to distinguish between known and unknowns, ours is the first proposed open set classifier that also *trains* with a distance based loss function.

4.4. Implementation Details

We use the same network architecture as specified by the evaluation protocol in [12]. We use a Stochastic Gradient Descent (SGD) optimizer with a learning rate of 0.01 and train until convergence. We then complete another training cycle with a lower learning rate of 0.001 and train again until convergence. Specific details about the training procedure and network architecture are in the supplementary material. For all datasets, we use an Anchor loss weight λ of 0.1 and a logit anchor magnitude α of 10.

5. Results and Discussion

Our evaluation revealed four main results that we discuss in the following: (1) CAC outperforms the existing distance-based open set classifiers [2, 25, 26] on the 5 most complex of the 6 tested datasets, without sacrificing classification accuracy (Section 5.1). (2) Compared to other distance losses, CAC achieves better open set performance (Section 5.2.1). (3) Training with *anchored* class centres achieves better open set performance on nearly all tested datasets. Compared to learnt centres, anchored centres also maintain open set performance better with increasing numbers of known classes (Section 5.2.2). (4) Training with CAC is insensitive to the choice of its two hyperparameters over a wide range of values (Section 5.2.3).

Method	Use Distance in Training	Use Distance in Testing	MNIST	SVHN	CIFAR10	CIFAR+10/+50	TinyImageNet
Softmax [8]	✗	✗	0.978	0.886	0.677	0.816/0.805	0.577
OSRCI [12]	✗	✗	0.988	0.910	0.699	0.838/0.827	0.586
C2AE [15]	✗	✗	*0.989*	*0.922*	**0.895**	**0.955/0.937**	*0.748*
OpenMax [2]	✗	✓	0.981	0.894	0.695	0.817/0.796	0.576
G-OpenMax [26]	✗	✓	0.984	0.896	0.675	0.827/0.819	0.580
CROSR [25]	✗	✓	**0.991**	0.899	-	-	0.589
CAC (Ours)	✓	✓	0.987	**0.942**	*0.803*	*0.863/0.872*	**0.772**

Table 1: Open set AUROC for state-of-the-art methods and our proposed approach. Best and second best performance are bolded and italicised respectively.

5.1. Comparison with State-of-the-Art Open Set Classifiers

The open set performance of our proposed approach is compared to the state-of-the-art methods in Table 1.

Comparison with other distance-based approaches: Compared to other state-of-the-art methods that use distance during testing [2, 25, 26], we achieve the best open set performance on TinyImageNet, SVHN, CIFAR10, CIFAR+10, and CIFAR+50. Our performance increase is most substantial on TinyImageNet and CIFAR10, where there is an increase of 18.3% and 12.8%.

Our proposed approach is the first method that trains with a distance-based loss when using distance during testing. To analyse the impact of distance-based training, we examine the distributions of known class and unknown class distances to a class centre. In Figure 2, we compare these distributions for a network trained with cross-entropy loss (as used by [2, 25, 26]) and a network trained with our proposed CAC loss. The CAC-trained network has a known distribution that clusters more tightly to the class centres (behaviour it was trained for), and as a result, there is a lower overlap with the unknown distribution. By reducing the overlap with the unknown distribution, the open set classifier can more accurately identify unknown inputs with distance, thus improving open set performance.

In Table 2, we quantitatively show that training with our distance-based loss decreases the overlap between the known and unknown class distance distributions in comparison to cross-entropy loss. The table shows the Bhattacharyya coefficient [3], an established measure of the overlap between two distributions. For each of the datasets, our CAC loss results in a lower Bhattacharyya coefficient, on average by 14.3%. These results indicate that training with distance improves distance-based distinction of known and unknown classes during testing.

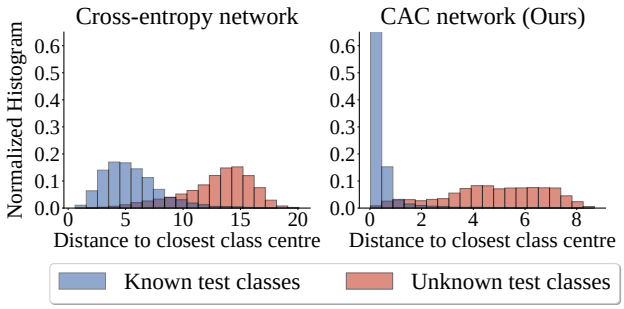


Figure 2: Distances to the closest anchor for known (blue) and unknown (red) test data. Training with CAC (right) leads to tighter clusters, compared to cross entropy (left), enabling better open set performance. Plots show the distribution for MNIST, see Table 2 for other datasets.

Dataset	Cross-Entropy	CAC (Ours)
MNIST	0.414	**0.324**
SVHN	0.700	**0.573**
CIFAR10	0.946	**0.868**
CIFAR+10	0.899	**0.766**
CIFAR+50	0.889	**0.751**
TinyImageNet	0.984	**0.913**

Table 2: The Bhattacharyya coefficient between the distributions of known class and unknown class distance to the closest class centre. A lower coefficient represents less overlap between the distributions and enables better distance-based open set recognition.

Comparison to non-distance-based approaches: Compared to non-distance-based open set classifiers, we achieve the best performance on TinyImageNet and SVHN, and come second to the class-conditioned auto-encoder (C2AE) approach [15] on CIFAR10 and CIFAR+10/+50.

While CAC achieves state-of-the-art performance on TinyImageNet with 20 known classes, it performs less well on CIFAR10 variations which have lower numbers of classes. As a distance-based method, CAC relies on a high-

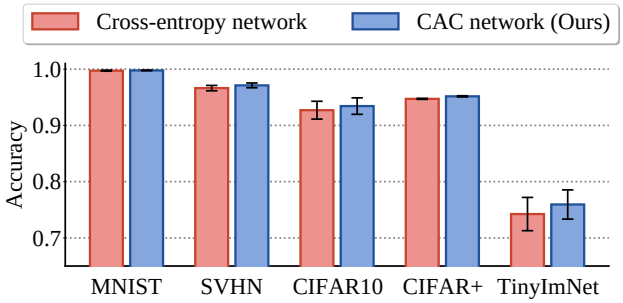


Figure 3: Classification accuracy (with standard deviation error bars) of a standard classifier trained with cross-entropy loss and our open set classifier trained with CAC loss, averaged over 5 trials with different known classes.

quality feature representation to separate known and unknown classes. When presented with a small number of known classes, such as only 4 for CIFAR+10/+50, the network may not be able to learn a rich feature representation that ensures known and unknown class inputs do not project to the same region in the logit space.

Despite this, our approach has other advantages over the C2AE approach. C2AE uses the reconstruction error from a class-conditioned auto-encoder to distinguish between known and unknown inputs. However, reconstruction error does not explicitly measure distance from the training data, and thus cannot guarantee that inputs far from the training data will be identified as unknown. It was also shown that autoencoders are able to reconstruct unknown class inputs with low error [6], and that reconstruction error can fail to distinguish known and unknown classes, particularly for more complex datasets [19]. These findings [6, 19] indicate that open set classifiers based on autoencoders perform well on simple datasets, but will not scale well to complex application domains. In contrast, our proposed method does not suffer from this limitation.

Maintaining classification accuracy: An open set classifier must not significantly degrade the classification accuracy achieved by a standard network with the same architecture. In Figure 3, we show that training with CAC loss maintains the closed set classification accuracy of a standard network. The standard network uses the same architecture but is trained with cross-entropy loss and uses the softmax score for classification. This result demonstrates that our improved open set performance does not compromise classification accuracy.

5.2. Ablation Studies

5.2.1 Comparison with Existing Distance Losses

We proposed CAC loss specifically for the task of *training* a distance-based open set classifier. However, other distance losses have been proposed for other computer vision tasks, e.g. metric learning [20] and facial recognition [23]. In this experiment, we compare the open set performance achieved when training with Center loss [23], Tuplet loss [20], the Anchor loss component $\mathcal{L}_A$ of CAC, and our proposed CAC loss. We train the same network architecture with each loss function and use our anchored class centres. Table 3 summarises the open set AUROC results for each of the distance losses.

Dataset	Center [23]	Tuplet [20]	$\mathcal{L}_A$ only (Ours)	CAC (Ours)
MNIST	**0.988**	0.957	0.979	0.987
SVHN	0.941	0.833	0.888	**0.942**
CIFAR10	0.786	0.739	0.751	**0.803**
CIFAR+10	0.854	0.844	0.804	**0.863**
CIFAR+50	0.863	0.837	0.816	**0.872**
TinyImageNet	0.765	0.717	0.749	**0.772**

Table 3: CAC provides better open set AUROC performance than the compared distance losses on nearly all the benchmark datasets.

CAC outperforms all other distance losses [20, 23] on SVHN, CIFAR10, CIFAR+10, CIFAR+50 and TinyImageNet, with Center loss [23] achieving second best performance. Center loss uses cross-entropy loss on the logits to *implicitly* encourage inputs to maximise distance to other class centres. In contrast, CAC *explicitly* forces this behaviour by applying Tuplet loss directly to the output distance vector. Interestingly, when used alone, our Anchor loss term and Tuplet loss cannot achieve the same performance as when they are combined to create CAC loss. This validates that both loss terms are important for distance-based open set classification, as together they simultaneously ensure minimised distance to the correct class centre as well as maximised distance to all other class centres.

5.2.2 Anchored versus Learnt Class Centres

In this section we investigate the benefits of using *anchored* class centres in the context of open set classification. While our work is the first to anchor class centres during the training process, previous distance losses such as Center loss [23] encourage clustering around class centres that are simultaneously *learnt* during training.

We compare the open set performance when training with learnt and anchored class centres, and repeat this experiment with CAC loss and Center loss [23].

To learn class centres, we use the approach described in [23]. Learning class centres with CAC required the additional use of cross-entropy loss for stability (see the supplementary material for more details).

In Table 4, we show that anchored class centres yield

Dataset	Center [23]		CAC (Ours)	
	Learnt	Anchored	Learnt	Anchored
MNIST	0.985	**0.988**	0.987	**0.987**
SVHN	0.937	**0.941**	**0.946**	0.942
CIFAR10	0.763	**0.786**	0.791	**0.803**
CIFAR+10	0.831	**0.854**	0.856	**0.863**
CIFAR+50	0.848	**0.863**	0.865	**0.872**
TinyImNet	0.738	**0.765**	0.764	**0.772**

Table 4: For both Center loss [23] and our proposed CAC loss, anchored class centres yields better open set AUROC than learnt class centres.

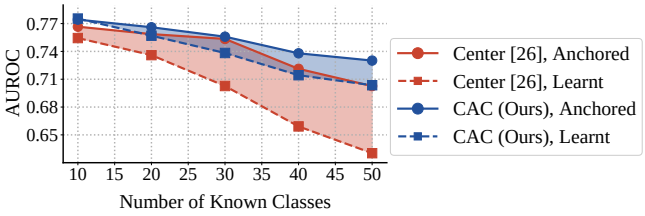


Figure 4: Anchored class centres perform better with increasing numbers of classes than learnt class centres. Results are averaged over 5 trials of random class splits on TinyImageNet. Openness of the problem is fixed at 18.35%.

better open set performance than learnt class centres, for both Center loss [23] and our proposed CAC loss. The performance difference between anchored and learnt centres is greatest for the object-based datasets (CIFAR10 variants and TinyImageNet), with an average 2.2% improvement for Center loss [23] and 0.85% for CAC.

Learning class centres during training relies on a stable learning signal from the images in each batch. However, CIFAR10 and TinyImageNet can exhibit considerable visual variations within each class, thus providing a potentially noisy learning signal for the class centre positions. By anchoring our class centres in the logit space, we eliminate this difficulty and allow for high performance on object-based datasets.

Learning class centres is even more difficult for tasks with large numbers of classes, as each batch will provide less data per class. To investigate this effect, we train with learnt and anchored class centres for Center loss [23] and CAC loss on *increasing* numbers of known classes, while keeping the openness of the open set task fixed at 18.35%. As we can see in Figure 4, open set performance of a network trained with learnt class centres degrades at a faster rate than a network trained with anchored class centres for both Center loss [23] and CAC loss.

In summary, we showed that training with anchored class centres yields better performance on object-based datasets and scaled better to datasets with larger numbers of training

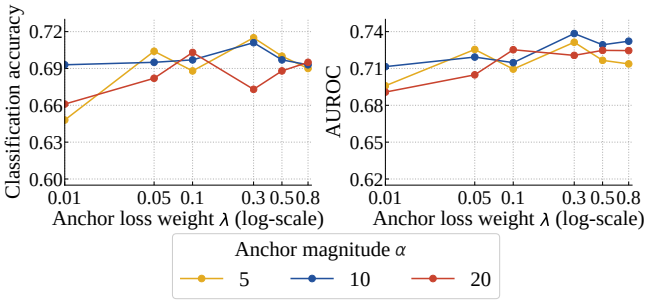


Figure 5: Effect of CAC anchor loss weight λ and anchored centre magnitude α on classification accuracy (left) and open set AUROC (right). Results are generated from 1 trial with a random split of known/unknown classes for TinyImageNet.

classes. We found this to be consistent for both tested loss functions. In addition, we observed training with anchored centres requires approximately half as many epochs to converge, speeding up the training process (more details in the supplementary).

5.2.3 Analysis of Hyperparameters of CAC loss

Our proposed CAC loss has two hyperparameters, the Anchor loss term weight α and the anchored centre magnitude λ, and their sensitivity is shown in Figure 5. With an Anchor loss weight $0.05 \leq \lambda \leq 0.8$ and an anchor magnitude $5 \leq \alpha \leq 20$, both the classification accuracy and open set AUROC vary slightly, by less than 4%.

6. Conclusions

The deployment of deep neural networks under open set conditions remains an important and difficult challenge for computer vision. Reliability and robustness in the presence of unknown class inputs is crucial for many safety-critical applications such as driverless cars or robotics. Beyond that, open set performance is also important for applications of computer vision in domains such as retail or augmented reality: open set errors that are a nuisance for the user slow down the rate of adoption and acceptance.

We introduced and demonstrated the benefits of anchored class centres and the novel Class Anchor Clustering loss for open set recognition. In the future, we hope to extend these ideas beyond image classification. Furthermore, we look forward to engaging with the community to develop new evaluation protocols and datasets beyond the simple ones commonly used in open set recognition, such as MNIST, SVHN, or even CIFAR10. Many practical applications rely on open set robustness, and we believe benchmark datasets should better reflect the complexity and richness of those real world applications.

References

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

[2] Abhijit Bendale and Terrance E Boult. Towards open set deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1563–1572, 2016.

[3] Anil Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distributions. *Bull. Calcutta Math. Soc.*, 35:99–109, 1943.

[4] Hermann Blum, Paul-Edouard Sarlin, Juan Nieto, Roland Siegwart, and Cesar Cadena. The fishyscapes benchmark: measuring blind spots in semantic segmentation. *arXiv preprint arXiv:1904.03215*, 2019.

[5] TE Boult, S Cruz, AR Dhamija, M Gunther, J Henrydoss, and WJ Scheirer. Learning and the unknown: Surveying steps toward open world recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9801–9807, 2019.

[6] Taylor Denouden, Rick Salay, Krzysztof Czarnecki, Vahdat Abdelzad, Buu Phan, and Sachin Vernekar. Improving reconstruction autoencoder out-of-distribution detection with mahalanobis distance. *arXiv preprint arXiv:1812.02765*, 2018.

[7] Akshay Raj Dhamija, Manuel Günther, and Terrance Boult. Reducing network agnostophobia. In *Advances in Neural Information Processing Systems (NeurIPS*, pages 9157–9168, 2018.

[8] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Machine Learning (ICML)*, 2017.

[9] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[10] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. 2010.

[11] Benjamin J Meyer and Tom Drummond. The importance of metric learning for robotic vision: Open set recognition and active learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 2924–2931. IEEE, 2019.

[12] Lawrence Neal, Matthew Olson, Xiaoli Fern, Weng-Keen Wong, and Fuxin Li. Open set learning with counterfactual images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 613–628, 2018.

[13] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[14] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 427–436, 2015.

[15] Poojan Oza and Vishal M Patel. C2ae: Class conditioned auto-encoder for open-set recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2307–2316, 2019.

[16] Qi Qian, Jiasheng Tang, Hao Li, Shenghuo Zhu, and Rong Jin. Large-scale distance metric learning with uncertainty. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8542–8550, 2018.

[17] Walter J Scheirer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E Boult. Toward open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1757–1772, 2013.

[18] Patrick Schlachter, Yiwen Liao, and Bin Yang. Open-set recognition using intra-class splitting. In *2019 27th European Signal Processing Conference (EUSIPCO)*, pages 1–5. IEEE, 2019.

[19] Alireza Shafaei, Mark Schmidt, and James J Little. A less biased evaluation of out-of-distribution sample detectors. *arXiv preprint arXiv:1809.04729*, 2018.

[20] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in neural information processing systems*, pages 1857–1865, 2016.

[21] 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.

[22] TinyImageNet. Tiny ImageNet Visual Recognition Challenge. https://tiny-imagenet.herokuapp.com/, Accessed: 2020-03-01.

[23] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *European conference on computer vision*, pages 499–515. Springer, 2016.

[24] Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krahenbuhl. Sampling matters in deep embedding learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2840–2848, 2017.

[25] Ryota Yoshihashi, Wen Shao, Rei Kawakami, Shaodi You, Makoto Iida, and Takeshi Naemura. Classification-reconstruction learning for open-set recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4016–4025, 2019.

[26] Sergey Demyanov Zongyuan Ge and Rahil Garnavi. Generative openmax for multi-class open set classification. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 42.1–42.12, 2017.