# DISTANCE-BASED BACKGROUND CLASS REGULARIZATION FOR OPEN-SET RECOGNITION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

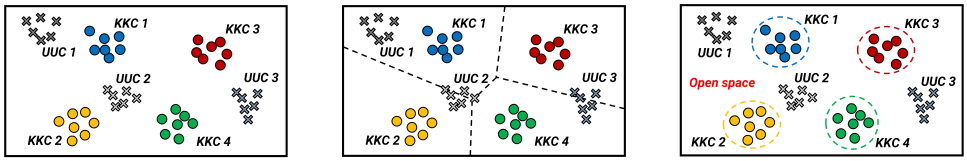
In open-set recognition (OSR), classifiers should be able to reject unknown-class samples while maintaining robust closed-set classification performance. To solve the OSR problem based on pre-trained Softmax classifiers, previous studies investigated offline analyses, e.g., distance-based sample rejection, which can limit the feature space of known-class data items. Since such classifiers are trained solely based on known-class samples, one can use background class regularization (BCR), which employs background-class data as surrogates of unknown-class data during training phase, to enhance OSR performance. However, previous regularization methods have limited OSR performance, since they categorized known-class data into a single group and then aimed to distinguish them from anomalies. In this paper, we propose a novel distance-based BCR method suitable for OSR, which limits the feature space of known-class data in a class-wise manner and then makes background-class samples located far away from the limited feature space. Instead of conventional Softmax classifiers, we use distance-based classifiers, which utilize the principle of linear discriminant analysis. Based on the distance measure used for classification, we design a novel regularization loss function that can contrast known-class and background-class samples while maintaining robust closed-set classification performance. Through our extensive experiments, we show that the proposed method provides robust OSR results with a simple inference process.

## 1 INTRODUCTION

In machine learning (ML), classification algorithms have achieved great success. Through recent advances in convolutional neural networks, their classification performance already surpassed the human-level performance in image classification (He et al., 2015). However, such algorithms are usually developed under a *closed-set* assumption, i.e., the class of each test sample is assumed to always belong to one of the pre-defined set of classes. Although this conventional assumption can be easily violated in real-world applications (classifiers can face unknown-class data), traditional algorithms are likely to force unknown-class samples to be classified into one of the known classes. To tackle this issue, the *open-set recognition (OSR)* problem (Scheirer et al., 2013) aims to properly classify unknown-class samples as "unknown" and known-class samples as one of the known classes.

According to the definition of OSR (Scheirer et al., 2013), it is required to properly limit the feature space of known-class data. To satisfy the requirement, various OSR methods were developed based on traditional ML models. Previously, Scheirer et al. (2014) calibrated the decision scores of support vector machines (SVMs). Based on the intuition that a large set of data samples of unknown classes can be rejected if those of known classes are accurately modeled, Jain et al. (2014) proposed $P_I$-SVM, which utilized the statistical modeling of known-class samples located near the decision boundary of SVMs. Afterward, Júnior et al. (2016) attempted to solve the OSR problem based on the principle of the nearest neighbors. Taking distribution information of data into account, Rudd et al. (2018) proposed the extreme value machine by utilizing the concept of margin distributions.

Since deep neural networks (DNNs) have robust classification performance by learning high-level representations of data, OSR methods for DNNs have received great attention. Based on the theoretical foundations used in traditional ML-based OSR methods, Bendale & Boult (2016) proposed the first OSR strategy for DNNs called Openmax, which calibrates the output logits of pre-trained Softmax classifiers. To improve Openmax, Yoshihashi et al. (2019) proposed the classification-reconstruction




(a) Latent feature space (b) Closed-set classification (c) Open-set classification

Figure 1: Given (a) a latent feature space, we demonstrate (b) closed-set and (c) open-set classification examples, where KKCs and UUCs are known and unknown classes, respectively.

learning to make robust latent feature vectors. Afterward, Oza & Patel (2019) proposed to exploit a class-conditioned autoencoder and use its reconstruction error to assess each input sample. Sun et al. (2020) employed several class-conditioned variational auto-encoders for generative modeling.

Although previous OSR methods using discriminative models applied *offline analyses* to pre-trained Softmax classifiers or employed complicated DNN models, they have limited performance since the classifiers were trained solely based on known-class samples. To mitigate the problem, one can use background class regularization (BCR) to achieve robust empirical results. However, previous BCR methods (Dhamija et al., 2018; Hendrycks et al., 2019; Liu et al., 2020) are insufficient to properly solve the OSR problem. To design an effective open-set classifier that can overcome the previous limitations, we propose a novel BCR method suitable for OSR, which uses a distance-based classifier and a novel loss function for regularization. We provide detailed description in the next section.

## 2 Preliminaries and Our Contributions

### 2.1 The Open-Set Recognition Problem

The OSR problem addresses the classification setting that can face test samples from classes unseen during training. In this setting, open-set classifiers aim to properly classify *known-class* samples while rejecting *unknown-class* samples simultaneously. A similar problem to OSR is out-of-distribution (OOD) detection (Hendrycks & Gimpel, 2017), which typically aims to reject data items drawn far away from the training data distribution. In previous OOD detection studies (Hendrycks & Gimpel, 2017; Liang et al., 2018; Lee et al., 2018a;b), OOD samples tend to be drawn from other datasets or even images of noise. In this paper, we aim to reject test data whose classes are unknown but related to the training data, which narrows down the scope of conventional OOD detection tasks.

Previously, Scheirer et al. (2013) introduced a formal definition of the OSR problem based on the notion of open-space risk $R_{\mathcal{O}}$, which is a relative measure of a positively labeled union of balls $\mathcal{S}_V$ and open space $\mathcal{O}$ located far from $\mathcal{S}_V$. Since labeling any data item in $\mathcal{O}$ incurs open-space risk, it is straightforward that a classifier cannot be a solution for the OSR problem if the classifier accepts data in infinitely wide regions, i.e., its open-space risk is unbounded ($R_{\mathcal{O}} = \infty$). The definition implies that essential requirements to solve the OSR problem are 1) *bounding open-space risk* and 2) *ideally balancing it with empirical risk* to maintain a low classification error rate.

Unlike traditional closed-set classifiers, open-set classifiers are required to limit the space of known-class data to bound their open-space risk. To ensure bounded open-space risk, Scheirer et al. (2014) proposed to formulate compact abating probability (CAP) models. The principle of CAP models is that if the support region of a classifier decays in all directions from the training data, thresholding the region will bound the classifier's open-space risk (Boult et al., 2019). As depicted in Figure 1, which compares traditional closed-set and open-set classification problems, properly building *class-wise* CAP models is an effective strategy to satisfy the two essential requirements of OSR.

### 2.2 Post-Classification Analysis for Pre-Trained Softmax Classifier

In this paper, we aim to solve the OSR problem by using a standard DNN-based classifier architecture $f$ as a latent feature extractor. Applying a fully-connected layer to $f$, a conventional Softmax classifier

computes the posterior probability of an input $\mathbf{x}$ belonging to the $c$-th known class by

$$P_s(y=c|\mathbf{x}) = \frac{\exp(\mathbf{w}_c^T f(\mathbf{x}) + b_c)}{\sum_{i=1}^{C} \exp(\mathbf{w}_i^T f(\mathbf{x}) + b_i)}, \tag{1}$$

where $c \in \{1, \cdots, C\}$, $f(\mathbf{x}) \in \mathbb{R}^n$ is the latent feature vector of $\mathbf{x}$, and $\mathbf{w}_c$ and $b_c$ are the weight and the bias for the $c$-th class, respectively. For pre-trained Softmax classifiers, Hendrycks & Gimpel (2017) proposed a baseline to detect anomalous samples, which imposes a threshold on the predictive confidence of Eq. (1). When using the baseline method to solve the OSR problem, one can estimate the class of each known-class sample and recognize unknown-class data by

$$\widehat{y} = \begin{cases} \arg\max_{c \in \{1,\cdots,C\}} P_s(y=c|\mathbf{x}), & \text{if } \max_{c \in \{1,\cdots,C\}} P_s(y=c|\mathbf{x}) \geq \tau, \\ C+1 \text{ (unknown class)}, & \text{otherwise.} \end{cases} \tag{2}$$

However, Eq. (2) cannot formally bound open-space risk since it only rejects test samples near the decision boundary of classifiers, thereby having infinitely wide regions of acceptance (Boult et al., 2019). Therefore, additional *post-classification analysis* methods using an auxiliary measure other than the Softmax probability are necessary to bound open-space risk, where distance measures have been widely used to build auxiliary CAP models in the latent feature space of $f$.

To construct class-wise CAP models, Openmax (Bendale & Boult, 2016) defined radial-basis decaying functions $\{s(\mathbf{x}, i)\}_{i=1}^{C}$, each of which measures the *class-belongingness* of $\mathbf{x}$ for the $c$-th class, in the latent feature space of $f$. For each $s(\mathbf{x}, c)$, the authors employed distance measures between $f(\mathbf{x})$ and an empirical class mean vector $\boldsymbol{\mu}_c$, e.g., $s(\mathbf{x}, c) = D_E^2(f(\mathbf{x}), \boldsymbol{\mu}_c) = (f(\mathbf{x}) - \boldsymbol{\mu}_c)^T (f(\mathbf{x}) - \boldsymbol{\mu}_c)$. To formulate effective CAP models based on $s(\mathbf{x}, c)$, they statistically modeled the distribution of $s(\mathbf{x}, c)$ for known-class data based on the extreme value theory (EVT) (Scheirer, 2017), which provides a theoretical foundation that the Weibull distribution is suitable for modeling known-class samples located far from the class mean vectors (extreme samples). To be specific, Openmax fits a Weibull distribution on extreme samples of the $c$-th class having the highest $D_E(f(\mathbf{x}), \boldsymbol{\mu}_c)$ values, where its cumulative distribution function (CDF) formulates the *probability of inclusion* $P_I(\mathbf{x}, c)$ (Jain et al., 2014; Rudd et al., 2018), i.e., $P_I(\mathbf{x}, c) = 1 - \texttt{WeibullCDF}$, which rapidly decays near extreme samples. Based on $P_I(\mathbf{x}, c)$, the decision rule of Eq. (2) can be calibrated to conduct robust OSR.

### 2.3 Generalized Open-Set Recognition and Background Class Regularization

Although they require additional inference procedures (e.g., EVT modeling) to bound the open-space risk of pre-trained Softmax classifiers, previous post-classification analyses may have limited OSR performance since the classifiers are trained solely based on known-class samples. To empirically obtain robust OSR results without complicated post-classification analyses, one can use the strategy of BCR at the training phase, which exploits background-class samples as surrogates of unknown-class samples. Geng et al. (2020) argued that the *generalized* OSR setting that utilizes background samples is still less explored and an important research direction to improve OSR performance. In the generalized OSR setting, classifiers should consider the following data classes among the infinite label space of all classes $\mathcal{Y}$ (Dhamija et al., 2018; Geng et al., 2020).

- Known known classes (KKCs; $\mathcal{K} = \{1, \cdots, C\} \subset \mathcal{Y}$) include distinctly labeled positive classes, which also have side information, where $\mathcal{U} = \mathcal{Y} \setminus \mathcal{K}$ is the entire unknown classes.
- Known unknown classes (KUCs; $\mathcal{B} \subset \mathcal{U}$) include background classes, e.g., labeled negative classes which are not necessarily grouped into a pre-defined set of KKCs $\mathcal{K}$.
- Unknown unknown classes (UUCs; $\mathcal{A} = \mathcal{U} \setminus \mathcal{B}$) represent the rest of $\mathcal{U}$, where UUC samples are not available at training time and have no side information, but occur at inference time.

Throughout this paper, we denote the corresponding datasets for the data classes as follows. $\mathcal{D}_{train}$ is a training set consisting of multiple pairs of a KKC data sample and the corresponding class label $y \in \{1, \cdots, C\}$. $\mathcal{D}_{test}^k$ and $\mathcal{D}_{test}^u$ are test sets of KKCs and UUCs, respectively. $\mathcal{D}_b$ is a background dataset of KUCs. Then, a loss function that trains classifiers with BCR can be formulated by

$$\mathcal{L} = \mathcal{L}_{cf} + \lambda \mathcal{L}_{bg} = \mathbb{E}_{(\mathbf{x}^k, y) \sim \mathcal{D}_{train}} \left[ -\log P_s(y|\mathbf{x}^k) + \lambda \cdot \mathbb{E}_{\mathbf{x}^b \sim \mathcal{D}_b} \left[ f_{reg}\left(\mathbf{x}^k, y, \mathbf{x}^b\right) \right] \right], \tag{3}$$

where $\mathcal{L}_{cf}$ and $\mathcal{L}_{bg}$ are the losses for classification and BCR, respectively, and $\lambda$ is a hyperparameter that balances $\mathcal{L}_{cf}$ and $\mathcal{L}_{bg}$. For $\mathcal{L}_{bg}$, previous studies designed their own $f_{reg}$, where Dhamija et al.

(2018) proposed the objectosphere loss for OSR, and Hendrycks et al. (2019) and Liu et al. (2020) employed the uniformity and the energy losses for OOD detection, respectively.

In this paper, we tackle the following limitations of the previous BCR methods.

- In the previous BCR methods, $\mathcal{L}_{bg}$ were designed to make normal data and anomalies more distinguishable in terms of the corresponding anomaly scores. Since they categorized normal samples into a single group (do not consider the classes) in $\mathcal{L}_{bg}$, the previous methods cannot be suitable for rejecting UUC samples located relatively close to KKC data and maintaining robust closed-set classification results.
- The previous methods using the decision rule of Eq. (2) (e.g., objectosphere (Dhamija et al., 2018) and uniformity (Hendrycks et al., 2019)) cannot bound open-space risk. Although one can use post-classification analyses to bound open-space risk, trained latent feature space are likely to be inappropriate for using another metric such as distance measures.
- To increase the gap between KKC and KUC samples in terms of latent feature magnitude and energy value in the objectosphere (Dhamija et al., 2018) and the energy (Liu et al., 2020) losses, respectively, it is necessary to find proper margin parameters for each dataset.

### 2.4 Our Contributions

Based on standard DNN-based classifier architecture $f$, we aim to design open-set classifiers having a simple and effective inference process by using the principle of BCR. To overcome the limitations described in Section 2.3, we use distance-based classifiers and propose a novel BCR strategy based on the framework of Eq. (3). In the following, we summarize our proposed strategy.

- Instead of applying fully-connected layers to feature extractors $f$, we use the principle of linear discriminant analysis (LDA) (Murphy, 2012) to classify each input based on a distance measure. By imposing a threshold on the distance measure as in Eq. (2), such distance-based classifiers can bound their open-space risk without additional offline analyses.
- To enhance the OSR performance of distance-based classifiers, we propose a novel distance-based BCR strategy. For $\mathcal{L}_{bg}$, we design a loss function that does not require data-dependent margin parameters. Our method first limits the feature space of KKC data by formulating class-wise boundaries based on the concept of the probability of inclusion, and then forces KUC data to be located outside the entire boundaries. By imposing additional regularization on each KKC data item to be located inside the corresponding class-wise boundary, our proposed method can also maintain robust closed-set classification performance.

## 3 Proposed Method

### 3.1 Distance-Based Classification Models

To train a robust open-set classifier as we described in Section 2.4, we employ

$$P_d(y=c|\mathbf{x}) = \frac{P(y=c)\cdot\mathcal{N}(f(\mathbf{x})|\boldsymbol{\mu}_c,\mathbf{I})}{\sum_{i=1}^{C}P(y=i)\cdot\mathcal{N}(f(\mathbf{x})|\boldsymbol{\mu}_i,\mathbf{I})} = \frac{P(y=c)\cdot\exp\left(-D_E^2(f(\mathbf{x}),\boldsymbol{\mu}_c)\right)}{\sum_{i=1}^{C}P(y=i)\cdot\exp\left(-D_E^2(f(\mathbf{x}),\boldsymbol{\mu}_i)\right)} \quad (4)$$

as an alternative of Eq. (1), where Eq. (4) applies the principle of LDA to $f(\mathbf{x})$. In Eq. (4), we use an identity covariance matrix $\mathbf{I}$ and $P(y=c)=C^{-1}$ for all $c$ to model KKC data. Then, Eq. (4) classifies each $\mathbf{x}$ by computing $D_E^2(f(\mathbf{x}),\boldsymbol{\mu}_c)=(f(\mathbf{x})-\boldsymbol{\mu}_c)^T(f(\mathbf{x})-\boldsymbol{\mu}_c)$, the Euclidean distance between $f(\mathbf{x})$ and $\boldsymbol{\mu}_c\in\mathbb{R}^n$, which is the class mean vector for the $c$-th class. It is noteworthy that the *distance-based classifier* of Eq. (4) is similar to the prototypical network (Snell et al., 2017) and the nearest class mean classifier (Mensink et al., 2012). Instead of computing empirical $\boldsymbol{\mu}_c$ as in the previous studies, we define the class mean vectors as *anchors* before training and fix them during the training process. To initialize class mean vectors, we randomly sample each $\boldsymbol{\mu}_c$ from the standard Gaussian distribution for all $c$, by following the approach of Izmailov et al. (2020).

In Eq. (4), we use LDA since it is a simple and popular strategy to define a generative classifier in a discriminative classifier, where the underlying assumption of LDA is that each class-conditioned latent feature vector is drawn from a unimodal Gaussian distribution. Alternatively, one can enforce each class-conditioned latent feature vector to follow a unimodal Gaussian distribution via generative

models. Recently, Izmailov et al. (2020) proposed a generative classifier called FlowGMM, which jointly trains a flow-based generative model (Dinh et al., 2017; Kingma & Dhariwal, 2018) with Eq. (4). Using FlowGMM, the authors also conducted semi- and fully-supervised classification tasks by using pre-fixed $\boldsymbol{\mu}_c$, $P(y=c)=C^{-1}$ for all $c$, and $\boldsymbol{\Sigma}_c=\mathbf{I}$, and obtained successful results. Since it requires additional loss functions or DNNs specialized to generative modeling, enforcing the distribution of class-conditioned $f(\mathbf{x})$ to be $\mathcal{N}(f(\mathbf{x})|\boldsymbol{\mu}_c,\mathbf{I})$ is out of the scope of this paper.

At inference time, each KKC sample $\mathbf{x}$ can be classified by $\widehat{y}=\arg\min_{c\in\{1,\cdots,C\}} D_E^2(f(\mathbf{x}),\boldsymbol{\mu}_c)$. It is noteworthy that thresholding $D_E^2(f(\mathbf{x}),\boldsymbol{\mu}_c)$, which is the same metric used for classification, can bound the open-space risk of a distance-based classifier by formulating class-wise CAP models. Therefore, one can properly conduct OSR without additional post-classification analyses as follows:

$$\widehat{y}=\begin{cases}\arg\min_{c\in\{1,\cdots,C\}} D_E^2(f(\mathbf{x}),\boldsymbol{\mu}_c), & \text{if } \max_{c\in\{1,\cdots,C\}} -D_E^2(f(\mathbf{x}),\boldsymbol{\mu}_c)\geq\tau,\\ C+1 \text{ (unknown class)}, & \text{otherwise.}\end{cases} \tag{5}$$

### 3.2 Background Class Regularization for Distance-based Classifiers

To obtain robust OSR performance based on distance-based classifiers using the simple inference of Eq. (5), we aim to design a BCR method, which uses $\mathcal{D}_{train}$ and $\mathcal{D}_b$ as surrogates of $\mathcal{D}_{test}^k$ and $\mathcal{D}_{test}^u$ at training time, respectively. Although it cannot provide any information of $\mathcal{D}_{test}^u$, $\mathcal{D}_b$ can be effective to limit the latent feature space of KKC data, while reserving space for UUC samples. Note that the decision rule of Eq. (5) conducts both closed-set classification and unknown-class rejection based on a single distance measure $D_E^2(f(\mathbf{x}),\boldsymbol{\mu}_c)$. Thus, it is intuitive that the primary objective of our BCR method is to make KUC samples located far away from $\boldsymbol{\mu}_i$ for all $i\in\{1,\cdots,C\}$.

Before we illustrate our BCR method, we first introduce hypersphere classifiers (HSCs) (Ruff et al., 2020), whose concept was also used in (Liznerski et al., 2021). An HSC conducts anomaly detection by using a DNN-based feature extractor $g$, where its anomaly score is the Euclidean distance between a mean vector $\boldsymbol{\mu}$ and the latent feature vector $g(\mathbf{x})$ of each input $\mathbf{x}$. When training the HSC model, the authors used normal and background data, $\mathcal{D}_{train}$ and $\mathcal{D}_b$, respectively, and a loss function

$$\mathcal{L}_{hsc}=\mathbb{E}_{\mathbf{x}^k\sim\mathcal{D}_{train}}\left[h\left(D_E^2\left(g\left(\mathbf{x}^k\right),\boldsymbol{\mu}\right)\right)\right]-\mathbb{E}_{\mathbf{x}^b\sim\mathcal{D}_b}\left[\log\left(1-\exp\left(-h\left(D_E^2\left(g\left(\mathbf{x}^b\right),\boldsymbol{\mu}\right)\right)\right)\right)\right], \tag{6}$$

which aims to decrease the Euclidean distances between normal samples $\mathbf{x}^k$ and $\boldsymbol{\mu}$ while increasing the distances for background samples $\mathbf{x}^b$. In Eq. (6), $h(x)=\sqrt{x+1}-1$, which implies that the Euclidean distance $D_E^2(g(\mathbf{x}),\boldsymbol{\mu})$ is scaled into the range of $(0,1]$ via $\exp(-h(D_E^2(g(\mathbf{x}),\boldsymbol{\mu})))$.

It is straightforward that the decision rule of Eq. (5) for distance-based classifiers uses the principle of HSCs in a class-wise manner, where the original HSC formulates its decision boundary for anomaly detection as a hypersphere having a constant radius. In other words, the class-wise HSC for the $c$-th class determines whether a test input belongs to the $c$-th class by computing $D_E^2(f(\mathbf{x}),\boldsymbol{\mu}_c)$, where the input is determined as an unknown-class sample if the entire class-wise HSCs reject the data item. Thus, a proper BCR strategy for distance-based classifiers should force each KUC sample $\mathbf{x}^b$ to be rejected by the entire class-wise HSCs (increase $D_E^2(f(\mathbf{x}^b),\boldsymbol{\mu}_i)$ for all $i$). Since it is inefficient to consider the entire class-wise HSCs to regularize $\mathbf{x}^b$ at each iteration, we approximate the process by only taking the *closest* class-wise HSC into account (increase $\min_{i\in\{1,\cdots,C\}} D_E^2(f(\mathbf{x}^b),\boldsymbol{\mu}_i)$).

Although one can adopt Eq. (6) to formulate $\mathcal{L}_{bg}$ for distance-based classifiers, scaling $D_E^2(f(\mathbf{x}),\boldsymbol{\mu}_c)$ into $(0,1]$ via $\exp(-h(D_E^2(f(\mathbf{x}),\boldsymbol{\mu}_c)))$ can be inappropriate to guarantee sufficient spaces for KKC data and simultaneously move KUC data far away from the limited space of KKC data, since $\exp(-h(D_E^2(f(\mathbf{x}),\boldsymbol{\mu}_c)))$ rapidly decays near $\boldsymbol{\mu}_c$. To overcome the issue, we propose to design $\mathcal{L}_{bg}$ based on the principle of the probability of inclusion (Jain et al., 2014; Bendale & Boult, 2016).

### 3.3 Probability of Inclusion and Class-Inclusion Loss

As we described in Section 2.2, the probability of inclusion builds effective CAP models, since it is designed to rapidly decay near extreme data, i.e., $P_I(\mathbf{x},c)\approx 1$ in the region that a majority of class-$c$ KKC samples are located. In the following, we describe how to utilize the principle of the probability of inclusion when training distance-based classifiers, and then formulate a loss function for BCR.

Based on pre-trained Softmax classifiers, Openmax (Bendale & Boult, 2016) formulated the probability of inclusion via EVT modeling of latent features at inference time. However, it is intractable to use such EVT-based analyses during the training phase of $f$, since it is inappropriate to limit the space of KKC data by analyzing extreme samples at each training iteration. Thus, we use the underlying assumption of LDA to formulate $P_I(\mathbf{x}, c)$, since we define the training and the inference processes of distance-based classifiers based on the principle of LDA. Under the assumption that each class-$c$ latent feature vector is assumed to be drawn from a unimodal Gaussian distribution $\mathcal{N}(f(\mathbf{x})|\boldsymbol{\mu}_c, \mathbf{I})$, the Euclidean distance $D_E^2(f(\mathbf{x}), \boldsymbol{\mu}_c)$, a simplified version of the Mahalanobis distance, can be assumed to follow the Chi-square distribution having the degree of freedom $n$. Then, we have

$$P\left(D_E^2(f(\mathbf{x}), \boldsymbol{\mu}_c) = t\right) = \frac{t^{\frac{n}{2}-1}}{2^{\frac{n}{2}} \cdot \Gamma(n/2)} \cdot \exp\left(-\frac{t}{2}\right), \tag{7}$$

where $t \geq 0$, $\Gamma(\cdot)$ is the Gamma function, and $n$ is equivalent to the dimension of $f(\mathbf{x})$.

As previous studies (Jain et al., 2014; Bendale & Boult, 2016; Rudd et al., 2018) formulated the probability of inclusion by computing the CDF of the Weibull distribution, we define our $P_I(\mathbf{x}, c)$ as

$$P_I(\mathbf{x}, c) = 1 - \int_0^{D_E^2(\mathbf{x},c)/2} \frac{t^{\frac{n}{2}-1}}{\Gamma(n/2)} \cdot \exp(-t)\, dt = \frac{\Gamma(n/2, D_E^2(f(\mathbf{x}), \boldsymbol{\mu}_c)/2)}{\Gamma(n/2)} \tag{8}$$

by using the CDF of Eq. (7), where $\Gamma(\cdot, \cdot)$ is the upper incomplete Gamma function. It is noteworthy that Eq. (8) can be easily computable via `igammac` function in PyTorch (Paszke et al., 2019).

Based on $\mathcal{D}_{train}$, $\mathcal{D}_b$, and $P_I(\mathbf{x}, c)$ of Eq. (7), the primary objective of our BCR, which is to make each KUC sample located far away from the closest class-wise HSC, can be achieved by using a loss function $\mathcal{L}_{bg,u} = \mathbb{E}_{\mathbf{x}^b \sim \mathcal{D}_b}[-\log(1 - \max_{i \in \{1,\cdots,C\}} P_I(\mathbf{x}^b, i))]$. To compare $P_I(\mathbf{x}, c)$ and $P_H(\mathbf{x}, c) = \exp(-h(D_E^2(f(\mathbf{x}), \boldsymbol{\mu}_c)))$, which was used in Eq. (6), we plot $P_I(\mathbf{x}, c)$ and $P_H(\mathbf{x}, c)$ in Figure 2 with respect to $||f(\mathbf{x}) - \boldsymbol{\mu}_c||$ by assuming $n = 128$. The figure implies that unlike $P_H(\mathbf{x}, c)$, our $P_I(\mathbf{x}, c)$ can effectively assign space for KKC data and force KUC samples to be located outside the space.

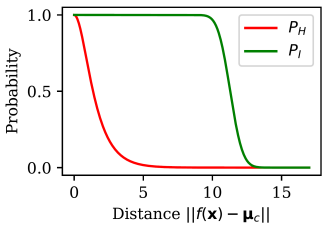



Figure 2: $P_H$ and $P_I$ (Ours)

At training time, note that each $P_I(\mathbf{x}, c) = 0.5$ builds an auxiliary decision boundary between the $c$-th class KKC sample and the others, where $\mathcal{L}_{bg,u}$ ensures a majority of KUC samples located outside the entire class-wise boundaries. However, $\mathcal{L}_{bg,u}$ can be insufficient to achieve robust UUC rejection and closed-set classification results, since it does not control correctly classified KKC samples to be located inside the corresponding class-wise boundaries. Thus in addition to $\mathcal{L}_{cf}$, we impose a regularization $\mathcal{L}_{bg,k}$ on KKC data to *maintain* closed-set classification performance and enhance the gap between KKC and KUC data in terms of the Euclidean distance. By formulating $\mathcal{L}_{bg,k} = \mathbb{E}_{(\mathbf{x}^k, y) \sim \mathcal{D}_{train}}[-\mathbb{1}(y = c^*) \cdot \log(P_I(\mathbf{x}^k, c^*))]$, where $c^* = \arg\max_{i \in \{1,\cdots,C\}} P_I(\mathbf{x}^k, i)$, we define our BCR loss as $\mathcal{L}_{bg} = \mathcal{L}_{bg,k} + \mathcal{L}_{bg,u}$ and call $\mathcal{L}_{bg}$ the *class-inclusion loss*. To sum up,

$$\mathcal{L} = \mathcal{L}_{cf} + \lambda \mathcal{L}_{bg} = \mathcal{L}_{cf} + \lambda(\mathcal{L}_{bg,k} + \mathcal{L}_{bg,u}) \tag{9}$$

is employed as our loss function, where $\mathcal{L}_{cf} = \mathbb{E}_{(\mathbf{x}^k, y) \sim \mathcal{D}_{train}}[-\log P_d(y|\mathbf{x}^k)]$ for Eq. (4).

## 4 Experiments

In this section, we compared our class-inclusion loss for distance-based classifiers to the objectosphere (Dhamija et al., 2018), the uniformity (also widely known as OE) (Hendrycks et al., 2019), and the energy (Liu et al., 2020) losses for Softmax classifiers. Through our extensive experiments, we aim to show that whether our proposed framework provides competitive UUC rejection results in comparison with the previous BCR methods while keeping robust closed-set classification results.

### 4.1 Experimental Settings

To assess the OSR performance of each method, we used the following two experimental settings.

`S1)` We split a single dataset into KKCs and UUCs, where we employed the KKCs in the training set as $\mathcal{D}_{train}$, and the KKCs and the UUCs in the test set as $\mathcal{D}_{test}^k$ and $\mathcal{D}_{test}^u$, respectively.

Table 1: Comparison with the previous BCR methods in the first setting of our OSR experiments.

| Experiments | Accuracy (↑) | AUROC (↑) | OSCR (↑) |
|---|---|---|---|
| | Objectosphere / Uniformity / Energy / Class-inclusion (Ours) | | |
| SVHN | 0.968 / 0.966 / 0.972 / **0.972** | 0.914 / 0.908 / 0.894 / **0.919** | 0.809 / 0.785 / 0.760 / **0.828** |
| CIFAR10 | 0.964 / 0.964 / 0.956 / **0.973** | 0.942 / 0.923 / 0.933 / **0.946** | 0.851 / 0.814 / 0.807 / **0.869** |
| CIFAR+10 | 0.958 / 0.969 / 0.949 / **0.976** | 0.945 / 0.950 / 0.936 / **0.960** | 0.839 / 0.867 / 0.808 / **0.880** |
| CIFAR+50 | | 0.944 / 0.942 / 0.937 / **0.955** | 0.837 / 0.837 / 0.808 / **0.865** |
| TinyImageNet | 0.778 / 0.779 / 0.715 / **0.801** | 0.755 / 0.771 / 0.727 / **0.784** | 0.484 / 0.488 / 0.357 / **0.492** |

S2) By using the training and the test sets of a single dataset as $\mathcal{D}_{train}$ and $\mathcal{D}_{test}^{k}$, respectively, we employed the test set of another dataset relatively close to $\mathcal{D}_{train}$ as $\mathcal{D}_{test}^{u}$.

When evaluating the OSR performance of each method, we first measured the closed-set classification accuracy. To quantify the performance of UUCs data rejection, we also measured the area under the receiver operating characteristic curve (AUROC). Furthermore, we used the open-set classification rate (OSCR) measure by quantifying the correct classification rate at the false positive rate of $10^{-1}$. For in-depth explanation of the OSCR measure, readers are referred to (Dhamija et al., 2018).

#### 4.1.1 Open-Set Recognition - Setting 1

Following the experiment protocol in (Neal et al., 2018), which were also employed in (Oza & Patel, 2019; Sun et al., 2020), we conducted experiments by using the following standard image datasets: SVHN (Netzer et al., 2011), CIFAR10 & CIFAR100 (Krizhevsky, 2009), and TinyImageNet (Le & Yang, 2015). As a background dataset $\mathcal{D}_b$, we used the ImageNet dataset (Russakovsky et al., 2015), which was also employed as a large-scale background dataset in (Li & Vasconcelos, 2020). To ensure that $\mathcal{D}_b$ and $\mathcal{D}_{test}^{u}$ are disjoint, we used only the remaining classes of the ImageNet dataset, which are not in TinyImageNet, as KUCs. In what follows, we describe KKCs and UUCs in each experiment.

**SVHN, CIFAR10** For the SVHN and CIFAR10 datasets, each of which consists of images of 10 classes, each dataset was randomly partitioned into 6 KKCs and 4 UUCs.

**CIFAR+10, CIFAR+50** For the CIFAR+$M$ experiments, we used 4 randomly selected classes of CIFAR10 as KKCs and $M$ randomly selected classes of CIFAR100 as UUCs.

**TinyImageNet** For experiments with a large number of classes, we randomly selected 20 classes of TinyImageNet as KKCs and then used the remaining 180 classes as UUCs.

#### 4.1.2 Open-Set Recognition - Setting 2

Adopting the experiment settings in (Yoshihashi et al., 2019) and (Liang et al., 2018), we employed the entire classes of a dataset as KKCs for CIFAR10 & CIFAR100. For UUC dataset, TinyImageNet, LSUN (Yu et al., 2015), and iSUN (Xu et al., 2015) were selected. TinyImageNet and LSUN consists of 10,000 test samples each, where the samples in each dataset were resized (RE) or cropped (CR) into the size $32 \times 32$. The iSUN dataset contains 8,925 test samples and they were also resized into the size of $32 \times 32$. The modified datasets were introduced in (Liang et al., 2018), where the datasets can be obtained in their Github repository[1]. For $\mathcal{D}_b$, we used the same dataset used in Setting 1.

### 4.2 Training Details

For $f$, we employed the Wide-ResNet (WRN) (Zagoruyko & Komodakis, 2016) and then used its penultimate layer $f(\mathbf{x}) \in \mathbb{R}^n$ for the latent feature vector of each input sample $\mathbf{x}$. For CIFAR10 and TinyImageNet, we used WRN 40-2 with a dropout rate of $0.3$, where WRN 28-10 was employed for CIFAR100 with the same dropout rate. For SVHN, we used WRN 16-4 with a dropout rate of $0.4$.

In the loss function $\mathcal{L} = \mathcal{L}_{cf} + \lambda \mathcal{L}_{bg}$, we set the mini-batch sizes of KKC training samples and KUC samples to $128$ for the entire BCR methods. We kept $\lambda$ as a constant during training, i.e., each $f$ was trained with the corresponding BCR method *from scratch*. To select hyperparameters and margin parameters of the previous regularization methods, we followed the official implementations in their

[1] https://github.com/facebookresearch/odin

Table 2: Comparison with the previous methods in the second setting of our OSR experiments. The corresponding classification accuracy values are reported in the first column.

| $\mathcal{D}_{train}/\mathcal{D}_{test}^{k}$ | $\mathcal{D}_{test}^{u}$ | AUROC (↑) | OSCR (↑) |
|---|---|---|---|
| | | Objectosphere / Uniformity / Energy / Class-inclusion (Ours) | |
| CIFAR10<br>0.940 / 0.939 / 0.925 / **0.947** | ImageNet-CR | **0.988** / 0.986 / 0.981 / 0.987 | 0.929 / 0.928 / 0.894 / **0.931** |
| | ImageNet-RE | 0.979 / 0.984 / 0.972 / **0.984** | 0.923 / 0.926 / 0.886 / **0.927** |
| | LSUN-CR | **0.994** / 0.990 / 0.989 / 0.993 | 0.938 / 0.931 / 0.904 / **0.940** |
| | LSUN-RE | 0.985 / 0.988 / 0.984 / **0.990** | 0.928 / 0.931 / 0.897 / **0.935** |
| | iSUN | 0.985 / 0.989 / 0.984 / **0.991** | 0.928 / 0.932 / 0.896 / **0.936** |
| | **Average** | 0.986 / 0.987 / 0.982 / **0.989** | 0.929 / 0.930 / 0.895 / **0.934** |
| CIFAR100<br>0.727 / 0.735 / 0.705 / **0.779** | ImageNet-CR | 0.886 / 0.929 / **0.925** / 0.922 | 0.641 / 0.686 / 0.652 / **0.693** |
| | ImageNet-RE | 0.815 / 0.910 / **0.934** / 0.902 | 0.572 / 0.674 / 0.658 / **0.683** |
| | LSUN-CR | **0.967** / 0.931 / 0.901 / 0.965 | 0.685 / 0.680 / 0.643 / **0.751** |
| | LSUN-RE | 0.844 / 0.930 / **0.959** / 0.945 | 0.608 / 0.695 / 0.684 / **0.731** |
| | iSUN | 0.842 / 0.923 / **0.954** / 0.943 | 0.603 / 0.689 / 0.680 / **0.729** |
| | **Average** | 0.871 / 0.925 / 0.935 / **0.935** | 0.621 / 0.685 / 0.663 / **0.717** |

Github repositories [2,3,4]. For SVHN, CIFAR10, CIFAR100, and TinyImageNet, we trained the corresponding classifiers for 80, 100, 200, and 200 epochs, respectively, where we used the stochastic gradient descent for optimization. For SVHN and the other datasets, we used initial learning rates of 0.01 and 0.1, respectively, and a cosine learning rate decay (Loshchilov & Hutter, 2016). We also used the learning rate warm-up strategy for the first 5 epochs of each training process.

### 4.3 Experimental Results

For the experiment settings described in Section 4.1, we reported the corresponding OSR results of our framework and the previous methods in Tables 1 and 2. All the reported values were averaged over five randomized trials, by randomly sampling seed, split of KKCs and UUCs, and class mean vectors. In the tables, ↑ indicates that it is better to have larger values for the corresponding measure.

**Setting 1** For the first setting of our experiments, Table 1 compares our proposed BCR methods for distance-based classifiers with the previous methods designed for Softmax classifiers. The results in the table demonstrate that our proposed method obtained robust UUC rejection results, which were superior to the results of the previous approaches. It is noteworthy that our method achieved higher classification accuracy values, which were critical in acquiring better OSR results in terms of OSCR, than the previous methods. Such results imply that the proposed framework can find an effective solution for the OSR problem, by satisfying the two essential requirements described in Section 2.1: bounding open-space risk and ideally balancing it with empirical risk.

**Setting 2** For our proposed BCR method and the previous methods, the corresponding OSR results in the second setting of our experiments are reported in Table 2. By using CIFAR10 and CIFAR100 as KKC datasets, we found that the proposed model achieved the highest closed-set classification accuracy, which is consistent with the experimental results in the first setting `S1`. Furthermore, by averaging the AUROC and the OSCR values over the various UUC datasets, the table shows that our model outperformed the previous methods in UUC sample rejection and OSR.

To sum up, the experimental results of `S1` and `S2` show that our proposed method can successfully train a robust open-set classifier by training a distance-based classifier with BCR from scratch.

### 4.4 Ablation Study and Discussions

For ablation study, we explored whether our method maintains the classification accuracy of each vanilla closed-set classifier. Also, we further investigated the OSR performance of our method by using various $\lambda$ in our loss function and different class mean vector initialization/training steps. In

[2] `https://github.com/Vastlab/Reducing-Network-Agnostophobia`
[3] `https://github.com/hendrycks/outlier-exposure`
[4] `https://github.com/wetliu/energy_ood`

Table 3: Comparison with the previous BCR method in text classification experiments.

| $\mathcal{D}_{train}/\mathcal{D}_{test}^{k}$ | $\mathcal{D}_{test}^{u}$ | AUROC (↑) | AUPR (↑) | FPR95 (↓) | OSCR (↑) |
|---|---|---|---|---|---|
| | | Uniformity / Class-inclusion (Ours) | | | |
| 20 Newsgroups<br>0.719 / **0.749** | Multi30k | 0.997 / **0.997** | **0.998** / 0.997 | **0.002** / 0.010 | 0.715 / **0.745** |
| | WMT16 | **0.997** / 0.996 | **0.997** / 0.995 | **0.010** / 0.016 | 0.715 / **0.742** |
| | IMDB | 0.805 / **0.999** | 0.692 / **0.999** | 0.367 / **0.003** | 0.585 / **0.747** |
| | **Average** | 0.933 / **0.997** | 0.896 / **0.997** | 0.126 / **0.010** | 0.672 / **0.745** |

addition, we compared our class-inclusion loss to the uniformity loss in text classification tasks.

**Classification accuracy** By training $f$ for CIFAR10 by using $\mathcal{L} = \mathcal{L}_{cf}$ in Eq. (1) and Eq. (4), we obtained the classification accuracy of $0.942$ and $0.943$, respectively. Also, we obtained the accuracy of $0.774$ for CIFAR100, where the results imply that our proposed method can maintain and even enhance the classification performance of the corresponding vanilla closed-set classifiers.

**Selecting $\lambda$** We conducted additional experiments in the settings of `S1` and `S2` with various $\lambda$, where we used $\lambda \in \{0.1, 0.5, 1, 5, 10\}$. By using various $\lambda$, we observed that $\lambda = 5$ showed the best OSR performance in the SVHN, CIFAR10, and CIFAR + $M$ experiments of `S1` and the CIFAR10 experiments of `S2`. Also, $\lambda = 1$ and $\lambda = 0.5$ showed the best results in the TinyImageNet experiments of `S1` and the CIFAR100 experiments of `S2`. Since the other $\lambda$ values did not significantly degrade OSR results, one can flexibly select $\lambda$ that can balance empirical and open-space risks.

**Class mean vectors** In Section 3.1, we mentioned that each class mean vector was sampled from the standard Gaussian distribution. Although one can use data-dependent initialization approaches, which are 1) train a SoftMax classifier and then use its empirical mean vectors for distance-based classifiers and 2) compute mean vectors from the initial feature representations of a distance-based classifier, such approaches showed worse results than the random initialization method.

Furthermore, we also attempted to update class mean vectors by re-computing empirical class mean vectors every few epochs or setting the class mean vectors as learnable parameters. We also observed that updating class mean vectors yield worse results than using fixed class mean vectors.

**Text classification** For text classification, we used 20 Newsgroups and WikiText103 for KKCs and KUCs, respectively, and trained a simple GRU model (Cho et al., 2014) for $f$ as in (Hendrycks et al., 2019). For UUCs, we used Multi30K, WMT16, and IMDB. Since the margin parameters of the objectosphere and the energy losses selected for image classification were not suitable for the text classification, we tested the uniformity loss for comparison. In Table 3, we present the results, where we additionally reported the area under the precision-recall curve (AUPR) and the false-positive rate at $95\%$ true-positive rate (FPR95). As it outperformed the uniformity loss in image classification tasks, our method also showed significantly better performance in text classification experiments.

## 5 Concluding Remarks

In this paper, we proposed a novel BCR method to train open-set classifiers that can provide robust OSR results with a simple inference step. By employing distance-based classifiers with the principle of LDA, we designed the class-inclusion loss for BCR, which effectively limits the feature space of KKC data in a class-wise manner and regularizes UUC samples to be located far away from the limited space, based on the principle of probability of inclusion. Through our extensive experiments and ablation study, we show that our method can achieve robust UUC rejection performance, while maintaining high closed-set classification accuracy. In this paper, we aim to improve the reliability of modern DNN-based classifiers. We hope our work to enhance reliability and robustness in various classification applications, by providing a novel methodology of handling UUC samples.

**Future works** When designing our class-inclusion loss for distance-based classifiers, we exploit the class-conditioned Gaussian assumption of LDA by assuming identity covariance matrices. For future works, one can integrate our proposed method with generative models that can build a class-conditioned Gaussian feature space or employ additional regularization or training methods that can learn a better formulation for covariance matrices. Furthermore, KUC data generation or resampling techniques can be another line of future research to acquire a compact and effective $\mathcal{D}_b$.

## 6 ETHICS STATEMENT

Although there are many beneficial applications of classifiers and anomaly detectors, they may lead to inadvertent discrimination and encoding societal biases in decision-making systems. This is one of the cases where ethical considerations depend strongly on the specific applications.

We use several standard datasets including the ImageNet dataset. Although they have been widely used in research, some of the datasets, created from images available on the web, include images of people, raising ethical questions related to human-derived data, such as encoding of biases and under-representation issues. The creators of ImageNet are also aware of some of these concerns, and have been attempted to address them for the past few years (https://image-net.org/update-sep-17-2019.php).

## 7 REPRODUCIBILITY STATEMENT

We describe details about distance-based classifiers, our loss function, and the corresponding training process in Sections 3.1, 3.3, and 4.2. Also, Section 4.1 provide the list of the datasets, which are used in this paper. Our source code and the corresponding instructions will be released.

## REFERENCES

Abhijit Bendale and Terrance E Boult. Towards open set deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1563–1572, 2016.

Terrance E Boult, Steve Cruz, Akshay Raj Dhamija, M Gunther, James Henrydoss, and Walter J Scheirer. Learning and the unknown: Surveying steps toward open world recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 9801–9807, 2019.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *The 2014 Conference on Empirical Methods in Natural Language Processing*, 2014.

Akshay Raj Dhamija, Manuel Günther, and Terrance Boult. Reducing network agnostophobia. *Advances in Neural Information Processing Systems*, 31:9157–9168, 2018.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. In *International Conference on Learning Representations*, 2017.

Chuanxing Geng, Sheng-jun Huang, and Songcan Chen. Recent advances in open set recognition: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1026–1034, 2015.

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*, pp. 770–778, 2016.

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

Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations*, 2019.

Pavel Izmailov, Polina Kirichenko, Marc Finzi, and Andrew Gordon Wilson. Semi-supervised learning with normalizing flows. In *International Conference on Machine Learning*, pp. 4615–4630, 2020.

Lalit P Jain, Walter J Scheirer, and Terrance E Boult. Multi-class open set recognition using probability of inclusion. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 393–409. Springer, 2014.

Pedro Ribeiro Mendes Júnior, Terrance E Boult, Jacques Wainer, and Anderson Rocha. Specialized support vector machines for open-set recognition. *arXiv preprint arXiv:1606.03802*, 2016.

Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pp. 10215–10224, 2018.

Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. 2015.

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

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in Neural Information Processing Systems*, 31:7167–7177, 2018b.

Yi Li and Nuno Vasconcelos. Background data resampling for outlier-aware classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13218–13227, 2020.

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

Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in Neural Information Processing Systems*, 33, 2020.

Philipp Liznerski, Lukas Ruff, Robert A Vandermeulen, Billy Joe Franks, Marius Kloft, and Klaus-Robert Müller. Explainable deep one-class classification. In *International Conference on Learning Representations*, 2021.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. Metric learning for large scale image classification: Generalizing to new classes at near-zero cost. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 488–501. Springer, 2012.

Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

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)*, pp. 613–628, 2018.

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

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*, pp. 2307–2316, 2019.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

Ethan M Rudd, Lalit P Jain, Walter J Scheirer, and Terrance E Boult. The extreme value machine. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3):762–768, 2018.

Lukas Ruff, Robert A Vandermeulen, Billy Joe Franks, Klaus-Robert Müller, and Marius Kloft. Rethinking assumptions in deep anomaly detection. *arXiv preprint arXiv:2006.00339*, 2020.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115 (3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Walter J Scheirer. Extreme value theory-based methods for visual recognition. *Synthesis Lectures on Computer Vision*, 7(1):1–131, 2017.

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.

Walter J Scheirer, Lalit P Jain, and Terrance E Boult. Probability models for open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2317–2324, 2014.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 30, pp. 4080–4090, 2017.

Xin Sun, Zhenning Yang, Chi Zhang, Keck-Voon Ling, and Guohao Peng. Conditional gaussian distribution learning for open set recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13480–13489, 2020.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

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

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*, pp. 4016–4025, 2019.

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

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

# A APPENDIX

This supplementary section presents additional open-set recognition (OSR) results for the experimental settings described in Section 4.1. In additional experiments, we employed CIFAR10 to make $\mathcal{D}_{train}$. For the first setting of our OSR experiments (S1), we split the 10 classes of CIFAR10 into 6 known known classes (KKCs) and 4 unknown unknown classes (UUCs). In the second setting (S2), we used CIFAR10 and another dataset as KKCs and as UUCs, respectively. For the first setting of our experiments, we report quantitative results in the form of **(Accuracy / AUROC / OSCR)**.

## A.1 LATENT FEATURE SPACE VISUALIZATION

By visualizing feature spaces via t-SNE (Van der Maaten & Hinton, 2008), we compared Softmax classifiers and distance-based classifiers, and their regularized versions in S1 and S2. We selected ImageNet-RE for UUC data in S2, since the results of Table 2 imply that the dataset is more difficult to be recognized by open-set classifiers than the others. In each t-SNE result, black dots indicate UUC samples. For the other colors, each distinct color represents the corresponding class of KKCs.

### A.1.1 VANILLA CLASSIFIERS (NO BACKGROUND-CLASS REGULARIZATION)

(a) S1 - Softmax (b) S1 - Distance (c) S2 - Softmax (d) S2 - Distance

Figure 3: t-SNE results of vanilla Softmax classifiers and distance-based classifiers.

Figure 3 depicts the t-SNE results of vanilla Softmax and distance-based classifiers, which are trained only with $\mathcal{L} = \mathcal{L}_{cf}$, in S1 and S2. The visualization results imply that in comparison with the UUC data used in S2, it would be much more difficult to distinguish UUC samples in S1 from KKCs.

### A.1.2 BACKGROUND CLASS REGULARIZED CLASSIFIERS

(a) S1 - Uniformity (b) S1 - Ours (c) S2 - Uniformity (d) S2 - Ours

Figure 4: t-SNE results of regularized Softmax classifiers and distance-based classifiers.

Figure 4 presents the t-SNE results of Softmax and distance-based classifiers trained with BCR ($\mathcal{L} = \mathcal{L}_{cf} + \lambda\mathcal{L}_{bg}$) in S1 and S2. For the Softmax and distance-based classifiers, we employed the uniformity loss (Hendrycks et al., 2019) and our class-inclusion loss as $\mathcal{L}_{bg}$, respectively. In comparison with Figure 3, Figure 4 shows that such regularization techniques assist classifiers to learn robust latent feature representations in distinguishing UUCs from KKCs, especially in S1. Also, as mentioned in Section 2.3, the figure shows that the latent feature space regularized by the uniformity loss can yield inaccurate results in distance-based post-classification analysis. By applying the Openmax approach (Bendale & Boult, 2016) to the model regularized by the uniformity loss in S1 (Figure 4. (a)), we obtained the OSR performance of **(0.923 / 0.886 / 0.752)**, which is worse than the original result **(0.964 / 0.923 / 0.814)** and the result of our framework **(0.973 / 0.946 / 0.869)**.

### A.2 QUANTITATIVE RESULTS WITH RESNET-18 ARCHITECTURE

In Section 4, we used the Wide-ResNet (WRN) architectures as feature extractors for the image datasets. To further investigate the effectiveness of our method, we used another standard classifier architecture, ResNet-18 (He et al., 2016). In `S1`, we obtained quantitative results of **(0.963 / 0.945 / 0.844)**, **(0.966 / 0.950 / 0.845)**, **(0.967 / 0.945 / 0.848)**, and **(0.971 / 0.947 / 0.850)** for the regularization methods using the objectosphere (Dhamija et al., 2018), the uniformity (Hendrycks et al., 2019), the energy (Liu et al., 2020), and our class-inclusion losses, respectively. Such quantitative OSR results and the results in Table 4 imply that our method can outperform the previous BCR methods with ResNet-18, as we observed in the experiments using the WRN architectures.

Table 4: Comparison with the previous methods in the second setting of our OSR experiments using ResNet-18. The corresponding classification accuracy values are reported in the first column.

| $\mathcal{D}_{train}/\mathcal{D}^{k}_{test}$ | $\mathcal{D}^{u}_{test}$ | AUROC (↑) | OSCR (↑) |
|---|---|---|---|
| | | Objectosphere / Uniformity / Energy / Class-inclusion (Ours) | |
| CIFAR10<br>0.937 / 0.949 / 0.933 / **0.951** | ImageNet-CR | 0.982 / 0.979 / 0.983 / **0.987** | 0.932 / 0.928 / 0.917 / **0.941** |
| | ImageNet-RE | 0.977 / 0.982 / 0.975 / **0.988** | 0.918 / 0.934 / 0.909 / **0.945** |
| | LSUN-CR | 0.991 / 0.984 / 0.987 / **0.993** | 0.934 / 0.935 / 0.919 / **0.946** |
| | LSUN-RE | 0.987 / 0.987 / 0.986 / **0.990** | 0.932 / 0.938 / 0.918 / **0.942** |
| | iSUN | 0.987 / 0.986 / 0.987 / **0.994** | 0.932 / 0.938 / 0.919 / **0.946** |
| | **Average** | 0.985 / 0.984 / 0.984 / **0.991** | 0.930 / 0.935 / 0.916 / **0.944** |