One-class Text Classification with Multi-modal Deep Support Vector Data Description

Chenlong Hu¹, Yukun Feng¹, Hidetaka Kamigaito¹, Hiroya Takamura^{1,2} and Manabu Okumura¹

¹Tokyo Institute of Technology

²National Institute of Advanced Industrial Science and Technology (AIST) {huchenlong, yukun, kamigaito, oku}@lr.pi.titech.ac.jp

takamura@pi.titech.ac.jp

Abstract

This work presents multi-modal deep SVDD (mSVDD) for one-class text classification. By extending the uni-modal SVDD to a multiple modal one, we build mSVDD with multiple hyperspheres, that enable us to build a much better description for target one-class data. Additionally, the end-to-end architecture of mSVDD can jointly handle neural feature learning and one-class text learning. We also introduce a mechanism for incorporating negative supervision in the absence of real negative data, which can be beneficial to the mSVDD model. We conduct experiments on Reuters and 20 Newsgroup datasets, and the experimental results demonstrate that mSVDD outperforms uni-modal SVDD and mSVDD can get further improvements when negative supervision is incorporated.

1 Introduction

One-Class Classification (OCC), a special classification problem, aims to learn a model on the basis of training samples only from *one* class. The learned model is expected to make an accurate description of the class (so called *target* or *normal*) and then to distinguish the *target* from samples for negative classes during testing (Moya et al., 1993; Tax, 2002). The one-class classification problem has arisen in many real-world applications, including anomaly or novelty detection (Roberts, 1999; Chandola et al., 2010; Gupta et al., 2013), bioinformatics (Alashwal et al., 2006), and especially computer vision (Rodner et al., 2011; Ruff et al., 2018).

One-class text classification would be beneficial to the scenario where anomalous text contents (e.g., web pages, spam emails) (Yu et al., 2004) need to be detected, and only a positive training corpus is available. One of the early work on oneclass text classification is Manevitz and Yousef (2001), who implemented versions of one-class support vector machines (OC-SVM) (Schölkopf et al., 2001) and showed good performances over the *Reuters* dataset (Dumais et al., 1998). OC-SVM and support vector data description (SVDD) (Tax and Duin, 2004) are *boundary*-based methods (Tax, 2002). Both try to describe the target data using a boundary. SVDD learns an optimal hypersphere with the minimum radius to include the most target data, while OC-SVM builds a hyperplane to maximally separate the data points from the origin where outlier examples lie around.

Reconstruction-based approaches, including AutoEncoder (Jacobs, 1995) and principal component analysis (PCA) (Bishop, 1995), which aim to learn a more compact representation for the description of target data. The compact representation could be a set of prototypes or subspaces obtained by optimizing a reconstruction error on the target training data.

Regarding the features for representing text in OCC, document-to-word co-occurrence matrices or hand-crafted features have been commonly used in most of the previous work (Manevitz and Yousef, 2001, 2007; Kumaraswamy et al., 2015). Pre-trained vectors have been popular for many NLP tasks (Mikolov et al., 2013; Bengio et al., 2003). The recent *context vector data description (CVDD)*, proposed by Ruff et al. (2019), fully uses word embedding knowledge and a neural network structure to process one-class classification problems.

Ruff et al. (2018) introduced *deep support vector data description (deep SVDD)*, a fully unsupervised method for deep one-class classification for image data. *Deep SVDD* learns to extract the common factors of target training samples with a neural network to minimize the radius of a hypersphere that encloses the network representations of the data. The learned hypersphere, with a center c and a neural feature transformer $\phi(x)$, can be an end-to-end feature learning and one-class classification model.

Target data samples may have distinctive distributions that are located in different regions. Therefore, uni-modal *deep SVDD* with one hypersphere may not be enough to describe the target samples. In this work, we extend *deep SVDD* to multiple modes, where each mode describes the target samples from a distinctive aspect. Given our *multimodal deep SVDD*, *mSVDD* in short, we can create an ensemble set of hyperspheres with different centers to build a better one-class model. Ghafoori and Leckie (2020) proposed *deep multi-sphere SVDD* (*DMSVDD*), a similar but different work from ours. We will also discuss the relationship between the two and compare them in the experiments.

In one-class classification, only samples from the target class are available for training, while the model needs to discriminate between the target class and other classes in testing. Due to the unavailability of training samples from negative classes, it is hard for the one-class models to learn effective discrimination information, especially for *mSVDD* with a multi-layer neural structure. In this study, we also propose an architecture for improving the discrimination ability of *mSVDD* by incorporating *negative supervision*. Specifically, we define two kinds of losses, *contrastive* and *triplet*, for joint training with the objective function of *mSVDD*, which is expected to enhance the discriminative power of *mSVDD*.

In summary, the main contributions of this work are as follows. 1) We propose a general one-class neural learning framework, called *mSVDD*, to extend the uni-modal *deep SVDD* to end-to-end multimodal. 2) We also prove that three one-class models, *deep SVDD*, *DMSVDD*, and *CVDD*, are all special cases of the *mSVDD* model. 3) We propose two approaches for effectively incorporating negative supervision information to improve the performance of the proposed *mSVDD*.

2 Preliminaries

Before describing our *mSVDD*, we first introduce SVDD (Tax and Duin, 2004) and its extension, *deep SVDD* (Ruff et al., 2018).

2.1 SVDD

SVDD is a support vector learning method for oneclass classification. It aims at constructing an optimal boundary in a feature space that includes almost all normal target data, given only the target training samples, $T = \{x_1, ..., x_n\}, x_i \in \mathcal{X}$, where $n \in \mathbb{N}$ is the size of the training data, and \mathcal{X} is a compact subset of \mathbb{R}^d . The main idea of SVDD is to optimize a hypersphere with a center cand radius R, that encloses the majority of the data. SVDD solves the following quadratic problem:

$$\min_{R,c,\boldsymbol{\xi}} \quad R^2 + C \sum_i \xi_i \tag{1}$$

s.t. $\|\boldsymbol{x}_i - \boldsymbol{c}\|_2^2 \le R^2 + \xi_i, \ \xi_i \ge 0, \quad \forall i = 1, ..., n,$

where ξ_i is a slack variable for allowing a flexible boundary. *C* is a regularization parameter, that is usually represented by $\frac{1}{\nu n}$, where $\nu \in (0, 1]$ is a parameter that controls the tradeoff between the radius of the hypersphere and the penalties ξ_i .

Several efforts have been proposed to extend SVDD with multi-spheres. Hao and Lin (2007) was early work to use multi-sphere SVDD, which was used for multi-class tasks. For one-class tasks, (Xiao et al., 2009) used multi-sphere SVDD to encode multi-distribution target data. Two more efforts have been proposed by (Le et al., 2010, 2013), which found the optimal solution by an iterative algorithm consisting of the following two steps: 1) calculate radii and centers, and 2) calculate the assignments of data to centers.

While one limitation of SVDD, along with its extensions, would be that it has to perform hand-crafted feature engineering (Pal and Foody, 2010), the limitation could be solved by incorporating neural models into SVDD.

2.2 Deep SVDD

Deep SVDD (Ruff et al., 2018) is an end-to-end deep neural model that not only optimizes the SVDD objective loss but also learns a neural feature transformation. Given target training samples $T = \{x_1, ..., x_n\}$, deep SVDD first transforms instance x into a data point of the output feature space with ϕ , which is a multi-layer neural network of $L \in \mathbb{N}$ layers with parameters $\mathcal{W} = \{\mathbf{W}^1, ..., \mathbf{W}^L\}$. Deep SVDD defines two kinds of loss functions:

Soft-boundary deep SVDD :

$$\frac{1}{\nu n} \sum_{i} \max(0, \|\phi(\boldsymbol{x}_{i}; \mathcal{W}) - \boldsymbol{c}\|_{2}^{2} - R^{2}) + R^{2} + \frac{\lambda}{2} \sum_{l} \|\boldsymbol{W}^{l}\|_{F}^{2}$$

$$(2)$$



Figure 1: mSVDD with two modes. ϕ is a neural network. Fivestars denote a center, and black points denote positive target samples, while triangles denote negative outliers that need to be rejected by hyperspheres.

The first penalty term is for samples lying outside the sphere, i.e., when the distance of x_i to the center, $\|\phi(x_i; W) - c\|_2^2$, is greater than radius R after the transformation by network ϕ . The above loss also regularizes the radius and neural weight parameters in the second term. As with SVDD, parameter $\nu \in (0, 1]$ adjusts the tradeoff between the radius of the hypersphere and the points outside the hypersphere.

Schölkopf et al. (2001) proved that, in singleclass classification, ν is the upper bound of the fraction of anomalies, and the lower bound of the fraction of training samples being anomalies or on the optimal boundary. Ruff et al. (2018) proved that this ν -property still holds for uni-modal softboundary *deep SVDD*.

Another simplified objective that minimizes the mean distance of all positive training samples to the center, the *one-class* form, can be defined as follows:

One-class deep SVDD (simplified form):

$$\frac{1}{n}\sum_{i} \|\phi(\boldsymbol{x}_{i}; \mathcal{W}) - \boldsymbol{c}\|_{2}^{2} + \frac{\lambda}{2}\sum_{l} \|\boldsymbol{W}^{l}\|_{F}^{2} \quad (3)$$

Here, we can rewrite both the above in a unified form:

$$\mathcal{L}_{DSVDD} = C \sum_{i} [\|\phi(\boldsymbol{x}_{i}; \mathcal{W}) - \boldsymbol{c}\|_{2}^{2} - \beta]_{+} \\ + \beta + \frac{\lambda}{2} \sum_{l} \|\boldsymbol{W}^{l}\|_{F}^{2},$$
⁽⁴⁾

where $[\cdot]_+ = max\{0, \cdot\}, \beta \in \{0, R^2\}$ and regularization parameter $C \in \{\frac{1}{n}, \frac{1}{\nu n}\}$ correspond to the two types of forms.

3 Multi-Modal Deep SVDD

In this section, we present our *mSVDD*, a method for deep one-class classification. Unlike a unimodal model with a hypersphere, *mSVDD* uses a set of hyperspheres to describe target class data and to reject samples from negative classes. Figure 1 shows the general idea of mSVDD with two modes. Consider that we have m modes, each of which is described by a hypersphere M_j with center c_j and radius R_j ; mSVDD uses each M_j to describe a distinctive aspect of the target class and then ensemble them. This ensembled deep mSVDD model could provide better descriptions for the target data.

As with deep SVDD, given target training samples $T = \{x_1, ..., x_n\}$, *mSVDD* first transforms instance x into a data point of the output feature space with ϕ , where ϕ is a deep neural network of $L \in \mathbb{N}$ layers with parameters $\mathcal{W} = \{ \mathbf{W}^1, ..., \mathbf{W}^L \}$. In contrast to deep SVDD, mSVDD uses m hyperspheres to include almost all of the target data with the minimum radii, i.e., $\frac{1}{m} \sum_{j=1}^{m} R_j^2$. As in kernel SVDD and deep SVDD, it should also punish points lying outside the sphere, i.e., if the distance of x to the center $\boldsymbol{c}, \|\phi(\boldsymbol{x}; \mathcal{W}) - \boldsymbol{c}\|_2^2$, is greater than radius R. Since we have a set of hyperspheres $M = \{M_1, ..., M_m\}$, one choice would be to punish x with respect to each hypersphere by adding $\sum_{j} \max\left(0, \|\phi\left(\boldsymbol{x}; \mathcal{W}\right) - \boldsymbol{c}_{j}\|_{2}^{2} - R_{j}^{2}\right)$ to the loss function. However, the above penalty term is very hard, where one sample should satisfy each *j*-th constraint corresponding to M_j . Therefore, we loosen the constraint. Given non-negative attention weight α_{ij} for x_i to each M_j , the penalty term can be computed as the weighted average over mconstraints. Now, only one ensembled constraint is required, i.e., the sum of radii is greater than the sum distance to the center. Formally, we can define our *mSVDD* objective as follows:

$$\frac{1}{\nu n} \sum_{i} \max(0, \sum_{j} \alpha_{ij}(\|\phi(\boldsymbol{x}_{i}; \mathcal{W}) - \boldsymbol{c}_{j}\|_{2}^{2} - R_{j}^{2})) + \frac{1}{m} \sum_{j} R_{j}^{2} + \frac{\lambda}{2} \sum_{l} \|\boldsymbol{W}^{l}\|_{F}^{2}, \qquad (5)$$

where $\frac{1}{m} \sum_{j=1}^{m} R_j^2$ is the regularization term for radii from all *m* hyperspheres to get a closer boundary around the target data. This form can be seen as *mSVDD* with weighted soft-boundary constraints, which we call *soft-boundary mSVDD*.

Although the ν -property, mentioned in Section 2.2, does not hold true for our multi-modal case as it is in general, it is still true when the attention weight α_{ij} is constant for different hyperspheres. This will give us an intuition on the role of ν .

Proposition 1. ¹ The ν -property holds if we set equal attention weight to each hypersphere: i.) ν is an upper bound for the fraction of outlier samples and ii.) ν is a lower bound for the fraction of training samples being rejected or on the optimal boundary.

3.1 One-class mSVDD (simplified from)

As in *deep SVDD*, we also have the simplified from and called: *one-class*² *mSVDD*. If we assume that the majority of the training data is not anomalous, then the radius can be ignored and we can define the simplified *mSVDD* as follows:

$$\frac{1}{n}\sum_{i}\sum_{j}\alpha_{ij}\|\phi\left(\boldsymbol{x}_{i};\mathcal{W}\right)-\boldsymbol{c}_{j}\|_{2}^{2}+\frac{\lambda}{2}\sum_{l}\|\boldsymbol{W}^{l}\|_{F}^{2},$$
(6)

where the attention weight α_{ij} will be kept, while the penalty of radius R is deleted.

3.2 Unified Form of mSVDD

We can write the two variants of *mSVDD* (i.e., *soft-boundary mSVDD* and simplified *one-class mSVDD*) in a unified form:

$$\mathcal{L}_{\text{mSVDD}} = C \sum_{i} \left[\sum_{j} \alpha_{ij} \left(\|\phi(\boldsymbol{x}_{i}; \mathcal{W}) - \boldsymbol{c}_{j}\|_{2}^{2} - \beta_{j} \right) \right] + \frac{1}{m} \sum_{j} \beta_{j} + \frac{\lambda}{2} \sum_{l} \|\boldsymbol{W}^{l}\|_{F}^{2},$$
(7)

where $\beta_j \in \{0, R_j^2\}$. $\beta_j = 0$ corresponds to simplified *one-class mSVDD*, and $\beta_j = R_j^2$ corresponds to *soft-boundary mSVDD*. For \boldsymbol{x}_i to the *j*-th hypersphere, attention weight α_{ij} should be inversely proportional to its distance to center \boldsymbol{c}_j . Thus, we define: $\alpha_{ij} = \exp(d(\boldsymbol{x}_i, \boldsymbol{c}_j)/\delta)) / \sum_{k=1}^{m} \exp(d(\boldsymbol{x}_i, \boldsymbol{c}_k)/\delta)$, where $\delta < 0$ is a temperature hyperparameter.

3.3 Discussions on relationships between mSVDD and other models

Relationship with Uni-modal Deep SVDD Their relationship is obvious and can be summarized by the following proposition.

Proposition 2. Deep SVDD is a special case of the unified form of mSVDD with one hypersphere used.

Proof. Obviously, mSVDD becomes uni-modal (Ruff et al., 2018) if we use only one hypersphere, i.e., m = 1.

Relationship between mSVDD and CVDD CVDD (Ruff et al., 2019) is a one-class model for text data. In CVDD, each training sample x_i (i.e., a text) is represented by r self-attention feature vectors $S_i = (s_{i1}, ..., s_{ir})$ (Lin et al., 2017). CVDDuses a group of r context vectors $C = (c_1, ..., c_r)$ to describe the target one-class data, where $c_k \in \mathbb{R}^p$. CVDD tries to reduce the one-to-one reconstruction distance between feature vectors S_i and context vectors C. The loss can be defined as:

$$\mathcal{L}_{CVDD} = \frac{1}{n} \sum_{i} \sum_{k} \sigma_{ik} d(\boldsymbol{c}_{k}, \boldsymbol{s}_{ik}), \quad (8)$$

where $d(c_k, s_{ik})$ computes the distance, and σ_{ik} denotes the attention weight. The following proposition implies the close connection between two to learn one-class text problem.

Proposition 3. *CVDD is a very special case of one-class mSVDD when mSVDD is applied to text-based tasks under certain conditions.*

+ Proof. W.L.O.G., rewrite the loss function of oneclass mSVDD in a simplified form for each sample as follows:

$$\mathcal{L}_{mSVDD} = \frac{1}{n} \sum_{i} \sum_{j} \alpha_{ij} \|\phi_j(\boldsymbol{x}_i; \mathcal{W}) - \boldsymbol{c}_j\|^2$$
$$= \frac{1}{n} \sum_{i} \sum_{j} \sigma_{ij} d(\boldsymbol{x}_i, \boldsymbol{c}_j) \approx \mathcal{L}_{CVDD},$$

where we drop the regularization terms for weights of ϕ and radii, and set m = r, $\sigma_{ij} = \alpha_{ij}$, $d(\boldsymbol{x}_i, \boldsymbol{c}_j) = \|\phi_j(\boldsymbol{x}_i; \mathcal{W}) - \boldsymbol{c}_j\|^2$. $\phi(\boldsymbol{x}_i; \mathcal{W})$ has to be a self-attention neural model, $\phi_j(\boldsymbol{x}_i; \mathcal{W})$ is the *j*-th feature vector of sample \boldsymbol{x}_i , and \boldsymbol{c}_j is the *j*-th context vector of target samples. Now the loss functions of *CVDD* and *one-class mSVDD* are almost the same.

Relationship between mSVDD and DMSVDD *DMSVDD* (Ghafoori and Leckie, 2020) also uses multi-hyperspheres to extend SVDD. The loss function of *DMSVDD* is as follows:

$$\mathcal{L}_{\text{DMSVDD}} = \frac{1}{\nu n} \sum_{i} [\|\phi(\boldsymbol{x}_{i}; \mathcal{W}) - \boldsymbol{c}_{i^{*}}\|_{2}^{2} - R_{i^{*}}^{2}]_{+} \\ + \frac{1}{K} \sum_{k} R_{k}^{2} + \frac{\lambda}{2} \sum_{l} \|\boldsymbol{W}^{l}\|_{F}^{2}, \quad (9)$$

¹The proofs of propositions can be found in the appendix. ²The term "*one-class*" is used following Ruff et al. (2018). Note that all the models discussed in this paper are one-class models.

where K is the number of hyperspheres³, c_{i^*} is the *nearest center* of sample x_i and R_{i^*} is its radius.

Proposition 4. *DMSVDD can be seen as a hardversion of soft-boundary mSVDD if we set the attention weight in some way.*

Proof. In the calculation of the attention weight for *mSVDD* with $\alpha_{ij} = \frac{\exp(d(\boldsymbol{x}_i, \boldsymbol{c}_j/\delta))}{\sum_{k=1}^{m} \exp(d(\boldsymbol{x}_i, \boldsymbol{c}_k)/\delta)}$, the temperature parameter δ could influence the assignment of center c_k . If we set $\delta \to 0^-$, the above formula acts as the *argmin* operation.⁴ In this case, $\alpha_{ii*} = 1$ if $i* = \arg\min_{k=1,...,K} d(\boldsymbol{x}_i, \boldsymbol{c}_k)$, and 0 otherwise. Now, we can get the form of *DMSVDD* from *soft-boundary mSVDD* (Eq. 5) through the adjustment of the attention weight. Therefore, we can prove that *DMSVDD* is also a special case of *mSVDD*.

The above relation illustrates key difference between them: *DMSVDD* puts value on one hypersphere with the largest weight.

3.4 Summarizing mSVDD

We summarize the proposed *mSVDD* in accordance with the discussions presented above. The proposed *multi-modal deep SVDD (mSVDD)* learns a compact description of one-class data with multiple hyperspheres. *mSVDD* is also a generic framework that includes *deep SVDD*, *CVDD*, *and DMSVDD* if the corresponding conditions are met.

4 Multi-Modal Deep SVDD with Negative Supervision

In this section, we incorporate negative supervision into the training of *mSVDD*. The SVDD-related models are usually trained with only positive samples from the target one-class, while, if negative samples (samples which should be *rejected*) are available, the models can be extended to train with them to improve the description (Tax, 2002). Note that these samples are *not* necessarily required to be from "*real*" negative class. In our experiment, we use some external data as the *pseudo*-negative samples.

Given a set of extended training samples $T' = \{(x_1, y_1), ..., (x_{n'}, y_{n'})\}$, where the first *n* samples are labeled $y_i = 1$, denoting positive,

whereas the others are labeled $y_i = 0$, which denotes negative samples that should be rejected by mSVDD. Our mSVDD is represented with m hyperspheres and is formulated as M = $\{M_1(c_1, R_1), \dots, M_m(c_m, R_m)\}$. It is required that the positive samples should be inside the mhyperspheres, while the negative samples should lie outside. Given training samples composed of positive and a negative samples, we can first get their corresponding distances to each center c_i . The goal of optimization should be to pull the positive samples closer the center and to push the negative ones away. Formally, we define the distance between one sample x_i and one center c_i as $d_{ij} = d(\boldsymbol{x}_i, \boldsymbol{c}_j) = \|\phi(\boldsymbol{x}_i; \mathcal{W}) - \boldsymbol{c}_j\|_2^2$. There are usually two types of losses to obtain the discriminative loss.

Contrastive type: The contrastive-type loss directly optimizes the distance by encouraging the distance between a positive sample and a center to be smaller, while it forces the larger distance to a negative sample:

$$\mathcal{L}_{Con_d}^{(ij)} = y_i [d_{ij} - R_j^2]_+ + (1 - y_i) \left[R_j^2 - d_{ij} \right]_+,$$
(10)

where R_j^2 can be seen as a *margin* (or *threshold*) with a function that prevents too much effort from being wasted in enlarging/reducing distances (Hadsell et al., 2006).

Triplet type: The triplet-type loss is defined for a pair of positive sample x_i and negative sample $x_{i'}$. If we consider center c_j as an *anchor* representative of target data, the triplet loss punishes only when d_{ij} , the distance from x_i to c_j , is greater than $d_{i'j}$, the distance from $x_{i'}$ to c_j , with a margin $\tau > 0$:

$$\mathcal{L}_{Tri_d}^{(ii'j)} = \left[d_{ij} - d_{i'j} + \tau \right]_+.$$
(11)

For clarity, Eqs. 10 and 11 show only the two types of losses for one hypersphere. Multi-modal version can be obtained by sum operation over $j \in \{1, ..., m\}$.

The triplet loss forces only positive samples to be closer to the center than negative samples, and the contrastive loss requires only keeping the distances for negative samples above the radius. These two types of objectives are easy to achieve, especially when we assume that negative samples are "*not real.*" This can result in failing to make a full use of negative supervision. Therefore, we will reformulate both \mathcal{L}_{Trid} and \mathcal{L}_{Cond} .

³In DMSVDD, K changes dynamically. However, we ignore this difference and focus on the comparison of the models.

 $^{^{4}\}delta$ approaches 0 from the negative side.

4.1 Reformulating Contrastive and Triplet Losses

Normalization layer In neural models with the contrastive or triplet loss, it is a common strategy to normalize the feature representations of samples for training stability (Schroff et al., 2015; Wang et al., 2017). Therefore, we apply the normalization to the input vectors: $\hat{x} = x/\sqrt{\sum |x_i|^2 + \epsilon}$, where $\epsilon > 0$ is a value avoiding division by zero.

Reformulation Given a center c_j and positive and negative samples, we can use the probability form in the optimization objective, rather than the two *non-probabilistic* ones: \mathcal{L}_{Con_d} and \mathcal{L}_{Tri_d} . We introduce $p(y_i = 1 | \boldsymbol{x}_i, \boldsymbol{c}_j)$, which is the probability that a hypersphere with center \boldsymbol{c}_j accepts the sample \boldsymbol{x}_i , and define it as follows:

$$p(y_i = 1 | \boldsymbol{x}_i, \boldsymbol{c}_j) = \sigma(s \hat{\boldsymbol{f}}_i^T \hat{\boldsymbol{c}}_j), \qquad (12)$$

where $\sigma(x) = \frac{1}{1+\exp(-x)}$, $f_i = \phi(x_i; W)$ denotes the feature output vector of x_i , and s is a scale hyper-parameter for preventing failed convergence (Wu et al., 2018) after the normalization. For each sample x_i , c_j acts as a *pseudo*-weight vector for the classification of the *j*-th hypersphere of *mSVDD*. Thus, given $p(y_i = 1 | x_i, c_j)$, the probability of a sample being accepted by hypersphere M_j , we can reformulate the two discriminative losses with the probability.

Contrastive type loss:

$$\mathcal{L}_{Con}^{(ij)} = -y_i \log p(y_i = 1 | \boldsymbol{x}_i, \boldsymbol{c}_j) - (1 - y_i) \log p(y_i = 0 | \boldsymbol{x}_i, \boldsymbol{c}_j)$$
(13)
$$= -y_i \log \sigma(s \hat{\boldsymbol{f}}_i^T \hat{\boldsymbol{c}}_j) - (1 - y_i) \log \sigma(-s \hat{\boldsymbol{f}}_i^T \hat{\boldsymbol{c}}_j)$$

This loss maximizes the likelihood of training positive samples being accepted or negative rejected.

Triplet type loss:

$$\mathcal{L}_{Tri}^{(ii'j)} = \left[\log\sigma(s\hat{\boldsymbol{f}}_{i'}^{T}\hat{\boldsymbol{c}}_{j}) - \log\sigma(s\hat{\boldsymbol{f}}_{i}^{T}\hat{\boldsymbol{c}}_{j}) + \tau\right]_{+}$$
$$= \left[\log p(y_{i'} = 1 | \boldsymbol{x}_{i'}, \boldsymbol{c}_{j}) - \log p(y_{i} = 1 | \boldsymbol{x}_{i}, \boldsymbol{c}_{j}) + \tau\right]_{+}$$
(14)

The loss will punish when the log probability of a negative sample is greater than a positive sample with a margin τ .

Algorithm 1: mSVDD with Negative Supervision

Input: data_loader load training batch from $T' = \{(x_1, y_1), \dots, x_{n'}, y_{n'})\}$ **Models:** model includes all modules for training, $f = \phi(x_i; W)$ is a neural feature encoder **Parameters :** Hyperparameters $\gamma, \nu, s, \tau, \epsilon$ **1 for** batch in batch_loader **do**

/* load positive and negative samples $d_p, d_n = batch // size: n_p == n_n$ 2 $p = f.forward(d_p) // shape: n_p \times d$ 3 4 $n = f.forward(d_n) / / shape: n_n \times d$ Calculate \mathcal{L}_{mSVDD} given (p) (Eq. 7) 5 6 Calculate $\mathcal{L}_{Con|Tri}$ given (p,n) (Eqs. 13) or 14). /* get final loss (Eq. 16) */ $\mathcal{L}oss = \mathcal{L}_{mSVDD} + \gamma * \mathcal{L}_{Con|Tri}$ 7 Loss.backwards() 8 9 model.update()

4.2 Reformulating Contrastive and Triplet Losses for Multiple Modes

While Eqs. (12), (13), and (14) show the uni-modal case, for the multi-modal one, we have to consider m different centers $\{c_1, \ldots, c_m\}$ in the calculation of the two reformulated discriminative losses. Therefore, we propose two strategies as follows:

$$p(y_i = 1 | \boldsymbol{x}_i) = \begin{cases} \max_{1 \leq j \leq m} p(y_i = 1 | \boldsymbol{x}_i, \boldsymbol{c}_j) & Max\\ \frac{1}{m} \sum_j p(y_i = 1 | \boldsymbol{x}_i, \boldsymbol{c}_j) & Mean \end{cases}$$
(15)

where *Max* references only M_j with the max logit output, while *Mean* takes account of all hyperspheres equally. Then, we can obtain the corresponding *Contrastive* and *Triplet* losses by substituting Eqs. (13) and (14) with the probability term (Eq. (15)).

4.3 Training Loss

+The final training loss for the *mSVDD with negative* supervision can be formulated as:

$$\mathcal{L} = \mathcal{L}_{mSVDD} + \gamma \mathcal{L}_{Con|Tri}, \qquad (16)$$

where γ adjusts between the *mSVDD* loss and the discrimination with negative supervision. In the training process, $\mathcal{L}_{Con|Tri}$ will sum the loss from

one batch samples with Eqs. (13) and (14). Algorithm 1 provides the training process for mSVDD with negative supervision in one epoch. Please see Section B in the appendix for more discussions on the relationship between mSVDD and the use of negative supervision.

5 Experiments

5.1 Datasets and Implementation Details

Datasets Experiments were conducted on two datasets: 20 Newsgroups⁵ and Reuters⁶, which have been commonly used in other one-class text classification work (Manevitz and Yousef, 2001; Ruff et al., 2019). We used the same pre-processing steps as the ones used in earlier work (Ruff et al., 2019), including lowercasing, removing stopwords, and tokenization. We used the external data for negative supervision in the absence of "real" labeled negative instances. We followed the similar logic for choosing our external data as the one in the field of pretrained word vectors, in which one general corpus, such as Wikipedia articles, is often adopted as the training dataset (Mikolov et al., 2013). So we also chose one publicly available corpus WikiText-2 (Merity et al., 2016), extracted from Wikipedia articles, as our external data. As shown in Algorithm 1, data loader loads one batch of negative samples, i.e., sentences from WikiText-2, which are labeled with 0.

Encoder For encoding the text input, i.e., $\phi(x, W)$, we used a Bidirectional LSTM with attention (Hochreiter and Schmidhuber, 1997; Xu et al., 2015), with the number of hidden units being 150. For the pre-trained word embeddings, we experimented with GloVe Vectors (Pennington et al., 2014) and set the dimension to 300. In our experiments, we did not adopt the widely used BERT model (Devlin et al., 2019), as Ruff et al. (2019) showed that BERT model did not improve the performance.

Settings As for the optimization of parameters, Adam (Kingma and Ba, 2014) with a base learning rate of 0.001 was used for 50 epochs. The batch sizes were set to 32 and 64 for *Reuters* and *Newsgroups*, respectively. For the initialization of *mSVDD* model, we employed two operation steps. In the absence of negative samples, *mSVDD* was

first pre-trained on target samples by using an AutoEncoder with two objectives: 1) warm-up and 2) reducing the *reconstruction* error for the target samples, such that the model could be more robust to noise or anomalous inputs (Jacobs, 1995; Hinton and Salakhutdinov, 2006). An AutoEncoder feed-forward network with a 0.5 compression rate, which consists of an encoder and a decoder, was put on the back of the BiLSTM feature network. Then, the weights of the m hyperspheres in mSVDDwere initialized by running k-means clustering on the features learned before (Lloyd, 1982). As for the regularization term of *mSVDD*, c_i was regularized (Ng, 2004), and a weight decay with 0.95 was applied for the parameters. As for the number of hyperspheres, different settings, 1, 3, 5, 10, were tested. For the hyperparameters, we set parameter s = 1.2 for scale, $\nu = 0.1, \delta = -0.9$ for the attention weight, $\tau = 0.1$ for the triplet loss, $\epsilon = 1e-6$ for norm, and $\gamma = 1$ for the training loss. The results were averaged over 10 runs with different random seeds.

Evaluation metrics The performance was measured by the *area under the receiver operating characteristics* (ROC) curve (AUCs), a commonly used metric for one-class text classification (Manevitz and Yousef, 2001; Ruff et al., 2019).

5.2 Results

5.2.1 Results of mSVDD

Table 1 shows the performance of *mSVDD* with different choices of m, i.e., the number of hyperspheres. Here, mSVDD(1) represents unimodal deep SVDD ((Ruff et al., 2018)). The results show that: 1) As for the one-class version, mSVDD could provide better performances than the uni-modal one, especially when more hyperspheres were used. We can see that mSVDD(10), which uses the largest number of m, outperforms mSVDD(1) in more times than mSVDD(5) and mSVDD(3), that performs comparable with unimodal mSVDD. Similar results can also be observed in soft-boundary, mSVDD with more hyperspheres (10 or 5) won more times, nine out of thirteen cases in two datasets, than other settings. This proves the necessity of incorporating more hyperspheres to better describe the target data.

2) While the performance of mSVDD did not improve linearly along with m, we can explain this from the following aspects. As for the model, mSVDD with more centers means that it has more

⁵http://qwone.com/json/20Newsgroups

⁶http://daviddlewis.com/resources/testcollections/ reuters21578/

Target	mSVDD (1)		mSVD	D (3)	mSVD	D (5)	mSVDD (10)		
Class	One-v	Soft-v	One-v	Soft-v	One-v	Soft-v	One-v	Soft-v	
Reuters									
earn	95.6	95.9	95.5	95.9	96.0	96.2	95.9	<u>96.1</u>	
acq	89.4	89.0	<u>90.0</u>	89.1	89.3	89.1	90.1	89.2	
crude	<u>92.7</u>	92.8	92.5	91.5	92.5	92.5	92.4	92.4	
trade	98.4	98.3	98.3	98.9	<u>98.8</u>	98.7	98.6	<u>98.8</u>	
money	86.3	86.2	85.1	86.2	86.4	86.4	87.1	86.8	
interest	<u>97.2</u>	97.3	97.2	96.9	<u>97.2</u>	96.6	97.3	96.8	
ship	92.5	91.7	93.8	91.6	93.8	92.3	<u>92.6</u>	91.7	
20 News									
comp	85.3	84.9	86.2	86.0	86.1	85.9	86.7	86.5	
rec	77.1	76.2	77.7	77.0	77.6	76.9	77.6	76.8	
sci	66.5	66.3	67.3	67.3	67.1	66.7	66.9	67.0	
misc	75.2	75.0	76.0	76.2	75.5	76.2	75.5	76.2	
pol	79.2	79.1	78.5	78.7	78.7	78.4	78.5	78.4	
rel	83.6	82.5	<u>83.1</u>	82.5	<u>83.1</u>	82.3	<u>83.1</u>	82.2	

Table 1: mSVDD with different settings of *m*. Numbers in brackets denote the number of hyperspheres in mSVDD. *One-v* and *Soft-v* denote the two versions of mSVDD, *One-class* and *Soft-boundary*, respectively. AUCs in % on the *Reuters* (upper part) and 20 News-group (lower part) datasets. Best scores in each row are presented in **bold**, while the second best are <u>underlined</u>.

parameters and a complex model structure, which is hard to be optimized, especially on the data with a small training size (e.g., *pol* or *rel*.) As for the data, some data might have simple data distributions without the need for more modes. Another aspect would be the attention weights of multiple hyperspheres. (Ghafoori and Leckie, 2020) showed that focusing on some "*good*" hyperspheres would be beneficial rather than over all hyperspheres. In the calculation of attentions, we did not adjust δ so as to have a large weight for one specific hypersphere. This may cause limited improvements. We will compare mSVDD with DMSVDD later.

5.2.2 Results of mSVDD with negative supervision

Table 2 shows the performance of mSVDD trained with negative supervision and compares the results with the other methods. From the discussion in the last subsection, we used m = 3 in this subsection. To perform negative supervision for *mSVDD*, we evaluated four approaches where different losses and their reformulated probability forms were selected. For the method of DMSVDD, we report the results in the setting of the initial number of spheres $K_{init} = 10$. As for the comparison between DMSVDD and mSVDD, DMSVDD puts value on one hypersphere and performs slightly better over mSVDD(3) in some cases (e.g., earn, acq and comp). This indicates one inspiration that discarding "bad" hyperspheres is sometimes necessary.

For Reuters, the results indicate that mSVDD

could benefit from the joint training of the discrimination losses, except for *acq* and *ship*. *mSVDD with negative supervision* also achieved the best scores in the four cases compared with other methods including DMSVDD.

We have more obvious comparisons for 20 Newsgroup. All four negative supervision methods could improve *mSVDD* markedly and perform the best over all baselines for all target classes of 20 Newsgroup. For example, mSVDD with negative supervision could increase 2-3 points for comp. For different losses for negative supervision, the contrastive type loss, which has larger punishment over negative data, performs better than the triplet type loss, which uses a relatively small margin. Much more distinct improvements can be seen in the comparison with CVDD for rec or with OC-SVM for misc, while we obtained their best scores from Ruff et al. (2019). Further, the contrastive loss consistently outperformed other models including the baselines. In addition, the performance of Con+Max was greater than the Con+Mean strategy to reformulate the probability. We hypothesize that focusing on one of the hyperspheres is effective when we used *mSVDD* with the *contrastive* loss.

5.2.3 Results of CVDD with negative supervision

Table 3 shows the results of *CVDD* with the proposed negative supervision for mSVDD. As mentioned in Section 3.3, *CVDD* can be seen as a special case of *mSVDD*. Therefore, the proposed negative supervision approaches to *mSVDD* can be also applied to *CVDD* theoretically. To highlight the usefulness of the negative supervision, we conducted the experiments to use the *triplet* loss with *Max* probability for CVDD. As for the implementation, since CVDD uses a different multi-head structure, we also used a different form to incorporate *Triplet+Max* to CVDD (See Section C in the appendix for the details of the implementation.).

Overall, we can see that the proposed negative supervision could enhance *CVDD* in most cases on the two datasets. The overall performance mainly shows the following: 1) The improvement by the negative supervision to CVDD is consistent with mSVDD due to the similarity between the two. 2) The generality of the negative supervision can be shown, as *Triple+Max* was successfully applied to the different multi-head structure.

Regarding different target-classes, *ship* with the smaller training data size may cause worse perfor-

	Reuters target class							20 Newsgroup target class					
Model	earn	acq	crude	trade	money	interest	ship	comp	rec	sci	misc	pol	rel
OC-SVM	91.1	93.1	92.4	99.0	88.6	97.4	93.1	82.0	75.6	64.1	63.1	75.5	79.2
CVDD	94.0	<u>91.5</u>	95.5	99.2	82.8	97.7	97.6	70.9	53.3	56.8	75.1	65.3	76.3
DSVDD	95.9	89.4	92.8	98.4	86.3	97.3	92.5	85.3	77.1	66.5	75.2	79.2	83.6
DMSVDD	96.0	89.8	92.1	98.8	87.1	97.2	93.0	86.3	77.1	66.8	75.3	78.5	82.0
mSVDD_One	95.5+	90.0	92.5+	98.3+	85.1+	97.2+	<u>93.8</u>	86.2+	77.7+	67.3+	76.0+	78.5+	83.1+
+Triple+Max	96.9	89.4	93.8	99.6	89.0	98.4	92.7	88.3	78.6	67.5	77.6	79.9	83.8
+Triple+Mean	<u>97.1</u>	89.9	<u>93.9</u>	<u>99.5</u>	89.3	<u>98.3</u>	92.3	87.9	77.5	67.7	75.5	79.1	83.9
+Con+Max	97.2	90.8	92.9	98.8	91.3	97.8	92.3	89.4	79.1	68.3	76.4	<u>80.7</u>	<u>84.2</u>
+Con+Mean	96.6	91.0	92.8	98.6	90.2	98.0	91.9	<u>89.2</u>	78.9	68.3	76.6	79.9	84.4
mSVDD_Soft	95.9+	89.1	91.5+	98.9+	86.2+	96.9+	91.6	86.0+	77.0+	67.3+	76.2+	78.7+	82.5+
+Triple+Max	<u>97.1</u>	89.6	92.8	99.4	89.3	97.4	92.6	87.8	78.5	<u>68.8</u>	76.1	79.9	82.9
+Triple+Mean	97.2	90.1	92.4	99.3	91.0	98.0	92.7	87.5	78.7	68.5	76.6	79.7	83.0
+Con+Max	97.0	88.2	93.1	99.2	<u>91.2</u>	98.4	91.6	88.6	78.3	69.0	78.3	80.5	83.2
+Con+Mean	97.2	88.1	93.0	98.9	91.3	98.4	90.4	88.3	78.7	68.4	78.1	80.8	83.4

Table 2: mSVDD with negative supervision. AUCs in % on the *Reuters* (left part) and 20 *Newsgroup* (right part) datasets. For OC-SVM and CVDD, two baselines, we adopted their *best* scores from (Ruff et al., 2019). DSVDD and DMSVDD were our implementations. *One* and *Soft* mean *One-class* and *Soft-boundary* forms, respectively. +Triple+Max, which denotes mSVDD with *Triplet* loss with *Max* probability strategy, followed by three other negative supervision methods. In the rows of mSVDD_*Soft* and mSVDD_*One*, '+' following numbers means that there were improvements with negative supervision (three of four methods.) The best scores in each *column* are presented in **bold**, while the second best are <u>underlined</u>.

	Reuters target class								
Model (r) ea		rn	acq	crude	trade	money	interest	ship	
CVDD (3) 94		.0	90.2	89.6	98.3	82.5	92.3	97.6	
+Triple+Max	+Triple+Max 96		90.2	97.3	98.3	84.2	92.4	91.8	
CVDD (5)	92	.8	88.7	92.5	98.2	76.7	91.7	96.9	
+Triple+Max	94	.0	94.4	96.7	98.7	84.0	97.3	92.5	
CVDD (10)	91	.8	91.5	95.5	99.2	82.8	97.7	95.6	
+Triple+Max	93	.0	91.2	97.4	99.6	85.7	98. 7	94.2	
	20 Newsgroup target class								
Model (r)			omp	rec	sci	misc	pol	rel	
CVDD (3)		7	0.9	50.8	56.7	75.1	62.9	76.3	
+Triple+Max			4.5	64.2	61.0	75.1	62.2	72.5	
CVDD (5)			6.4	52.8	56.8	70.2	65.3	72.9	
+Triple+Max			3.2	64.5	58.4	76.2	63.6	76.1	
CVDD (10)			3.3	53.3	55.7	68.6	65.1	70.7	
+Triple+Max			8.3	69.7	60.5	73.3	67.5	79.1	

Table 3: CVDD with the proposed negative supervision. AUCs in % on the *Reuters* (upper part) and 20 *Newsgroup* (lower part) datasets. Number r in brackets denotes the number of heads in CVDD. **Bold** means the better AUCs score.

mance, so does *real* with CVDD(3), which are similar phenomena with mSVDD. In addition, the negative supervision could also prevent over-fitting for CVDD. For example, CVDD(3) with the minimal parameters achieved the best score for *comp* when varying "*r*" among 3, 5 and 10. In contrast, when the negative supervision was used, CVDD(10) with the maximal parameters attained the best and also performed better for all six target classes of the *20 Newsgroup* dataset.

6 Conclusion

In this work, we proposed mSVDD, a new generic one-class text classification framework that uses multi-modal deep SVDD. Rather than the unimodal deep SVDD, mSVDD can enhance the description ability to the target one-class data with multiple hyperspheres. We also proved that this generic framework can include three variants, deep SVDD, DMSVDD, and CVDD under certain conditions. In addition, in the absence of "real" negative training data, we also proposed approaches for effectively adding negative supervision to further improve the performance of mSVDD. The experiments validated that the proposed *mSVDD* provides better performance compared to uni-modal SVDD. The experiments also showed the further improvements in most cases when negative supervision was used for mSVDD and CVDD. For future work of this study, we will use some sampling strategies to improve the current work.

Acknowledgments

The authors would like to gratefully acknowledge the anonymous reviewers for their helpful comments and suggestions. Chenlong Hu acknowledges the support from China Scholarship Council(CSC).

References

- Hany Alashwal, Safaai Deris, and Razib M Othman. 2006. One-class support vector machines for protein-protein interactions prediction. *International Journal of Biological and Medical Sciences*, 1(2).
- Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.
- Christopher M. Bishop. 1995. Neural Networks for Pattern Recognition. Oxford University Press, Inc., New York, NY, USA.
- Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2010. Anomaly detection for discrete sequences: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 24(5):823–839.
- Pai-Hsuen Chen, Chih-Jen Lin, and Bernhard Schölkopf. 2005. A tutorial on ν-support vector machines. Applied Stochastic Models in Business and Industry, 21(2):111–136.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186.
- ST Dumais, J Platt, D Heckerman, and M Sahami. 1998. Inductive learning algorithms and representations for text categorization. 1998. In *Proceedings* of CIKM-98, 7th ACM International Conference on Information and Knowledge Management (Bethesda, MD, 1998), pages 148–155.
- Zahra Ghafoori and Christopher Leckie. 2020. Deep multi-sphere support vector data description. In *Proceedings of the 2020 SIAM International Conference on Data Mining*, pages 109–117. SIAM.
- Manish Gupta, Jing Gao, Charu C Aggarwal, and Jiawei Han. 2013. Outlier detection for temporal data: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 26(9):2250–2267.
- Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), volume 2, pages 1735–1742. IEEE.
- Pei-Yi Hao and Yen-Hsiu Lin. 2007. A new multiclass support vector machine with multi-sphere in the feature space. In *New Trends in Applied Artificial Intelligence*, pages 756–765, Berlin, Heidelberg. Springer Berlin Heidelberg.

- Geoffrey E Hinton and Ruslan R Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Robert A Jacobs. 1995. Methods for combining experts' probability assessments. *Neural computation*, 7(5):867–888.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Raksha Kumaraswamy, Anurag Wazalwar, Tushar Khot, Jude Shavlik, and Sriraam Natarajan. 2015. Anomaly detection in text: The value of domain knowledge. In *The Twenty-Eighth International Flairs Conference*.
- Trung Le, Dat Tran, and Wanli Ma. 2013. Fuzzy multisphere support vector data description. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 570–581. Springer.
- Trung Le, Dat Tran, Wanli Ma, and Dharmendra Sharma. 2010. A theoretical framework for multisphere support vector data description. In *International Conference on Neural Information Processing*, pages 132–142. Springer.
- Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. arXiv preprint arXiv:1703.03130.
- Stuart Lloyd. 1982. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137.
- Larry Manevitz and Malik Yousef. 2007. One-class document classification via neural networks. *Neurocomputing*, 70(7-9):1466–1481.
- Larry M Manevitz and Malik Yousef. 2001. One-class svms for document classification. *Journal of machine Learning research*, 2(Dec):139–154.
- Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.
- Mary M Moya, Mark W Koch, and Larry D Hostetler. 1993. One-class classifier networks for target recognition applications. *NASA STI/Recon Technical Report N*, 93.

- Andrew Y Ng. 2004. Feature selection, 1 1 vs. 1 2 regularization, and rotational invariance. In *Proceedings* of the twenty-first international conference on Machine learning, page 78.
- Mahesh Pal and Giles M Foody. 2010. Feature selection for classification of hyperspectral data by svm. *IEEE Transactions on Geoscience and Remote Sensing*, 48(5):2297–2307.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.
- Stephen J Roberts. 1999. Novelty detection using extreme value statistics. *IEE Proceedings-Vision, Im*age and Signal Processing, 146(3):124–129.
- Erik Rodner, Esther-Sabrina Wacker, Michael Kemmler, and Joachim Denzler. 2011. One-class classification for anomaly detection in wire ropes with gaussian processes in a few lines of code. *training*, 1:1– 5.
- Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. 2018. Deep one-class classification. In *International Conference on Machine Learning*, pages 4393–4402.
- Lukas Ruff, Yury Zemlyanskiy, Robert Vandermeulen, Thomas Schnake, and Marius Kloft. 2019. Selfattentive, multi-context one-class classification for unsupervised anomaly detection on text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4061–4071.
- Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. 2001. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471.
- Bernhard Schölkopf, Alex J Smola, Robert C Williamson, and Peter L Bartlett. 2000. New support vector algorithms. *Neural computation*, 12(5):1207–1245.
- Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823.
- David Martinus Johannes Tax. 2002. One-class classification: Concept learning in the absence of counterexamples.
- David MJ Tax and Robert PW Duin. 2004. Support vector data description. *Machine learning*, 54(1):45–66.

- Feng Wang, Xiang Xiang, Jian Cheng, and Alan Loddon Yuille. 2017. Normface: L2 hypersphere embedding for face verification. In *Proceedings of the* 25th ACM international conference on Multimedia, pages 1041–1049.
- Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. 2018. Unsupervised feature learning via nonparametric instance discrimination. In *Proceedings* of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3733–3742.
- Yanshan Xiao, Bo Liu, Longbing Cao, Xindong Wu, Chengqi Zhang, Zhifeng Hao, Fengzhao Yang, and Jie Cao. 2009. Multi-sphere support vector data description for outliers detection on multi-distribution data. In 2009 IEEE international conference on data mining workshops, pages 82–87. IEEE.
- Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057.
- Hwanjo Yu, Jiawei Han, and Kevin Chen-Chuan Chang. 2004. Pebl: Web page classification without negative examples. *IEEE Transactions on Knowledge & Data Engineering*, (1):70–81.

A Proofs of Proposition 1

Schölkopf et al. (2001) proved that, in single-class classification, ν is the upper bound of the fraction of anomalies, and the lower bound of the fraction of training samples being anomalies or on the optimal boundary. Ruff et al. (2018) proved that this ν -property still holds for uni-modal soft-boundary *deep SVDD*. Although the same proposition does not hold true for our multi-modal case as it is in general, it is still true when the attention weight α_{ij} is constant for different hyperspheres. This will give us an intuition on the role of ν .

Proposition 1. $(\nu$ -property $)^7$ The hyperparamter $\nu \in (0, 1]$ in soft-boundary deep mSVDD holds if we set an equal attention weight to each hypersphere:

- *i.* ν *is an upper bound on the fraction of outlier samples.*
- *ν* is a lower bound on the fraction of training samples being rejected or on the optimal boundary.

Proof. Ad (i). For each training instance x_i , its loss function is defined as *hinge-loss*: $l(f(x_i)) = \max\{0, \sum_j \alpha_{ij} (\|\phi(x_i; W) - c_j\|^2 - R_j^2)\}$, where f is the model with parameters. Let us define $d_i = \sum_j \alpha_{ij} d(x_i, c_j)$. Assume $\alpha_{ij} = 1/m$, we have $d_i = \frac{1}{m} \sum_j d(x_i, c_j)$. We also define $R_s = \frac{1}{m} \sum_j R_j^2$. And W.L.O.G, we also assume $d_1 \leq \ldots \leq d_n$ which means d_n is *n*-th farthest sum distance. The number of outliers is given by $n_{out} = |\{i|d_i > R_s\}|$. Rewrite the objective of *soft-boundary* deep mSVDD (Eq. 5) as:

$$J_{softm} = R_s - \frac{n_{out}}{\nu n} R_s = (1 - \frac{n_{out}}{\nu n}) R_s$$

Since the objective of *mSVDD* is to get a minimum R_s , therefore $1 - \frac{n_{out}}{\nu n}$ should be positive, Thus, $n_{out} \leq \nu n$ must hold in the training. It implies that at most νn outliers should be rejected.

Ad (ii). The optimal R_s^* has to hold the inequality $n_{out} \leq \nu n$. If $R_s^* >= d_n$, then n_{out} takes the minimum value of 0 which means the boundary includes all the samples. Since n_{out} is increased as long as R_s decreased. If n_{out} take the maximum value of νn under condition (i), we can have the minimal $R_s^* = d_{i*}$, where $i* = n - n_{out}$ means d_{i*} is $(n - n_{out})$ -th farthest distance. We



Figure 2: Example of **ii** of ν -property. Up arrow means $R^* = d_8$, where n = 10 means 10 samples, $\nu = 0.2$.

define $\{x_i | d_i \ge R_s^*\}$ is the set of training samples being rejected $(d_i > R_s^*)$ or on the optimal boundary $(d_i = R_s^*)$. Then we have inequality: $|\{x_i | d_i \ge R_s^*\}| = |\{x_i | d_i > R_s^*\} \cup \{x_i | d_i = R_s^*\}| \ge n_{out} + 1 \ge \nu n$. This implies that at least νn samples being rejected or just on the optimal boundary. Figure 2 shows an example with 10 training samples.

Proposition 1 and its proof refer to works (Ruff et al., 2018; Chen et al., 2005; Schölkopf et al., 2000).

B Discussions on mSVDD with Negative Supervision

B.1 Relationship between mSVDD and the use of negative supervision

mSVDD and negative supervision are not two independent sub-architectures. Negative supervision, including contrastive and triplet losses, are specially equipped to mSVDD. Specifically, these two components are closely connected by the center of the hypersphere, c_i . Both mSVDD (Eq. 7) and negative supervision (Eq. 13 or 14) contain c_i . Since there is no *real* negative data, external data are used as pseudo negative samples to complete negative supervision. The use of negative supervision could improve the discrimination ability of mSVDD. In training, negative supervision loss forces mSVDD to reject unseen samples since real negative data in testing are also unseen in training. This improves inter-class discrepancy, compared with intra-class loss mSVDD optimized. However, in testing, the decision function will be the same as mSVDD trained with only positive samples.

B.2 Necessity of joint loss

In training loss of mSVDD with *negative supervision*(Eq. 16), \mathcal{L}_{mSVDD} aims to minimize the intra-class variations while $\mathcal{L}_{Con|Tri}$ tries to maximize the inter-class discrimination. If γ in Eq. 16 is set to 0, it will train mSVDD only with target

⁷Rewrite this proposition in the main body

positive samples, where discriminative information could not be learned. On the other hand, if we use only $\mathcal{L}_{Con|Tri}$ loss for training, it may result in large intra-target variations, especially when the *triplet* type loss is chosen, since it requires that only positive samples to be closer than *pseudo* negative samples. Additionally, because of the absence of *real* negative samples, it is another problem to sample the "appropriate" *pseudo* negative samples, such that the *contrastive* or *triplet* losses could fit our original objective, that is, learn a compact description boundary for the target one-class data. Therefore, it is necessary to jointly train with the loss of negative supervision.

C Implementation of CVDD with negative supervision

The proposed negative supervision methods can also be applied to CVDD. Now, we introduce our implementation of CVDD with *triplet* type loss and the *Max* probability strategy. CVDD uses a group of *r* context vectors $C = (c_1, ..., c_r)$ to describe the target one-class data, where $c_k \in \mathbb{R}^p$. Given one context vector $c_k, \forall k \in \{1, ..., r\}$ and a pair of training positive and negative samples, we can get the reformulated probability form. First, CVDD maps a training sample x_i to *r* heads of feature vectors $S_i = (s_{i1}, ..., s_{ir})$. Then, we denote $p(y_i = 1 | s_{ik}, c_k)$ as the probability that *k*-th s_{ik} reconstructs *k*-th context vector c_k well.

$$p(y_i = 1 | \boldsymbol{s}_{ik}, \boldsymbol{c}_k) = \sigma(\hat{\boldsymbol{s}_{ik}}^T \hat{\boldsymbol{c}_k}) \qquad (17)$$

And with *triplet* and *Max* probability strategy, we can define the negative supervision loss as:

$$\mathcal{L}_{Tri}^{(ii')} = [\log p(y_{i'} = 1 | \boldsymbol{x}_{i'}) - \log p(y_i = 1 | \boldsymbol{x}_i) + \tau]_+$$

= $[\log \max_{k=1,...,r} p(y_{i'} = 1 | \boldsymbol{s}_{i'k}, \boldsymbol{c}_k) -$
 $-\log \max_{k=1,...,r} p(y_i = 1 | \boldsymbol{s}_{ik}, \boldsymbol{c}_k) + \tau]_+$
(18)

where τ is a margin. Then, $\mathcal{L}_{Tri}^{(ii')}$ can then be added to Eq. 8 to obtain the training loss.