A. OCLN

A.1. DROCC–LF Proof

Proof of Proposition 1. Recall the problem:

$$\min_{\tilde{x}} \|\tilde{x} - z\|^2, \ s.t., \ r^2 \leq \|\tilde{x} - x\|_\Sigma^2 \leq \gamma^2 r^2.$$

Note that both the constraints cannot be active at the same time, so we can consider either $r^2 \leq \|\tilde{x} - x\|_\Sigma^2$ constraint or $\|\tilde{x} - x\|_\Sigma^2 \leq \gamma^2 r^2$. Below, we give calculation when the former constraint is active, later's proof follows along same lines.

Let $\tau \leq 0$ be the Lagrangian multiplier, then the Lagrangian function of the above problem is given by:

$$L(\tilde{x}, \tau) = \|\tilde{x} - z\|^2 + \tau(\|\tilde{x} - x\|_\Sigma^2 - r^2).$$

Using KKT first-order necessary condition (Boyd & Vandenberghe, 2004), the following should hold for any optimal solution $\tilde{x}, \tau$:

$$\nabla_{\tilde{x}} L(\tilde{x}, \tau) = 0.$$

That is,

$$\tilde{x} = (I + \tau\Sigma)^{-1}(z + \tau \cdot \Sigma x) = x + (I + \tau \cdot \Sigma)^{-1}\delta,$$

where $\delta = z - x$. This proves the first part of the lemma.

Now, by using primal and dual feasibility required by the KKT conditions, we have:

$$\min_{\tau \leq 0} \|\tilde{x} - z\|^2, \ s.t., \ \|\tilde{x} - x\|_\Sigma^2 \geq r^2,$$

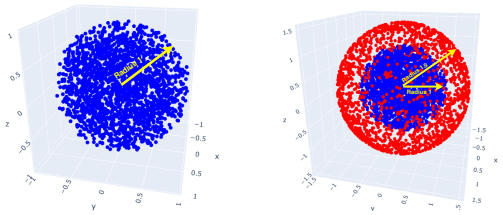
where $\tilde{x} = (I+\tau\Sigma)^{-1}(z+\tau\cdot\Sigma x) = x+(I+\tau\cdot\Sigma)^{-1}\delta$. The lemma now follows by substituting $\tilde{x}$ above and by using the fact that Σ is a diagonal matrix with $\Sigma(i,i) = \sigma_i$. □

A.2. DROCC–LF Algorithm

See Algorithm Box 2.

Algorithm 2 Training neural networks via DROCC–LF

Input: Training data $D = [(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)]$.
Parameters: Radius r, $\lambda \geq 0$, $\mu \geq 0$, step-size η, number of gradient steps m, number of initial training steps n_0.
Initial steps: For $B = 1, \ldots n_0$
  X_B: Batch of training inputs
  $\theta = \theta - \text{Gradient-Step}\Big(\sum_{\substack{(x,y)\\ \in X_B}} \ell(f_\theta(x), y)\Big)$
DROCC steps: For $B = n_0, \ldots n_0 + N$
  X_B: Batch of *normal* training inputs ($y = 1$)
  $\forall x \in X_B : h \sim \mathcal{N}(0, I_d)$
  Adversarial search: For $i = 1, \ldots m$
    1. $\ell(h) = \ell(f_\theta(x + h), -1)$
    2. $h = h + \eta \frac{\nabla_h \ell(h)}{\|\nabla_h \ell(h)\|}$
    3. $h =$ Projection given by Proposition 1($\delta = h$)
  $\ell^{itr} = \lambda\|\theta\|^2 + \sum_{\substack{(x,y)\\ \in X_B}} \ell(f_\theta(x), y) + \mu\ell(f_\theta(x + h), -1)$
  $\theta = \theta - \text{Gradient-Step}(\ell^{itr})$



(a) (b)

Figure 4. (a) Spherical manifold (a unit sphere) that captures the normal data distribution. Points are uniformly sampled from the volume of the unit sphere. (b) OOD points (red) are sampled on the *surface* of a sphere of varying radius. Table 6 shows AUC values with varying radius.

B. Synthetic Experiments

B.1. 1-D Sine Manifold

In Section 5.1.1 we presented results on a synthetic dataset of 1024 points sampled from a 1-D sine wave (See Figure 1a). We compare DROCC to other anomaly detection methods by plotting the decision boundaries on this same dataset. Figure 5 shows the decision boundary for a) DROCC b) OC-SVM with RBF kernel c) OC-SVM with 20-degree polynomial kernel d) DeepSVDD. All methods are trained only on positive points from the 1-D manifold.

We further evaluate these methods for varied sampling of negative points near the positive manifold. Negative points are sampled from a 1-D sine manifold vertically displaced in both directions (See Figure 6). Table 7 compares DROCC against various baselines on this dataset.

B.2. Spherical Manifold

OC-SVM and DeepSVDD try to find a minimum enclosing ball for the whole set of positive points, while DROCC assumes that the true points low on a low dimensional manifold. We now test these methods on a different synthetic dataset: spherical manifold where the positive points are within a sphere, as shown in Figure 4a. Normal/Positive points are sampled uniformly from the volume of the unit sphere. Table 6 compares DROCC against various baselines when the OOD points are sampled on the *surface* of a sphere of varying radius (See Figure 4b). DROCC again outperforms all the baselines even in the case when minimum enclosing ball would suit the best. Suppose instead of neural networks, we were operating with purely linear models, then DROCC also essentially finds the minimum enclosing ball (for a suitable radius r). If r is too small, the training doesn't converge since there is no separating

Table 6. Average AUC for Spherical manifold experiment (Section B.2). Normal points are sampled uniformly from the volume of a unit sphere and OOD points are sampled from the *surface* of a unit sphere of varying radius (See Figure 4b). Again DROCC outperforms all the baselines when the OOD points are quite close to the normal distribution.

Radius	Nearest Neighbor	OC-SVM	AutoEncoder	DeepSVDD	DROCC (Ours)
1.2	100 ± 0.00	92.00 ± 0.00	91.81 ± 2.12	93.26 ± 0.91	99.44 ± 0.10
1.4	100 ± 0.00	92.97 ± 0.00	97.85 ± 1.41	98.81 ± 0.34	99.99 ± 0.00
1.6	100 ± 0.00	92.97 ± 0.00	99.92 ± 0.11	99.99 ± 0.00	100.00 ± 0.00
1.8	100 ± 0.00	91.87 ± 0.00	99.98 ± 0.04	100.00 ± 0.00	100.00 ± 0.00
2.0	100 ± 0.00	91.83 ± 0.00	100 ± 0.00	100.00 ± 0.00	100.00 ± 0.00

Table 7. Average AUC for the synthetic 1-D Sine Wave manifold experiment (Section B.1). Normal points are sampled from a sine wave and OOD points from a vertically displaced manifold (See Figure 6). The results demonstrate that only DROCC is able to capture the manifold tightly

Vertical Displacement	Nearest Neighbor	OC-SVM	AutoEncoder	DeepSVDD	DROCC (Ours)
0.2	100 ± 0.00	56.99 ± 0.00	52.48 ± 1.15	65.91 ± 0.64	96.80 ± 0.65
0.4	100 ± 0.00	68.84 ± 0.00	58.59 ± 0.61	78.18 ± 1.67	99.31 ± 0.80
0.6	100 ± 0.00	76.95 ± 0.00	66.59 ± 1.21	82.85 ± 1.96	99.92 ± 0.11
0.8	100 ± 0.00	81.73 ± 0.00	77.42 ± 3.62	86.26 ± 1.69	99.98 ± 0.01
1.0	100 ± 0.00	88.18 ± 0.00	86.14 ± 2.52	90.51 ± 2.62	100 ± 0.00
2.0	100 ± 0.00	98.56 ± 0.00	100 ± 0.00	100 ± 0.00	100 ± 0.00

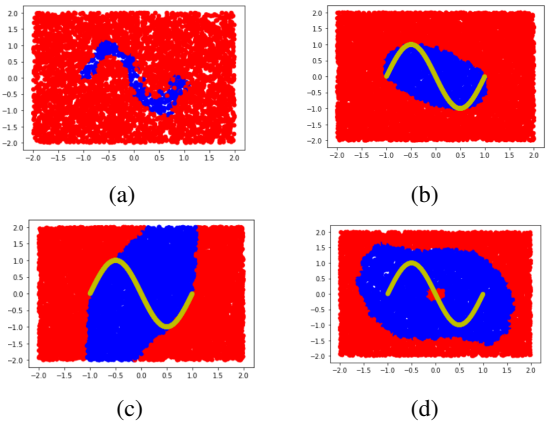


Figure 5. (a) Decision boundary of DROCC trained only on the positive points lying on the 1-D sine manifold in Figure 1a. Blue represents points classified as normal and red classified as abnormal. (b) Decision boundary of classical OC-SVM using RBF kernel and same experiment settings as in (a). Yellow sine wave just shows the underlying train data. (c) Decision boundary of classical OC-SVM using a 20-degree polynomial kernel. (d) Decision boundary of DeepSVDD.

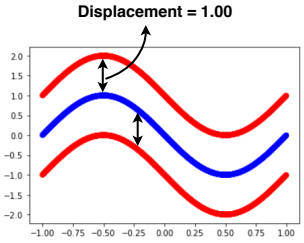


Figure 6. Illustration of the negative points sampled at various displacements of the sine wave; used for reporting the AUC values in the Table 7. In this figure, vertical displacement is 1.0. Blue represents the positive points (also the training data) and red represents the negative/OOD points

Table 8. Ablation Study on CIFAR-10: Sampling negative points randomly in the set $N_i(r)$ (DROCC–Rand) instead of gradient ascent (DROCC).

CIFAR Class	One-Class Deep SVDD	DROCC	DROCC–Rand
Airplane	61.7±4.1	81.66 ± 0.22	79.67 ± 2.09
Automobile	65.9±2.1	76.74 ± 0.99	73.48 ± 1.44
Bird	50.8±0.8	66.66 ± 0.96	62.76 ± 1.59
Cat	59.1±1.4	67.13 ± 1.51	67.33 ± 0.72
Deer	60.9±1.1	73.62 ± 2.00	56.09 ± 1.19
Dog	65.7±2.5	74.43 ± 1.95	65.88 ± 0.64
Frog	67.7±2.6	74.43 ± 0.92	74.82 ± 1.77
Horse	67.3±0.9	71.39 ± 0.22	62.08 ± 2.03
Ship	75.9±1.2	80.01 ± 1.69	80.04 ± 1.71
Truck	73.1±1.2	76.21 ± 0.67	70.80 ± 2.73

boundary). Assuming neural networks are implicitly regularized to find the simplest boundary, DROCC with neural networks also learns essentially a minimum enclosing ball in this case, however, at a slightly larger radius. Therefore, we get 100% AUC only at radius 1.6 rather than $1 + \epsilon$ for some very small ϵ.

C. LFOC Supplementary Experiments

In Section 5.2.1, we compared DROCC–LF with various baselines for the OCLN task where the goal is to learn a classifier that is accurate for both the positive class and the arbitrary OOD negatives. Figure 9 compares the recall obtained by different methods on 2 keywords "Forward" and "Follow" with 2 different FPR. Table 9 lists the close negatives which were synthesized for each of the keywords.

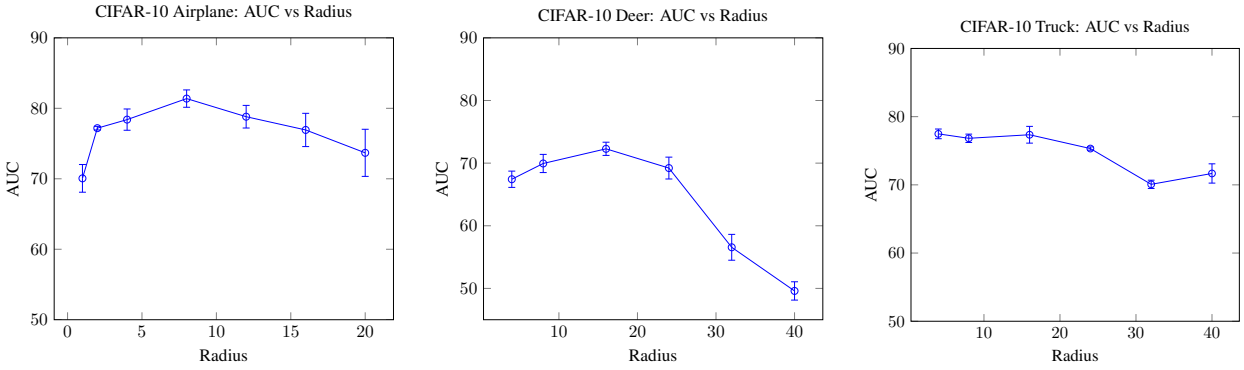


Figure 7. Ablation Study : Variation in the performance DROCC when r (with $\gamma = 1$) is changed from the optimal value.

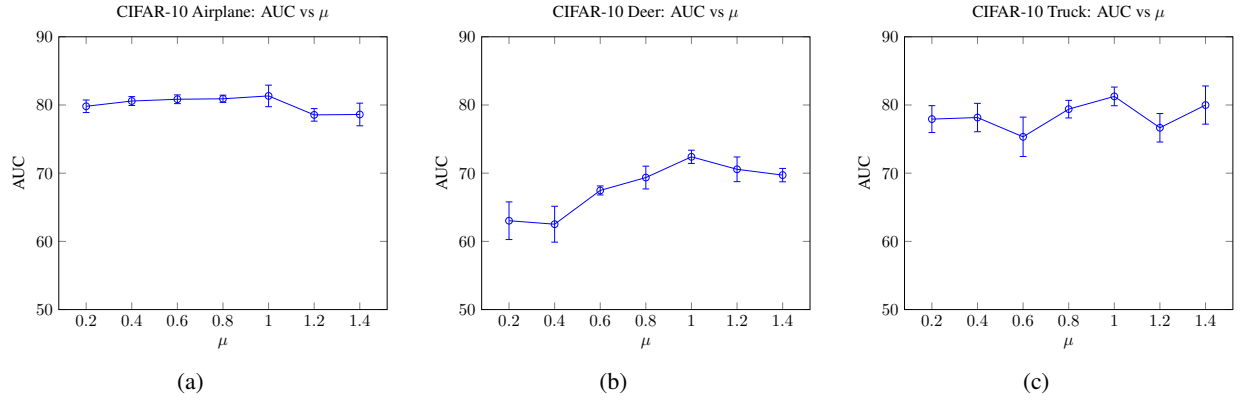


Figure 8. Ablation Study : Variation in the performance of DROCC with μ (1) which is the weightage given to the loss from adversarially sampled negative points

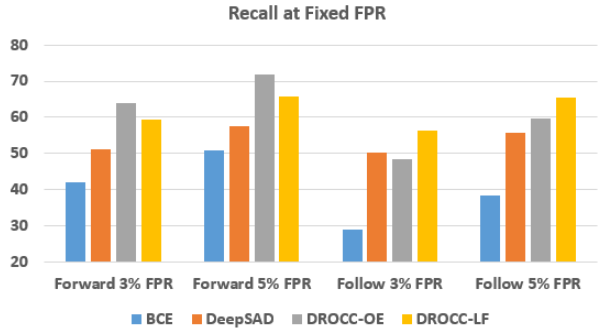


Figure 9. OCLN on Audio Commands: Comparison of Recall for key words — "Forward" and "Follow" when the False Positive Rate(FPR) is fixed to be 3% and 5%.

Table 9. Synthesized near-negatives for keywords in Audio Commands

Marvin	Forward	Seven	Follow
mar	for	one	fall
marlin	fervor	eleven	fellow
arvin	ward	heaven	low
marvik	reward	when	hollow
arvi	onward	devon	wallow

Table 10. Hyperparameters: Tabular Experiments

Dataset	Radius	μ	Optimizer	Learning Rate	Adversarial Ascent Step Size
Abalone	3	1.0	Adam	10^{-3}	0.01
Arrhythmia	16	1.0	Adam	10^{-4}	0.01
Thyroid	2.5	1.0	Adam	10^{-3}	0.01

D. Ablation Study

D.1. Hyper-Parameters

Here we analyze the effect of two important hyper-parameters — radius r of the ball outside, which we sample negative points (set $N_i(r)$), and μ which is the weightage given to the loss from adversarially generated negative points (See Equation 1). We set $\gamma = 1$ and hence recall that the negative points are sampled to be at a distance of r from the positive points.

Figure 7a, 7b and 7c show the performance of DROCC with

Table 11. Hyperparameters: CIFAR-10

Class	Radius	μ	Optimizer	Learning Rate	Adversarial Ascent Step Size
Airplane	8	1	Adam	0.001	0.001
Automobile	8	0.5	SGD	0.001	0.001
Bird	40	0.5	Adam	0.001	0.001
Cat	28	1	SGD	0.001	0.001
Deer	32	1	SGD	0.001	0.001
Dog	24	0.5	SGD	0.01	0.001
Frog	36	1	SGD	0.001	0.01
Horse	32	0.5	SGD	0.001	0.001
Ship	28	0.5	SGD	0.001	0.001
Truck	16	0.5	SGD	0.001	0.001

Table 12. Hyperparameters: ImageNet

Class	Radius	μ	Optimizer	Learning Rate	Adversarial Ascent Step Size
Tench	30	1	SGD	0.01	0.001
English_springer	16	1	SGD	0.001	0.001
Cassette_player	40	1	Adam	0.005	0.001
Chain_saw	20	1	SGD	0.01	0.001
Church	40	1	Adam	0.01	0.001
French_horn	20	1	SGD	0.05	0.001
Garbage_truck	30	1	Adam	0.005	0.001
Gas_pump	30	1	Adam	0.01	0.001
Golf_ball	30	1	SGD	0.01	0.001
Parachute	12	1	Adam	0.001	0.001

varied values of r on the CIFAR-10 dataset. The graphs demonstrate that sampling negative points quite far from the manifold (setting r to be very large), causes a drop in the accuracy since now DROCC would be covering the normal data manifold loosely causing high false positives. At the other extreme, if the radius is set too small, the decision boundary could be too close to the positive and hence lead to overfitting and difficulty in training the neural network. Hence, setting an appropriate radius value is very critical for the good performance of DROCC.

Figure 8a, 8b and 8c show the effect of μ on the performance of DROCC on CIFAR-10.

D.2. Importance of gradient ascent-descent technique

In the Section 3 we formulated the DROCC's optimization objective as a saddle point problem (Equation 1). We adopted the standard gradient descent-ascent technique to solve the problem replacing the ℓ_p ball with $N_i(r)$. Here, we present an analysis of DROCC without the gradient ascent part i.e., we now sample points at random in the set of negatives $N_i(r)$. We call this formulation as DROCC–Rand. Table 8 shows the drop in performance when negative points are sampled randomly on the CIFAR-10, hence emphasizing the importance of gradient ascent-descent technique. Since $N_i(r)$ is high dimensional, random sampling does not find points close enough to manifold of positive points.

Table 13. Hyperparameters: Timeseries Experiments

Dataset	Radius	μ	Optimizer	Learning Rate	Adversarial Ascent Step Size
Epilepsy	10	0.5	Adam	10^{-5}	0.1
Audio Commands	16	1.0	Adam	10^{-3}	0.1

Table 14. Hyperparameters: LFOC Experiments

Keyword	Radius	μ	Optimizer	Learning Rate	Adversarial Ascent Step Size
Marvin	32	1	Adam	0.001	0.01
Seven	36	1	Adam	0.001	0.01
Forward	40	1	Adam	0.001	0.01
Follow	20	1	Adam	0.0001	0.01

E. Experiment details and Hyper-Parameters for Reproducibility

E.1. Tabular Datasets

Following previous work, we use a base network consisting of a single fully-connected layer with 128 units for the deep learning baselines. For the classical algorithms, the features are input to the model. Table 10 lists all the hyper-parameters for reproducibility.

E.2. CIFAR-10

DeepSVDD uses the representations learnt in the penultimate layer of LeNet (LeCun et al., 1998) for minimizing their one-class objective. To make a fair comparison, we use the same base architecture. However, since DROCC formulates the problem as a binary classification task, we add a final fully connected layer over the learned representations to get the binary classification scores. Table 11 lists the hyper-parameters which were used to run the experiments on the standard test split of CIFAR-10.

E.3. ImageNet-10

MobileNetv2 (Sandler et al., 2018a) was used as the base architecture for DeepSVDD and DROCC. Again we use the representations from the penultimate layer of MobileNetv2 for optimizing the one-class objective of DeepSVDD. The width multiplier for MobileNetv2 was set to be 1.0. Table 12 lists all the hyper-parameters.

E.4. Time Series Datasets

To keep the focus only on comparing DROCC against the baseline formulations for OOD detection, we use a single layer LSTM for all the experiments on Epileptic Seizure Detection, and the Audio Commands dataset. The hidden state from the last time step is used for optimizing the one class objective of DeepSVDD. For DROCC we add a fully connected layer over the last hidden state to get the binary

classification scores. Table 13 lists all the hyper-parameters for reproducibility.

E.5. LFOC Experiments on Audio Commands

For the Low-FPR classification task, we use keywords from the Audio Commands dataset along with some synthesized near-negatives. The training set consists of 1000 examples of the keyword and 2000 randomly sampled examples from the remaining classes in the dataset. The validation and test set consist of 600 examples of the keyword, the same number of words from other classes of Audio Commands dataset and an extra synthesized 600 examples of close negatives of the keyword (see Table 9) A single layer LSTM, along with a fully connected layer on top on the hidden state at last time step was used. Similar to experiments with DeepSVDD, DeepSAD uses the hidden state of the final timestep as the representation in the one-class objective. An important aspect of training DeepSAD is the pretraining of the network as the encoder in an autoencoder. We also tuned this pretraining to ensure the best results.