WHAT IS WRONG WITH ONE-CLASS ANOMALY DETECTION?

Junekyu Park, Jeong-Hyeon Moon, Namhyuk Ahn & Kyung-Ah Sohn
Department of Artificial Intelligence, Ajou University
idbluefish@gmail.com, {mjh319,aa0dfg,kasohn}@ajou.ac.kr

ABSTRACT

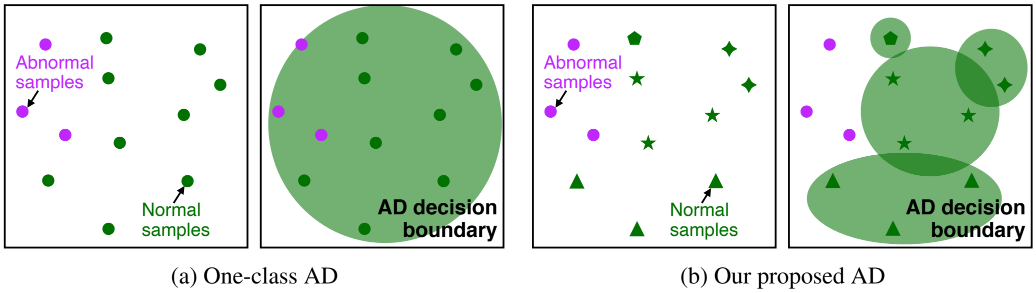
From a safety perspective, a machine learning method embedded in real-world applications is required to distinguish irregular situations. For this reason, there has been a growing interest in the anomaly detection (AD) task. Since we cannot observe abnormal samples for most of the cases, recent AD methods attempt to formulate it as a task of classifying whether the sample is normal or not. However, they potentially fail when the given normal samples are inherited from diverse semantic labels. To tackle this problem, we introduce a latent class-condition-based AD scenario. In addition, we propose a confidence-based self-labeling AD framework tailored to our proposed scenario. Since our method leverages the hidden class information, it successfully avoids generating the undesirable loose decision region that one-class methods suffer. Our proposed framework outperforms the recent one-class AD methods in the latent multi-class scenarios.

1 INTRODUCTION

With the rapid increase in the performance of deep learning-based methods, the demands for applying this technology are emerging in recent. However, simply adopting it may not be ideal due to the mismatched label information between the training and test set. A self-driving system, for example, should make the best decision on the abnormal scene or status even though it never observed this condition before. Under this scenario, it is required to detect whether the given data is unseen (abnormal) or not, to build a reliable and secure machine learning system.

The anomaly detection (AD) task focuses on to identify suspicions or abnormal events. We can categorize this task into supervised (Liang et al., 2017) or unsupervised (Chalapathy et al., 2017; Oza & Patel, 2018) by the training strategy. The former train the model with both normal and abnormal samples, while the latter use normal data only. Since collecting the abnormal cases is time-consuming or impossible in the real-world, unsupervised learning gets more attention despite the superior performance of the supervised scheme. In an unsupervised approach, most of the methods formulate the task as an one-class classification problem (*i.e.* classify as normal or not). While such simplicity works on the well-refined scenarios, they potentially fail when the given normal dataset is composed of samples from diverse classes (Figure 1a). Since conventional methods ignore the latent class information, they tend to draw a single binary decision boundary that loosely covering a wide range. This could be problematic when the normal dataset contains multiple latent class information. Can we leverage this semantic knowledge to guide the AD methods to judge the anomalies more accurately? We believe a different approach is needed.

Base on this assumption, we introduce a latent class-condition-based AD scenario and its benchmark datasets. This simulates the circumstance where both normal and abnormal data have multi-class samples (Figure 1b). We would like to emphasize that our proposed scenario is realistic; in the real-world, various (object) classes could be normal, and the abnormal cases could consist of diverse classes as well (*e.g.* street signs of different countries). Because of the nature of our scenario, the optimal solution needs to generate semantic-aware *tight* decision boundaries (as in Figure 1b). However, neither supervised nor unsupervised approaches can handle the scenario properly. For example, supervised models cannot be applied since no label is given. Conventional unsupervised AD methods create a single and loose decision region (since they ignore latent class information). At this point, one natural question can arise: How can we use latent semantic information? Since the



(a) One-class AD

(b) Our proposed AD

Figure 1: Comparison of two AD scenarios. **(a)** One-class AD scenario where the model creates a single decision boundary to cover all normal data. Some abnormal samples are misclassified due to the *loose* boundary. **(b)** Our proposed AD scenario. An optimal solution can model latent class-condition information and draws *tight* class-wise decision boundaries. For normal samples, each class is denoted as different shapes. Note that the latent class information are not observable to the models; only the oracle is able to access this.

latent label is accessible to the oracle (or human) only, we may detour this limitation, for example, by utilizing the pseudo-latent class label.

To tackle this, we propose a Confidence-based self-Labeling Anomaly Detection (CLAD) framework to bridge the gap between the supervised and unsupervised approaches. We model the latent class information using self-labeling so that supervised learning can be adapted (Figure 2). To do this, we first train the feature extraction network following Xie et al. (2016). Then, we cluster the training samples using the extracted latent feature and allocate pseudo labels via self-labeling (Lee et al., 2013). Now the classification network can learn to predict pseudo labels by standard supervised learning. At the inference, we decide whether the test sample is abnormal or not by the confidence-based AD method (Liang et al., 2017). Since our method leverages the hidden class information, it successfully avoids the generation of undesirable loose decision regions typically suffered by one-class methods. Several experiments on latent multi-class scenarios demonstrate that the proposed method substantially outperforms recent one-class AD methods.[1]

2 Latent Class-condition Anomaly Detection Scenario

Traditional one-class AD methods learn a decision boundary based on the given normal samples in the training dataset (Figure 1a). Such one-class strategy is effective when a single class label is treated as normal alone (and the rest of them are abnormal). However, this assumption may not hold in a real environment since the normal category can consist of heterogeneous semantic labels. With this circumstance, conventional one-class AD generates a loose decision boundary to cover all samples drawn from diverse classes and thus vulnerable to a false negative.

To reduce the gap between the real-world and the one-class AD scenarios, we simulate the scenario environment where the latent sub-classes exist implicitly (Figure 1b). With this environment, it is crucial to learn a decision boundary by seeing not only the normality of the data samples but also its semantics. Note that such class information is not observable, thus the AD framework may require learning the semantic representation in an unsupervised or self-supervised manner.

3 Confidence-based Self-labeling Anomaly Detection

Before going into the details, let us define the problem statement. We have a training set $X_{tr} = \{\mathbf{x}_i\}_{i=1}^N$ and a test set (X_{te}, Y_{te}). X_{te} contains both normal/abnormal samples with corresponding label Y_{te}; 0 if normal and 1 otherwise. The core idea of our framework is to consider the concealed semantic information. To do that, we generate the pseudo-label $y^* \in \mathcal{Y}^* = \{1, ..., L\}$, where the cardinality of $\mathcal{Y}^*$ is assumed pseudo-class counts. Then, our AD framework inferences a true $y \in Y_{te}$ based on the classifier F that is trained with generated pairs (X_{tr}, Y_{tr}^*), where $Y_{tr}^* = \{\mathbf{y}_i^*\}_{i=1}^N$.

[1] Our code is available at `https://github.com/JuneKyu/CLAD`

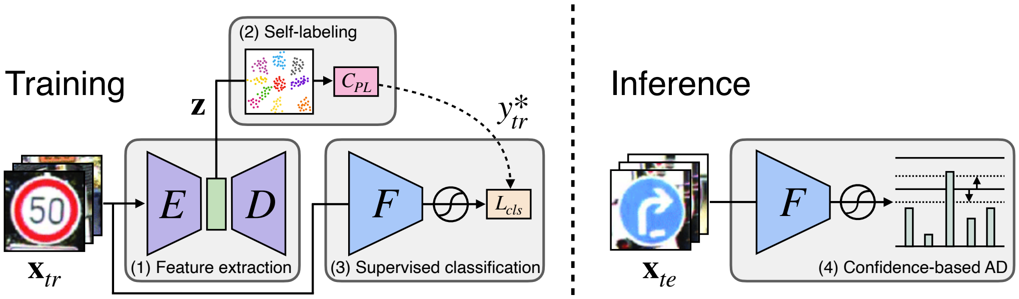


Figure 2: Overview of our proposed confidence-based self-labeling anomaly detection framework. **(1)** Feature extraction by training the convolutional autoencoder to extract the latent feature $\mathbf{z}$. **(2)** Self-labeling via clustering to assign latent class y^* to sample $\mathbf{x}$. **(3)** Train a classifier F using (X_{tr}, Y^*_{tr}) pairs in supervised learning manner. L_{cls} denotes a classification loss. **(4)** We perform anomaly detection by the confidence score of classifying normal sample $\mathbf{x}$ to y^*.

(1) Feature extraction. First, we train the autoencoder (E and D) with reconstruction loss on training dataset X_{tr} and calculate the latent feature $\mathbf{z}$ using the trained encoder E.

(2) Self-labeling via clustering. We further fine-tune the encoder E to increase the possibility of assigning a sample $\mathbf{x}$ to the given cluster C_i by the Kullback-Leibler divergence between the latent feature $\mathbf{z}$ and soft-alignment of C_i (Xie et al., 2016). We simultaneously update the encoder and cluster assignment and this makes the clustering algorithm to be robust. Then, we assign a label y^* by referring to the allocated cluster C_i. Now the X_{tr} have associated pseudo-labels Y^*_{tr}.

(3) Supervised classification. With (pseudo) latent class labels Y^*_{tr} and the normal sample X_{tr}, a classifier F is trained with a standard supervised learning scheme. By referring to the class information that is intrinsic in the latent labels, our framework can generate the semantic-aware *tight* decision boundaries that envelop each relevant class sample only.

(4) Confidence-based AD. Because of the tight decision boundary by the pairs (X_{tr}, Y^*_{tr}), we argue that the AD can be viewed as the out-of-distribution (OOD) task. In detail, the OOD sample is defined as the data where the class label is not included in the training dataset. We assume the label set $\mathcal{Y}^*$ is a multi-classes label set from Y^*_{tr}. Then, we treat $(\mathbf{x}, y^*)$ as an OOD sample when the $y^* \notin \mathcal{Y}^*_{tr}$. Because the confidence-based algorithms predict the samples as OOD when the classifier outputs a small probability for all classes, we can safely convert anomaly detection into an OOD task. (*i.e.*, it is an abnormal sample when $y^* \notin \mathcal{Y}^*_{tr}$).

Under these assumptions, we measure anomaly detection scores by adapting the scoring scheme in ODIN (Liang et al., 2017). We define the score $s(\mathbf{x}; T, \delta) = max_i p(\tilde{\mathbf{x}}; T)_{(i)}$, where $p(\tilde{\mathbf{x}}; T)_{(i)}$ is output of the classifier F in each class i. T is the parameter for the temperature scaling and $\tilde{\mathbf{x}}$ is the input term $\tilde{\mathbf{x}} = \mathbf{x} - \epsilon(\mathbf{x})$ perturbed by the reverse FGSM method (Goodfellow et al., 2014) that makes the OOD samples more separable. If the score $s(x; T, \delta)$ is greater than a given threshold δ, then we predict the $\hat{y}$ as normal.

4 Experiment

Baselines. We compare with one-class AD methods: OCSVM (Schölkopf et al., 2001), OCNN (Chalapathy et al., 2017), OCCNN (Oza & Patel, 2018), SVDD (Tax & Duin, 2004), and DeepSVDD (Ruff et al., 2018). We follow the implementation setups based on the official codes.

Datasets. We use following datasets: MNIST (LeCun & Cortes, 2010), CIFAR-10 (Krizhevsky et al., 2009), GTSRB (Stallkamp et al., 2012), and Tiny-ImageNet (Russakovsky et al., 2015).

We devise the super-categories by merging the semantic labels to simulate our AD scenario. For example, MNIST is as {`Curly, Straight, Mix`} and GTSRB based on the semantic meanings of traffic signs. Note that both datasets share a similar domain prior which represents the text or symbols. For the natural scene datasets such as CIFAR-10 and Tiny-ImageNet, we set as {`Thing, Living`} and {`Animal, Insect, Instrument, Structure, Vehicle`},

Table 1: Performance comparison on the newly proposed AD scenario with MNIST and GTSRB.

Method	**MNIST**			**GTSRB**					
	CUR	STR	MIX	SPDL	INST	WARN	DIRC	SPEC	REGN
OCSVM	69.9	87.4	54.8	66.3	60.9	52.7	60.7	50.5	77.2
OCNN	81.4	76.2	60.2	65.7	65.7	51.8	62.0	52.7	78.3
OCCNN	77.5	78.3	51.5	59.3	57.5	60.4	56.4	59.1	64.3
SVDD	58.8	72.4	52.6	58.2	55.1	51.1	50.5	56.9	67.4
DeepSVDD	81.7	82.2	69.7	57.7	**69.7**	**73.4**	59.3	**70.2**	77.8
CLAD (ours)	**94.0**	**96.1**	**92.6**	**66.7**	66.5	64.9	**67.4**	62.2	**79.1**

Table 2: Performance comparison on our AD scenario with CIFAR-10 and Tiny-ImageNet.

Method	**CIFAR-10**		**Tiny-ImageNet**				
	THG	LIV	ANML	ISCT	ISTM	STRT	VHCL
OCSVM	51.9	67.7	63.5	60.9	50.2	56.7	55.5
OCNN	58.5	67.2	58.2	57.1	50.3	51.0	55.1
OCCNN	59.8	62.5	59.7	62.4	51.5	54.2	**66.4**
SVDD	50.8	61.4	51.5	51.8	51.5	50.6	51.9
DeepSVDD	65.0	52.7	59.0	53.5	53.4	55.4	53.5
CLAD (ours)	**74.9**	**72.8**	**65.9**	**66.2**	**55.6**	**62.0**	64.7

respectively. When we evaluate the models, we pick one super-category for training and the rest of the subsets as a test dataset. Please see Appendix D for the detailed settings of the scenario.

Results. Table 1 shows the AUROC of the MNIST and GTSRB datasets. Our framework surpasses all the one-class AD methods in MNIST. Among them, it is notable that CLAD outperforms others on Mixed in a huge margin. This scenario is very challenging since it carries complex information due to the mixed shape of digits such as '2' or '6'. For the GTSRB dataset, our CLAD reaches the best performance for SPDL, DIRC, REGN and second-best for the rest. The only method comparable to ours is DeepSVDD. However, it suffers the inconsistent performance (worst score in SPEC) while our framework shows stable and high performances for all the cases.

To demonstrate the superior performance of CLAD in the complex image domain, we evaluate it on the CIFAR-10 and Tiny-ImageNet datasets (Table 2). Our framework achieves the best performance for all the scenarios in CIFAR-10 and four of five cases in Tiny-ImageNet. Similar to GTSRB, the scores of DeepSVDD are inconsistent; we claim that DeepSVDD is sensitive to the latent class labels. It maps all the (normal) samples into a single-modal hypersphere (Figure 1a), making it vulnerable to the anomalies that close to the normals in terms of the class information.

5 Discussion & Conclusion

We introduced a new aspect of the AD task and proposed a confidence-based AD framework. We assume that (ab)normal categories can have (unobservable) multi-class samples in contrast to the one-class AD scenarios. We believe that our scenario and method are practical in real world. One possible usage is for the industrial environment. When the *normal* sensor signals with various range can be altered by the surroundings, conventional one-class AD methods suffer spurious detection.

Since our framework trains a classifier using self-labeling, extracting the right representation of the data sample is crucial. However, most of the feature extraction methods have difficulty with handling a dataset bias (Bahng et al., 2020). As future work, we will focus on the disentanglement of the scene context with a key concept of the object discovery (Burgess et al., 2019).

ACKNOWLEDGEMENTS

This research was supported by the National Research Foundation of Korea grant funded by the Korea government (MSIT) (No. NRF-2019R1A2C1006608), and also under the ITRC (Information Technology Research Center) support program (IITP-2020-2018-0-01431) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation).

REFERENCES

Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. *arXiv preprint arXiv:1911.05371*, 2019.

Hyojin Bahng, Sanghyuk Chun, Sangdoo Yun, Jaegul Choo, and Seong Joon Oh. Learning de-biased representations with biased representations. In *International Conference on Machine Learning*, pp. 528–539. PMLR, 2020.

Christopher P Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation. *arXiv preprint arXiv:1901.11390*, 2019.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 132–149, 2018.

Raghavendra Chalapathy, Aditya Krishna Menon, and Sanjay Chawla. Robust, deep and inductive anomaly detection. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 36–51. Springer, 2017.

Carl Doersch and Andrew Zisserman. Multi-task self-supervised visual learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2051–2060, 2017.

Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pp. 1422–1430, 2015.

Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(9):1734–1747, 2015.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

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.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. URL `http://yann.lecun.com/exdb/mnist/`.

Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, 2013.

Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2017.

George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11): 39–41, 1995.

Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pp. 69–84. Springer, 2016.

Mehdi Noroozi, Hamed Pirsiavash, and Paolo Favaro. Representation learning by learning to count. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5898–5906, 2017.

Poojan Oza and Vishal M Patel. One-class convolutional neural network. *IEEE Signal Processing Letters*, 26(2):277–281, 2018.

Aristotelis-Angelos Papadopoulos, Mohammad Reza Rajati, Nazim Shaikh, and Jiamian Wang. Outlier exposure with confidence control for out-of-distribution detection. *arXiv preprint arXiv:1906.03509*, 2019.

Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *International conference on machine learning*, pp. 4393–4402. PMLR, 2018.

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.

Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, (0):–, 2012. ISSN 0893-6080. doi: 10.1016/j.neunet.2012.02.016.

David MJ Tax and Robert PW Duin. Support vector data description. *Machine learning*, 54(1): 45–66, 2004.

Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pp. 478–487. PMLR, 2016.

A Related Work

One-class anomaly detection. Most of the anomaly detection (AD) methods treat a task as a binary classification: normal and abnormal. Under this assumption, OCSVM (Schölkopf et al., 2001) learns a decision boundary with normal samples only. OC-NN and OC-CNN (Chalapathy et al., 2017; Oza & Patel, 2018) are the earlier attempts on using a deep learning-based approach with an unsupervised learning regime. However, unlike OCSVM, these works have the potential risk to be a trivial solution due to the insufficient theoretical analysis. That is, the learnable parameters may become a trivial solution when the given data samples are all normal. On the other hand, SVDD (Tax & Duin, 2004) and DeepSVDD (Ruff et al., 2018) successfully avoid such trivialness by the theoretical basis. Although the aforementioned methods have shown promising results, they are limited to the one-class scenario which is not suitable for real-world applications. In contrast, our proposed task assumes that the data points could be allocated into the various classes, which is more realistic.

Self-labeling. The self-labeling is one of the most rapidly developing approaches in the self-supervised learning context (Dosovitskiy et al., 2015; Doersch et al., 2015; Noroozi & Favaro, 2016; Noroozi et al., 2017; Doersch & Zisserman, 2017; Gidaris et al., 2018). Base on the success of self-supervised learning, recent works attempt to create a self-label using traditional clustering methods. For example, DeepCluster (Caron et al., 2018) first makes initial self-labels using a convolutional network and iteratively updates the network parameters with the re-assigning process by the clustering algorithms. Asano et al. (2019) extended DeepCluster making it to learn visual representation and clustering simultaneously based on the information theory as such maximizing the information between the labels and the input data. Although self-labeling has shown outstanding performance in the computer vision field, to the best of our knowledge, none of the studies exists to apply this strategy to the anomaly detection field.

Out-of-distribution detection. Out-of-distribution (OOD) detection is the task to discriminate the samples whether they are from the training distribution or not. Because the deep learning-based classification model tends to predict as a wrong class with high-confidence when the given test samples are the class not in the training set (*i.e.* high-confidence problem). To tackle this issue, many works have been studied on this problem. ODIN (Liang et al., 2017) addressed the high-confidence problem using temperature scaling and an input-preprocessing method from the FGSM (Goodfellow et al., 2014). Their method pursues better separating the out-of- and in-distribution samples. Papadopoulos et al. (2019) proposed an additional loss function from ODIN and applied other machine learning tasks such as natural language problems. The aforementioned methods rely on the multi-class labels when training, *i.e.* supervised approach. This limits the OOD detection methods difficult to apply to real-world anomaly detection problems.

B Model Analysis

In this section, we analyze our proposed method. First, we evaluate CLAD on the previous one-class AD scenario. Second, we show how our method is robust to the hyper-parameter settings such as the number of the clusters or the hidden dimension size.

One-class AD. We can view this task as a simplification of our proposed scenario. In one-class AD, the normal category has a single semantic label only, in contrast to ours which sets the normal condition to contain multiple class information. We evaluate the methods on MNIST and CIFAR-10 datasets as shown in Table 3 and 4. Our CLAD shows comparable results on MNIST to the other one-class-based AD methods and competes on par with DeepSVDD on the CIFAR-10 dataset.

We would like to note that CLAD is not designed for the one-class AD task. Because of the feature clustering and label assignment, our method could create fragmented decision boundaries in this conventional scenario. In contrast, DeepSVDD learns to generate a spherical decision boundary that tightly wraps the single-class normal samples. However, on CIFAR-10 which may have various latent semantics within the single class (*e.g.* different pose, intra-class diversity), DeepSVDD fails to detect abnormal in some cases (*e.g.* bird, deer) while CLAD shows consistent scores.

The effects of the hyper-parameters. In our framework, we use feature extraction and clustering modules. Since these modules are the basis of the self-labeling procedure, we analyze how the hyper-

Table 3: One-class AD performance on MNIST dataset.

Method	MNIST									
	0	1	2	3	4	5	6	7	8	9
OCSVM	98.3	99.5	82.0	88.5	91.6	79.7	93.1	93.4	83.9	91.3
OCNN	97.6	99.5	87.3	86.5	93.3	86.5	97.1	93.6	88.5	93.5
OCCNN	91.8	98.7	74.9	78.2	88.3	72.7	75.6	85.6	69.1	78.5
SVDD	**98.6**	99.5	82.5	88.1	**94.9**	77.1	96.5	93.7	88.9	93.1
DeepSVDD	98.0	**99.7**	**91.7**	**91.9**	**94.9**	88.5	**98.3**	**94.6**	**93.9**	**96.5**
CLAD (ours)	96.3	97.9	89.8	87.2	92.2	**90.7**	92.5	91.5	80.0	92.0

Table 4: One-class AD performance on CIFAR-10 dataset.

Method	CIFAR-10									
	airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck
OCSVM	65.1	59.0	**65.2**	50.1	**75.1**	51.3	71.7	51.2	67.6	51.0
OCNN	60.4	62.0	63.7	53.6	67.4	56.1	63.3	60.1	64.7	60.3
OCCNN	65.0	65.6	62.8	51.2	72.7	50.9	64.7	52.2	66.5	66.9
SVDD	61.6	63.8	50.0	55.9	66.0	62.4	74.7	62.6	74.9	**75.9**
DeepSVDD	61.7	**65.9**	50.8	59.1	60.9	**65.7**	67.7	**67.3**	**75.9**	73.1
CLAD (ours)	**73.0**	64.4	58.3	**60.1**	73.1	63.9	**76.0**	60.1	70.7	69.8

parameters of such modules affect the AD performance. Figure 3 shows the performance tendency when we change the number of the clusters or hidden dimension of feature z.

We vary the number of clusters from 2 to 20 and hidden dimension size from 10 to 100. The red dashed line indicates the average scores of DeepSVDD in three scenarios. If we set the number of cluster sizes to more than four, our method surpasses DeepSVDD by a huge margin. This result implies that the robustness of CLAD to the rough self-labeling. Our framework also shows the robustness with the change of the hidden dimension size $\mathbf{z}$; only marginal fluctuations are observed.

C Implementation Detail

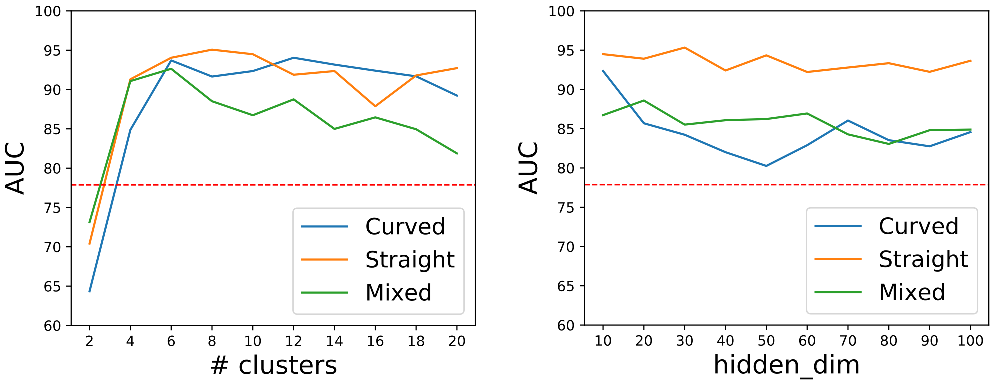
Latent feature extraction. We use autoencoder-based architecture for this network. Both encoder and decoder have five convolutional layers with increasing channel size followed by two linear layers. We additionally apply dropout (Srivastava et al., 2014) to avoid overfitting. The model training is done for 100 epochs using Adam (Kingma & Ba, 2014) with a learning rate of 0.01. Based on the model analysis, we set the number of clusters as 10 and the size of the hidden dimension as 100.

Self-labeling via clustering. We adopt DEC (Xie et al., 2016) to self-assign labels to data samples. In detail, we minimize KL divergence between the embedded data samples from encoder E and the *soft-alignments*. When training this module, an SGD optimizer was used with a momentum of 0.9 and a learning rate of 0.01 for 100 epochs.

Classifier for confidence-based AD. We use ResNet-18 (He et al., 2016) as a classifier F. We train this network using Adam (Kingma & Ba, 2014) with a learning rate of 0.0001 for 100 epochs from scratch. At the inference phase, we adopted the temperature scaling and input perturbation following ODIN (Liang et al., 2017).

D Scenario Setting

MNIST. We categorized the class labels into {`Curly, Straight, Mixed`} by the shape of the digit. When training, we choose a single scenario as normal and the rest of the others as abnormal. Note that the case where `Mixed` as normal is the most challenging since this category has similar features with other scenarios (*e.g.* ‘6’ versus ‘8’).



(a) The effect of # clusters of our scenarios.

(b) The effect of different hidden dimension sizes.

Figure 3: Ablation study results. The red dash-line denotes the average scores of DeepSVDD on our three MNIST scenarios. **(a)** With more than four clusters, our method achieves better performance compared to DeepSVDD. **(b)** We vary the hidden dimension sizes from 10 to 100. Our method shows consistent performance for all the hidden dimension sizes.

GTSRB. This dataset is originally proposed for the traffic sign recognition task. We choose the scenarios by following the subset as introduced in Stallkamp et al. (2012). With these subsets, we can simulate the abnormal cases from the driver or self-driving car perspective on various traffic signs. Table 5 shows the overall scenarios and its containing class labels.

CIFAR-10. We divided the conditions into two simple scenarios as {`Thing, Living`} to mimic the real-world anomaly detection that can be used in general object recognition applications.

Tiny-ImageNet. We first categorized class labels with representative subsets as {`Animal, Insect, Instrument, Structure, Vehicle`}. Since the class labels of this dataset are annotated based on the WordNet (Miller, 1995), we selected the equal number of the classes for each scenario by referring to the same hierarchy as shown in Table 5 and the representative images for each scenario are shown in Figure 4.

Dataset	Scenario	Class labels
MNIST	CUR (Curly)	0, 3, 8
	STR (Straight)	1, 4, 7
	MIX (Mixed)	2, 5, 6, 9
GTSRB	SPDL (Speed Limit)	Speed limit (20km/h), Speed limit (30km/h), Speed limit (50km/h), Speed limit (60km/h), Speed limit (70km/h), Speed limit (80km/h), Speed limit (100km/h), Speed limit (120km/h)
	INST (Driving Instruction)	No passing, No passing for vehicles over 3.5 metric tons, No vehicles, Vehicles over 3.5 metric tons prohibited
	WARN (Warning)	Right-of-way at the next intersection, General caution, Dangerous curve to the left, Dangerous curve to the right, Double curve, Bumpy road, Slippery road, Road narrows on the right, Road work, Traffic signals, Pedestrians, Children crossing, Bicycles crossing, Beware of ice/snow, Wild animals crossing
	DIRC (Direction)	Turn right ahead, Turn left ahead, Ahead only, Go straight or right, Go straight or left, Keep right, Keep left, Roundabout mandatory
	SPEC (Special Sign)	Priority Road, Yield, Stop, No entry
	REGN (Regulation)	End of speed limit (80km/h), End of no passing, End of all speed and passing limits, End of no passing by vehicles over 3.5 metrics tons
CIFAR-10	THG (Thing)	Airplane, Automobile, Ship, Truck
	LIV (Living)	Bird, Cat, Deer, Dog, Frog, Horse
Tiny-ImageNet	ANML (Animal)	Golden retriever, Chihuahua, German shepherd, Labrador retriever, Standard poodle, Yorkshire terrier, Cougar/Puma, Persian cat
	ISCT (Insect)	Dragonfly, Roach, Bee, Grasshopper, Fly, Mantis, Monarch butterfly, Sulphur butterfly
	ISTM (Instrument)	Water jug, Beer bottle, Tea pot, Pop bottle/Soda bottle, Beaker, Rugby ball, Volley ball, Pill bottle
	STRT (Structure)	Triumphal arch, Suspension bridge, Fountain, Viaduct, Bannister, Steel arch bridge, Obelisk, Beacon
	VHCL (Vehicle)	School bus, Trolly bus, Sports car, bullet train, Convertible, Tractor, Police van, beach wagon

Table 5: Descriptions of sub-class scenario selection for each dataset.

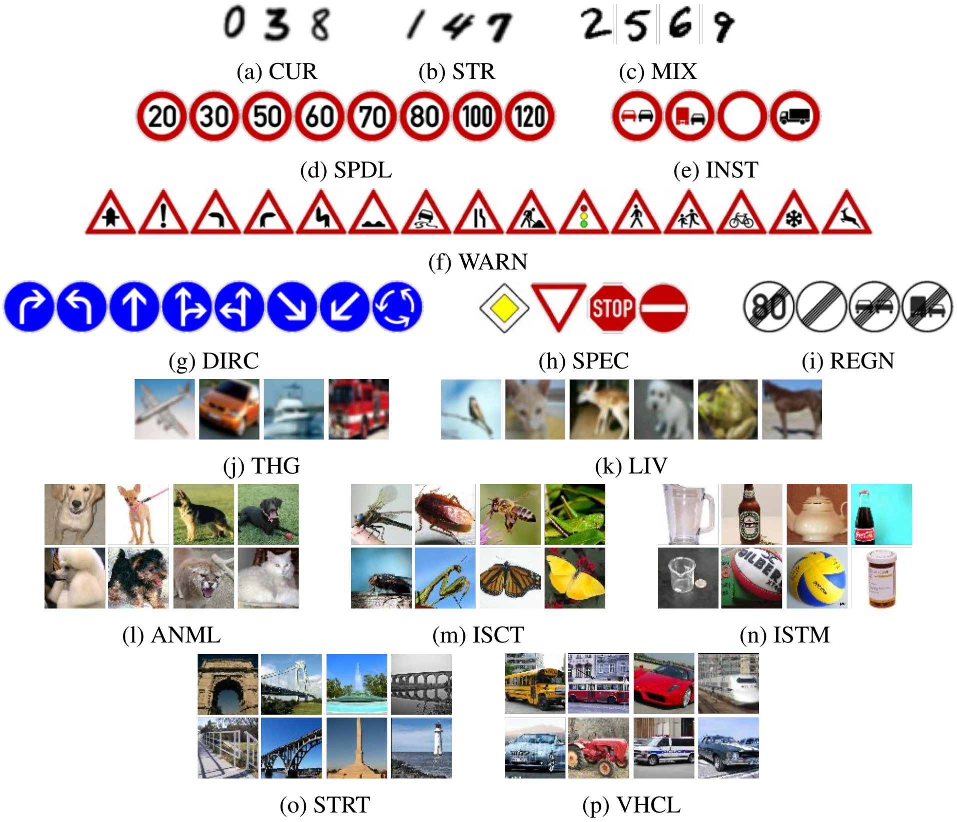


Figure 4: Representative images of the super-categories of each benchmark datasets: **(a-c)** MNIST. **(d-i)** GTSRB. **(j-k)** CIFAR-10. **(l-p)** Tiny-ImageNet.