E3Outlier: A Self-supervised Framework for Unsupervised Deep Outlier Detection

Journal:	*Transactions on Pattern Analysis and Machine Intelligence*
Manuscript ID	TPAMI-2020-06-0836.R1
Manuscript Type:	Regular
Keywords:	outlier detection, deep neural networks, unsupervised learning, self-supervised learning

Manuscript ID: TPAMI-2020-06-0836.R1

Manuscript Title: "*E*^{3}*Outlier*: A Self-supervised Framework for Unsupervised Deep Outlier Detection"

Original Title: "Self-supervised Deep Outlier Removal with Network Uncertainty and Score Refinement"

Authors: Siqi Wang, Yijie Zeng, Guang Yu, Zhen Cheng, Guang Yu, Zhen Cheng, Xinwang Liu, Sihang Zhou, En Zhu, Marius Kloft, Jianping Yin, Qing Liao

Dear editor and reviewers,

On behalf of all authors, I want to express our sincere gratitude to the editor and reviewers for those constructive comments. Based on those comments, we have made a significant modification to the original manuscript to address the issues raised by the reviewers, and more extensions have been made to enrich the content of the paper. The revised manuscript benefits greatly from these insightful and perspicacious comments. We have uploaded our responses as a separate file named "Summary of Changes" along with our revised manuscript. At the end of response letter, we also attach the revised manuscript that highlights all major changes in revision (**"Revised Manuscript with Highlights"**) for the convenience of review.

Best Regards,
Siqi Wang

RESPONSE TO COMMENTS OF ASSOCIATE EDITOR

General comments: The authors are invited to improve the article and submit a "MAJOR" revision. Please understand that this does not imply the paper will be accepted simply because a revision is made. To really be considered for acceptance, the article would need to be a truly impactful extension of the original work, not a minor one.

Response: We deeply appreciate the editor's kind decision and the precious opportunity offered for revising the manuscript. In the revised manuscript, we have done our utmost to make more meaningful extensions, which will be detailed in the rest part of the response letter. Meanwhile, all issues raised by reviewers are responded in a point-by-point manner.

Comment 1: Impact beyond original NeurIPS paper (R1-1).

Response: Please refer to our response to comment 1.1 of the reviewer #1.

Comment 2: Improved Empirical Comparison and Additional Studies needed (R1-2 and R2-2).

Response: Please refer to our responses to comment 2 of the reviewer #1 and comment 2 of reviewer #2.

Comment 3: Review work on unsupervised anomaly detection (references missing), greater depth (not just high-level) in the related work also (R1-3).

Response: Please refer to our response to comment 3 of the reviewer #1.

Comment 4: Technical points mentioned by R2 (R2-1 and R2-3).

Response: Please refer to our response to comment 1 and comment 3 of the reviewer #2.

Comment 5: There are a number of smaller local issues/concerns by both reviewers.

Response: Please refer to our point-by-point response to the rest of issues raised by reviewers.

RESPONSES TO COMMENTS OF REVIEWER #1

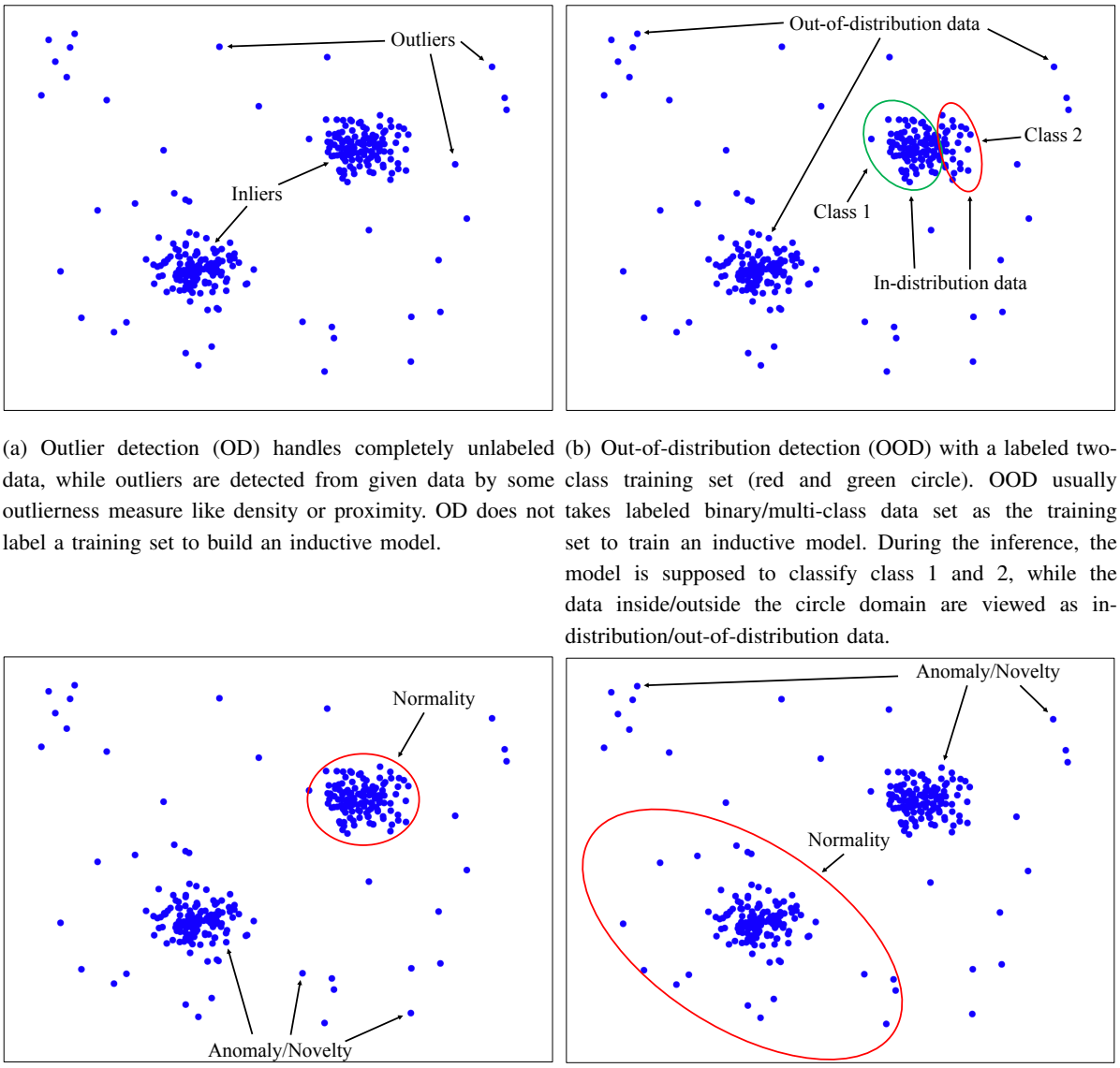
Comment 1: This journal version does not make significant extensions to the NeurIPS version.

Comment 1.1: The main technical and theoretical contributions of this paper are still the same as that in the conference version. The extensions are minor in the sense that 1) the performance improvement is rather limited compared to the method in the conference version, and 2) the technical extensions are also limited as it is easy to incorporate operations like score refinement and/or ensemble strategy to improve the performance of a model.

Response: Thank you for your comments! We would like to express our sincere gratitude to the reviewer for pointing out the drawbacks in our extension. In addition to our previous extensions, the revised manuscript further differs from the conference version [1] in terms of the following aspects: **(1)** Compared with the conference version that only explores the discriminative learning paradigm for deep OD, we further design two brand-new deep OD solutions that can leverage generative learning paradigm (see *generative* E^3*Outlier* in the new Sec. 3.6.1 of revised manuscript) and contrastive learning paradigm (see *contrastive* E^3*Outlier* in the new Sec. 3.6.2 of revised manuscript) to provide self-supervision. With the applicability to different learning paradigms, we extend the proposed E^3*Outlier* from a specific single deep OD solution to a more general self-supervised deep OD framework. **(2)** Compared with the performance in the conference version, the new contrastive E^3*Outlier* is able to achieve significant performance gain (up to 4% to 6% AUROC) on relatively difficult colored benchmarks CIFAR10/SVHN/CIFAR100 (see Sec. 4.2.1 and Table 1 of the revised manuscript). Besides, the new generative E^3*Outlier* can leverage the same CAE architecture to achieve evidently superior OD performance to existing CAE based deep OD solutions, which further justifies the effectiveness of exploiting self-supervision information in deep OD. **(3)** Compared with the conference version that only applies E^3*Outlier* to the outlier image removal task, we further demonstrate its effectiveness to another important deep OD application–unsupervised video abnormal event detection (UVAD). In particular, our evaluations on commonly-used video benchmark datasets show that our E^3*Outlier* based UVAD solution significantly outperforms state-of-the-art UVAD methods by about 4% to 10% AUROC (see the new Sec. 4.3 and Table 6 of revised manuscript). **(4)** More relevant analysis and discussion are presented in highlighted part of Sec. 4.2.1 and Sec. 4.2.3 of the revised manuscript. Besides, we clarify the novelty of exploring network uncertainty in deep OD in our response to Comment 1.2, and we add a separated paragraph at the end of Sec. 1 of the revised manuscript to summarize all extensions made in the journal version. Considering those new extensions above, we have changed the manuscript title to better reflect current content of the revised manuscript.

Comment 1.2: the authors argue the novelty of exploring network uncertainty here, but I disagree with the argument in that 1) it is widely used in highly similar tasks like out-of-distribution detection and 2) to my understanding, the gained improvement is mainly due to the score refinement operation rather than the network uncertainty.

Response: Thank you for your comments! We would like to respond to the reviewer in terms of the following aspects: **(1)** We must clarify that *outlier detection* (OD) problem discussed in this paper is an essentially different task from *out-of-distribution detection (OOD)* or *(semi-supervised) anomaly detection (AD)* problem, and the sense that they are similar mainly comes from the mixed use of terms in literature,



(a) Outlier detection (OD) handles completely unlabeled data, while outliers are detected from given data by some outlierness measure like density or proximity. OD does not label a training set to build an inductive model.

(b) Out-of-distribution detection (OOD) with a labeled two-class training set (red and green circle). OOD usually takes labeled binary/multi-class data set as the training set to train an inductive model. During the inference, the model is supposed to classify class 1 and 2, while the data inside/outside the circle domain are viewed as in-distribution/out-of-distribution data.

(c) Anomaly detection (AD) with upper right cluster labeled as normal (red circle). The training set for AD usually shares one common "normal" label when compared with OOD. The primary goal of AD is to provide a valid description of the normal data domain with a single-class training set.

(d) Anomaly detection (AD) with all lower left data labeled as normal (red circle). Note that the labeling of normal data domain can be different in AD, which may lead to evidently different detection results.

Fig. 1. The comparison of outlier detection/out-of-distribution detection/anomaly detection formulation in this paper.

i.e. OD/AD/OOD are often used interchangeably without a clear and strict definition. To avoid any further confusion to the reviewer here, we resort to the example in Fig. 1 to differentiate them in this paper:

- OD is a long-standing problem [2] that handles completely *unlabeled* data, and it aims to detect those minority data that divert significantly from the majority data using some outlierness measures (e.g. proximity or density). Meanwhile, OD follows a *transductive* learning setup, i.e. OD directly computes outlier scores of all given unlabeled data, and it does not require a separated labeled training set to establish an inductive model. For example, as shown in Fig. 1(a), without any labeled training data, two data clusters are likely to be viewed as inliers, while the rest of data that are distributed distantly are viewed as outliers. Consequently, OD is a *fully-unsupervised* task.

- OOD an is emerging topic [3] that aims to determine whether an incoming datum is from the same data distribution of an trained model's training data set. It often follows an *inductive* learning setup, as it usually involves a labeled binary/multi-class training set to train an inductive model in a supervised manner. As the example shows in Fig. 1(b), OOD leverages the labeled two-class data in the circle to train a binary classifier. It is supposed to classify newly-incoming data into two classes and exclude out-of-distribution data outside the training distribution. OOD differs from OD in two facets: 1) OOD often requires a separated labeled training set to know which data should be viewed as in-distribution data, while OD directly sorts out outliers from given unlabeled data by some outlierness measure. 2) The labeled binary/multi-class training set still provides abundant supervision information for OOD, and the OOD model (usually discriminative DNNs) can be easily trained in a *supervised* manner. It significantly facilitates OOD to learn more reasonable representations than deep OD.
- AD, which may also be referred as *novelty detection* or *one-class classification*, is another classic topic that aims to detect anomalies that are different from the labeled normal data. In fact, AD (rather than OD) is a highly similar *inductive* task to OOD, because it also requires a labeled training set to build a normality model, which is then used to discriminate anomalies or novelties in inference. However, AD's main difference from OOD is that its training data are usually labeled by one rough label ("normal" or "observed"). Due to the absence of subclass labels within training data, AD does not require classifying subclasses like OOD does during inference, and it is often viewed as a *semi-supervised* problem that aims to establish a valid description of appointed normal data domain (a.k.a data description [4]). Therefore, the labeling of normality domain plays an important role in AD: As shown in Fig. 1(c) and Fig. 1(d), AD is expected to output completely different anomalies when the labeling of normality is different. In other words, the detected anomalies/novelties are often influenced by the definition of normality rather than the data distribution itself. This is different from OD that manifests outliers by some intrinsic data characteristics within the unlabeled dataset.

The above clarifications of terms are also provided Sec. 1 of the revised manuscript (highlighted on page 1, right column, line 43-51) and Sec. 6 of supplementary material. Therefore, OD discussed in this paper is essentially a different task from OOD in the first place. Compared with OOD, OD usually follows a more traditional technical roadmap and relies on more intuitive and understandable outlierness measures, such as distance, density or clustering structure [2]. To our best knowledge, none of previous works has ever explored network uncertainty to conduct OD, although it has been used in the relevant but different task (OOD). In the meantime, the idea to use network uncertainty in OD is based on a different assumption from OOD, which makes it not as straightforward as OOD: As we explained previously, OOD is an inductive task, and the DNN is only trained on labeled in-distribution data. Thus, it is safe to assume that out-of-distribution data, which are not "seen" by the DNN in training, have large uncertainty in the inference stage. By contrast, OD is a transductive task, and in our approach both inliers and outliers are simultaneously fed into DNN to perform the same self-supervised learning process, where the process is irrelevant to whether a datum is an inlier or outlier. Therefore, the DNN has "seen" both inliers and outliers indiscriminately in the learning process, and it is no longer straightforward to assume that outliers have larger uncertainty like OOD. In fact, the assumption that outliers have large uncertainty is based on the proposed inlier priority in self-supervised learning, rather than whether a datum has been "seen" by DNN in the previous training stage like OOD. As a consequence, we believe that pointing out that network

uncertainty can be used as an effective outlierness measure for OD is novel in itself. **(2)** We would like to clarify that the major intention for introducing the concept of network uncertainty into OD is not for performance enhancement like score refinement. In our conference version [1], we only compared several outlier scores empirically, and did not look deeper into the reason *why* maximum probability and entropy based score outperform the baseline score by a notable margin in most cases. In this paper, the goal of our extension is to provide more insights into this phenomenon. To this end, we unveil the common principle, network uncertainty, behind those good-performing scores, and analyze the reason why those uncertainty based scores can be more effective for discriminative $E^3Outlier$ (presented in Sec. 3.4.1 and Sec. 3.4.2 of the original manuscript). Afterwards, we further justify our analysis by designing a new outlier score with a different uncertainty estimation method (MC-Dropout) in Sec. 3.4.3 of the original manuscript. The new score is shown to produce fairly satisfactory OD performance as well, which is consistent with our expectation. **(3)** As the reviewer suggests, we have compressed the content on network uncertainty, so as to make room for our new extensions. Yet, we still believe that the discussion on network uncertainty in deep OD is meaningful, as it will contribute to the readers' understanding in this topic.

Comment 2: The empirical comparison needs to substantially improved.

Comment 2.1: First, the competing methods have some major issues. First of all, they are overwhelmingly dominated by deep methods. Some of them are redundant, e.g., the competing methods are mainly AE-based methods and further they basically do not work at all; why do we need RSRAE and RSRAE+ since both of them work ineffectively.

Response: Thank you for your comment! We would like to make the following responses: **(1)** As the reviewer suggests, we have added more competing methods that exploit classic shallow OD models and features extracted from pretrained DNN models (see our responses to comment 2.3 below) for a more comprehensive comparison. Besides, as the reviewer suggests, we have deleted the results of RSRAE+ and only preserve RSRAE in the comparison. **(2)** The focus of this paper is deep OD with visual data like images. It is indeed a challenging topic that not much effort and progress has been made in literature (please also refer to our detailed response to comment 2.2 below) when compared with other realms like OOD and semi-supervised deep AD. As a results, we mainly compare deep methods in our previous version, and their performance is indeed unsatisfactory. Thus, we did not deliberately exclude any deep OD method that can yield better performance. In fact, such a gap is exactly the motivation of this paper.

Comment 2.2: Although there are significantly less work on deep unsupervised anomaly detection, there still exist different approaches beyond just AE-based methods (some of them may be found in recent survey papers in this area; others should be easily identified using proper key words).

Response: Thank you for your comments! We would like to make the following responses: **(1)** Actually, we have reviewed and compared both AE and non-AE based deep OD methods within our best knowledge at that time. Specifically, we have included MOGAAL [5], which is based on GANs and active learning, into our comparison, and MOGAAL is the only non-AE based deep OD method we knew at that time. In recent survey papers, AE is also described as the "commonly-used" [6], "core" [7], "central" [8] technique used on deep OD, which are consistent with our review. Hence, the dominant role of AE in deep OD is exactly the reason why most of deep OD methods reviewed in Sec. 2.2 of original manuscript are AE based.

(2) After the reviewer pointed out this issue, we have carefully reviewed literature again by inspecting recent survey papers [7], [8], [6], [2] and search engine like Google Scholar. Until the day we write this response, we only discover one additional deep OD approach named RAMODO [9] that is not based on AE. However, RAMODO performs very poorly in our experiments, because it is originally designed for tabular data and does not contain any specialized module to perform image encoding like CAE. Thus, due to the page limit, we omit the RAMODO in empirical evaluation and add the review on RAMODO to Sec. 2.2 of the revised manuscript. **(3)** Above all, we must clarify that the definition of "unsupervised deep outlier/anomaly detection" in many previous works is different from the definition of "unsupervised OD" in this paper: Due to the lack of an unified definition, most works simply view "anomaly detection" and "outlier detection" as interchangeable terms, and consider the semi-supervised case where model is trained on pure normal data to be "unsupervised". In fact, other research like the recent survey paper [6] also pointed out such a confusion in terms. However, our paper strictly differentiates two terms in this paper (see Sec. 1 on page 1, right column, line 43-51 of our paper) because they are essentially different problems (see the detailed explanation in our response to Comment 1.2 above). We indeed notice few deep methods that are not based on AE, but they are typically designed for the semi-supervised inductive setup that assumes a training set with pure normal data, which is not in the scope of our paper. As a consequence, the exploration of non-AE based deep OD is indeed insufficient.

Comment 2.3: I would suggest to include more two-stage methods as the competing methods using strong DNN architecture like ResNet50 to extract image features, such as ResNet50+LOF, ResNet50+distance-based measure, etc. This suggestion is based on the facts that 1) CAE-IF generally performs much better than CAE, I would expect very good results of IF when better deep models are used for the feature extraction, and 2) there are still many doubts in the area of outlier/anomaly detection that deep methods are better than shallow methods.

Response: Thank you for your comment! As the reviewer suggests, we have added the comparison with two-stage solutions (pretrained ResNet50+LoF and ResNet50+IF) to Sec. 4.1.2 and Table 1 of the revised manuscript. The results suggest that they can indeed achieve highly competitive performance in some cases, but they are still remarkably inferior to our discriminative and contrastive E^3*Outlier*.

Comment 2.4: Second, as the ensemble strategy is used in the proposed method, it would be an unfair comparison if it is not used in the competing methods. Please clarify.

Response: Thank you for your comment! As the reviewer points out, we test the ensemble strategy with other deep OD approaches (e.g. CAE and DRAE), and the results show that it can also achieve performance gain (typically 2% AUROC). Therefore, ensemble can actually be used as a more general score refinement technique. For a fair comparison, we instead report the raw performance of E^3*Outlier* without any score refinement in Sec. 4.2.1 and Table 1 of the revised manuscript, while we discuss the effect of score refinement separately in Sec. 4.2.2 of the revised manuscript. Note that removing the ensemble strategy actually has no influence on the superiority of our method.

Comment 2.5: Third, the introduction motivates the readers with some really practical applications, but the empirical results are just based on some popular image classification dataset benchmarks. I would suggest the authors to add some datasets from real-life application cases. For example, as the authors

mention video surveillance in the introduction and there are a number of publicly available datasets in this direction, this may be used as a very good example in the empirical justification.

Response: Thank you for your comment! As the reviewer suggests, we have designed a E^3*Outlier* based solution to the application of unsupervised video abnormal event detection (UVAD) (detailed in the new Sec. 4.3 of the revised manuscript). We evaluate our new UVAD solution on commonly-used public video datasets of this direction. The experimental results (see Table 6 of the revised manuscript) demonstrate that our solution significantly outperforms recent state-of-the-art UVAD solutions by 4% to 10% AUROC, which justifies the flexibility and effectiveness of the proposed E^3*Outlier* framework.

Comment 3: Important references are missing. Since the paper is focused on unsupervised anomaly detection, it is important to review progress in this direction, especially deep unsupervised methods as well as recently proposed shallow unsupervised methods. The current related work only has a rather high-level summarization of classic outlier detection methods and AE-based deep methods.

Response: Thank you for your comment! As the reviewer points out, we have significantly expanded the section of literature review, so as to provide a thorough and detailed review on both shallow and deep OD methods (see Sec. 2.1 and Sec. 2.2 on page 3 of the revised manuscript).

Comment 4: There are a number of over-claimed/misleading statements.

Comment 4.1: "While this setup is the most relevant and applicable one to the practical applications, it also renders OD a highly challenging problem" Do you have any evidence to support that the studied setting is the 'most relevant and applicable one'?

Response: Thanks for your comments! we realize that it is inappropriate and misleading to describe unsupervised OD as "most relevant and applicable one", so we have removed the corresponding statement. In Sec. 1 of the revised manuscript (highlighted part on page 1, line 42 of left column to line 33 of right column), we re-elaborate the importance of OD by the following new statement: "OD is of great importance in practice: First, as data labeling is usually expensive and time-consuming, it is often required to deal with massive unlabeled data. As a result, OD has been a frequently-encountered unsupervised task when handling prevalent unlabeled data. Second, even for supervised/semi-supervised tasks, OD plays a vital role in the data cleansing stage (e.g. removing wrongly-labeled data or noise when building a data set), which is the foundation for obtaining high-quality models."

Comment 4.2: "We for the first time design a flexible self-supervised learning paradigm for DNN based OD." This is not true, there are some studies exploring this framework well before your work, e.g., "Deep anomaly detection using geometric transformations." In Advances in Neural Information Processing Systems, pp. 9758-9769. 2018.

Response: Thank you for your comments! However, we would like to clarify that the NeurIPS paper mentioned by the reviewer [10] actually applies self-supervised learning to the *(semi-supervised) anomaly detection (AD)* (also named *novelty detection* or *one-class classification*), which is an easily-confused but different problem from the *outlier detection (OD)* problem discussed in this paper. Please refer to our response to Comment 1.2 for a detailed explanation on the differences between two topics. After we

carefully review the literature again, to our best knowledge, we are indeed the first work to explore and design the self-supervised learning paradigm for unsupervised deep OD.

Comment 4.3: "... generative DNNs happen to be the most frequently-used solution to unsupervised representation learning". This is incorrect. Many deep methods for outlier/anomaly detection are not generative models. The authors seem to classify generic autoencoders methods as generative models, too. This is misleading.

Response: Thank you for your comments! Based on the reviewer's comment and the our recent literature review (detailed in our response to Comment 2.2), we have removed this incorrect statement and replaced it with a more accurate description of status quo (see Sec. 3.2.1 of the revised manuscript). In the meantime, we have also revised all statement that may mislead the readers into the impression that generic autoencoders are equivalent to generative models.

Comment 5: Why do we follow a definition in a toolbox: "this paper follows the definition used in the popular open source machine learning toolbox Scikit-learn [3]"? I think stronger reasons may be found.

Response: Thanks for your comments! we do agree with the reviewer that it is inappropriate to use the definition from a toolbox, so in the revised manuscript we follow a more formal definition from the latest survey paper on outlier detection [2], which is published on a reputable journal (see highlighted part on page 1, left column, line 37-41 of the revised manuscript). In the original manuscript, we use the definition from Scikit-learn because it provides a very clear differentiation between outlier detection and anomaly/novelty detection[1]. However, we notice that many works in the literature simply use two terms interchangeably, which we believe to be misleading. In our revised manuscript, we also clarify the difference between two tasks (see page 1, right column, line 43-51 of the revised manuscript).

Comment 6: I would suggest the authors to elaborate the motivation examples in the introduction in more details, e.g., to clarify why fully unsupervised is more applicable to these settings than semi-supervised cases (having only normal data).

Response: Thank you for your comment! As we responded above, we have removed the misleading statement and re-elaborated the importance of unsupervised OD by a new statement (please refer to our response to Comment 4.1).

Comment 7: Why SSD-IF results are not found in many tables? Please clarify

Response: Thank you for your comment! In our original manuscript, we removed SSD-IF from those tables due to the limit of page width, but the results of SSD-IF were still provided in Fig. 7 of original manuscript. In the revised manuscript, we have re-arranged the table of main results, and results of SSD-IF have been added back (see Table 1 of the revised manuscript).

Comment 8: Why do we need the results on inlier detection, e.g., in table 1? Please clarify.

Response: Thank you for your comment! We believe that the reviewer was referring to the metric "AUPR-In" ("PR-I" in the revised manuscript). In fact, AUPR-In refers to the AUC of precision-recall curve when

[1]https://scikit-learn.org/stable/modules/outlier_detection.html

inliers are viewed as positive class, while inliers and outliers can both be viewed as the positive class when computing the PR curve. They actually reflect the performance of detector from two different angles. Since we do not assume preference to detecting inliers or outliers, we simply follow the standard practice in relevant realms like OOD [3] and semi-supervised AD [10], and computes both AUPR-In and AUPR-Out as parallel metrics for a more comprehensive evaluation.

RESPONSES TO COMMENTS OF REVIEWER #2

Comment 1: Some technical points are not introduced and motivated in the introduction.

Comment 1.1: How to design pseudo labels and why use transformation operation types as supervision rather than other forms of pseudo supervision? What is the underlying intuition?

Response: Thank you for your comment! In fact, using types of operations as pseudo labels is motivated by the previous work [11], which realizes highly effective unsupervised representation learning by predicting types of rotation. The main intuition behind this practice is that such a design forces DNN to capture the high-level semantics (e.g. structure and texture) in a image to fulfill such a classification task. For example, to recognize what type of rotation is imposed on the original image, the DNN must learn to localize salient object in images and recognize the orientation of its high-level parts [11], such as the head and legs of a human. However, it is not the only feasible way to introduce self-supervision: First, in the revised manuscript, we also show that the pseudo supervision can be introduced by a generative learning paradigm (see Sec. 3.6.1 of the revised manuscript) or a contrastive learning paradigm (see Sec. 3.6.2 of the revised manuscript). Second, our experiments also show that it is plausible to use a different way to assign pseudo labels, e.g. the multi-label way suggested by the reviewer (see the response to comment 2.1 below). Finally, as the reviewer suggests, we have added the explanation of this intuition to the end of Sec. 3.2.2 of the revised manuscript (highlighted part on page 5, left column, line 33-46).

Comment 1.2: Why is uncertainty better than other measures like density when using inlier priority? What is the intuition?

Response: Thank you for your comment! This is because the network uncertainty is usually directly optimized during the training of DNN (e.g. the training process will generally increase DNN's prediction probability and decrease outputs' entropy of training data), while other measures like density or proximity are not an explicit goal of DNN optimization. Therefore, we believe that network uncertainty can be a more direct indicator of inlier priority than other traditional measures, thus making it a very effective outlierness measure here. As the reviewer suggests, we have added this intuition to Sec. 3.4.1 of the revised manuscript (see highlighted part on page 7, right column, line 39-45).

Comment 2: Some studies are missing.

Comment 2.1: Since different type of transformations are used, why not juse adopt a multi-label way to supervise the network? That is, rotation type constitutes the first label, flip or not constitutes the second one, shifting type for the third and the forth one. The network predicts four labels for each image. This might be a baseline to compare.

Response: Thank you for your comment! As the reviewer suggests, we have conducted experiments to explore the possibility to use such a multi-label way for deep OD. Interestingly, the results suggest that such a multi-label way can not only achieve reasonable OD performance, but also performs slightly better than the original single-label way on most benchmark datasets. We have added the discussion on this issue to Sec. 4.2.3 of the revised manuscript (see point (4) on page 14, left column).

Comment 3: The proof in section 3.3.2 does not reflect the necessity of using the specially designed transformation discrimination task. The conclusion seems to be invariant to the chosen of self-supervised tasks or pseudo labels. If so, why not just use random labels and try to train a network to predict these random labels to see if the conclusion still holds? I know a carefully designed self-supervised task might yield good representations, but are good representations really important for outlier detection if we can simply discriminate inliers and outliers merely from inlier priority that is uncorrelated to the way of supervision?

Response: Thank you for your comment! We would like to respond in terms of the following aspects: **(1)** In the outlier image removal task, it should be noted that the difference between outliers and inliers lie in their semantics, e.g. high-level structure and appearance. To encourage the semantic similarity within inliers and maximize the semantic difference between inliers and outliers, it is necessary to learn good representations with rich semantics in the first place. In other words, a learning task that can yield semantically meaningful representations is the foundation for inliers to be semantically similar and joint their efforts into a priority against outliers. **(2)** As the reviewer suggests, we also conduct an experiment that alternate the original pseudo labels by random labels. As we expected, DNNs trained in this way yield very poor detection performance that is almost equal to random guess (approximately 50% AUROC) in the experiments. Therefore, good representation learning is of paramount importance in forming inlier priority, and we have added this explanation to the end of Sec. 3.3.4 of the revised manuscript (see highlighted part on page 7, line 56 of left column to line 21 of right column).

REFERENCES

[1] S. Wang, Y. Zeng, X. Liu, E. Zhu, J. Yin, C. Xu, and M. Kloft, "Effective end-to-end unsupervised outlier detection via inlier priority of discriminative network," in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2019, pp. 5962–5975.

[2] A. Boukerche, L. Zheng, and O. Alfandi, "Outlier detection: Methods, models, and classification," *ACM Computing Surveys (CSUR)*, vol. 53, no. 3, pp. 1–37, 2020.

[3] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," 2017.

[4] D. M. Tax and R. P. Duin, "Support vector data description," *Machine learning*, vol. 54, no. 1, pp. 45–66, 2004.

[5] Y. Liu, Z. Li, C. Zhou, Y. Jiang, J. Sun, M. Wang, and X. He, "Generative adversarial active learning for unsupervised outlier detection," *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[6] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM Computing Surveys (CSUR)*, vol. 54, no. 2, pp. 1–38, 2021.

[7] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," *arXiv preprint arXiv:1901.03407*, 2019.

[8] H. Wang, M. J. Bah, and M. Hammad, "Progress in outlier detection techniques: A survey," *IEEE Access*, vol. 7, pp. 1–1, 2019.

[9] G. Pang, L. Cao, L. Chen, and H. Liu, "Learning representations of ultrahigh-dimensional data for random distance-based outlier detection," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 2041–2050.

[10] I. Golan and R. El-Yaniv, "Deep anomaly detection using geometric transformations," in *Advances in Neural Information Processing Systems*, 2018, pp. 9758–9769.

[11] N. Komodakis and S. Gidaris, "Unsupervised representation learning by predicting image rotations," in *International Conference on Learning Representations (ICLR)*, 2018.

E^3*Outlier*: A Self-supervised Framework for Unsupervised Deep Outlier Detection

Siqi Wang, Yijie Zeng, Guang Yu, Zhen Cheng, Xinwang Liu, Sihang Zhou,
En Zhu, Marius Kloft, Jianping Yin, Qing Liao

Abstract—Existing unsupervised outlier detection (OD) solutions face a grave challenge with surging visual data like images. Although deep neural networks (DNNs) proves successful for visual data, deep OD remains difficult due to OD's unsupervised nature. This paper proposes a novel framework named *E^3Outlier* that can performs **e**ffective and **e**nd-to-**e**nd deep outlier removal. Its core idea is to introduce *self-supervision* into deep OD. Specifically, our major solution is to adopt a discriminative learning paradigm that creates multiple pseudo classes from given unlabeled data by various data operations, which enables us to apply prevalent discriminative DNNs (e.g. ResNet) to the unsupervised OD problem. Then, with theoretical and empirical demonstration, we argue that inlier priority, a property that encourages DNN to prioritize inliers during self-supervised learning, makes it possible to perform end-to-end OD. Meanwhile, unlike frequently-used outlierness measures (e.g. density, proximity) in previous OD methods, we explore network uncertainty and validate it as a highly effective outlierness measure, while two practical score refinement strategies are also designed to improve OD performance. Finally, in addition to the discriminative learning paradigm above, we also explore the solutions that exploit other learning paradigms (i.e. generative learning and contrastive learning) to introduce self-supervision for *E^3Outlier*. Such extendibility not only brings further performance gain on relatively difficult datasets, but also enables *E^3Outlier* to be applied to other OD applications like video abnormal event detection. Extensive experiments demonstrate that *E^3Outlier* can considerably outperform state-of-the-art counterparts by 10%-30% AUROC. All codes are available at https://github.com/demonzyj56/E3Outlier.

Index Terms—outlier detection, deep neural networks, unsupervised learning, self-supervised learning

1 Introduction

In realms like machine learning and data science, outliers, which are also called novelties, anomalies, deviants, exceptions, irregularities, etc [1], have a pervasive existence. Outlier detection (OD), which may also be referred as unsupervised anomaly/outlier detection, is a long-standing problem that draws continuous attention from the research community. To provide a clear and strict formulation of OD problem, this paper follows the definition used in the recent OD survey paper [2]: Given a set of data instances, OD is an unsupervised task that aims to identify those instances that deviate significantly from the rest of data. Thus, outliers are discerned from given unlabeled data by a *transductive* learning setup. OD is of great importance in practice: First, as data labeling is usually expensive and time-consuming, it is often required to deal with massive unlabeled data. As a result, OD has been a frequently-encountered unsupervised task when handling prevalent unlabeled data. Second, even for supervised/semi-supervised tasks, OD plays a vital role in the data cleansing stage (e.g. removing wrongly-labeled data or noise when building a data set), which is the foundation for obtaining high-quality models. OD enjoys a variety of real-world applications, such as financial fraud detection [3], emerging topic detection [4], computer-aided medical diagnosis [5], motion trajectory analysis [6], etc. Since the only prior knowledge is that outliers have rare occurrence when compared with inliers, no supervision information is available for OD here. Due to its unsupervised nature, OD is usually addressed by exploiting some intrinsic properties of data, e.g. density, proximity, cluster membership, etc. A more detailed review of classic OD is given in Sec. 2.1. In particular, we distinguish OD in this paper from the *(semi-supervised) anomaly detection* or *one-class classification* [7], which builds a normality model from a pure set of labeled normal data and detects deviants in a separated test set by an *inductive* learning setup. To avoid any confusion, a detailed clarification of terms is also provided in Sec. 6 of the supplementary material, so as to differentiate OD here from other relevant but different realms like (semi-supervised) anomaly detection and out-of-distribution detection.

With the widespread use of photographic equipment (e.g. cameras, smart phones), visual data like images and videos have undergone an explosive growth in these years. In this context, a marriage of OD and visual data is pretty natural, and it gives birth to many novel applications, such as the refinement of web image search results [8], [9] and video abnormal event detection [10], [11]. Among various forms of visual data, images have constantly played

- *S. Wang, G. Yu, Z. Cheng, X. Liu and E. Zhu are with College of Computer, National University of Defense Technology (NUDT), Changsha, 410073, China. E-mail: {wangsiqi10c, xinwangliu, enzhu}@nudt.edu.cn.*
- *Y. Zeng is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798. E-mail: yzeng004@e.ntu.edu.sg.*
- *S. Zhou is with College of Intelligent Science and Technology, NUDT, Changsha, 410073, China. E-mail: sihangjoe@gmail.com.*
- *Marius Kloft is with Department of Computer Science, TU Kaiserslautern, Germany. E-mail: kloft@cs.uni-kl.de.*
- *J. Yin is with Dongguan University of Technology, Dongguan, 523808, China. E-mail: jpyin@dgut.edu.cn.*
- *Q. Liao is with School of Computer Science and Technology, Harbin Institue of Technology (Shenzhen), China. E-mail: liaoqing@hit.edu.cn.*

Manuscript received June 30, 2020.

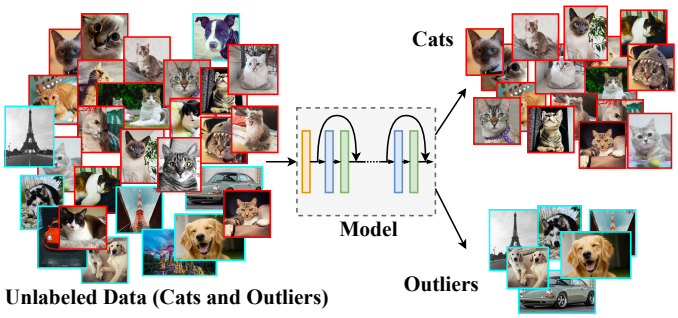


Fig. 1: An example of deep outlier image removal task.

a fundamental role in all sorts of visual analysis. Therefore, this paper will focus on OD for image data, i.e. the image outlier removal task. For an intuitive illustration, we show an example that aims to remove outliers from images of cats (inliers) in Fig. 1. Compared with frequently-seen tabular data (or vectorized data), image data exhibit evidently different characteristics: They possess a variety of high-level spatial structures that are endowed with rich semantics, and low-level details (i.e. image pixels) alone are much less meaningful to perception. As a consequence, a direct application of those classic OD methods to image data usually leads to poor performance, and proper image representations will be a prerequisite for successful outlier removal. As a simple solution, some works [8], [12] extract the image representations by hand-crafted feature descriptors (e.g. SIFT [13], sparsity-constrained linear coding [14]), and then feed the extracted feature vectors into a classic OD method. However, such solutions bring about complex feature engineering issues, and they often suffer from sub-optimal image representations and poor transferability. To this end, an emerging trend is to learn good representations automatically via deep neural networks (DNNs) during the learning process, so as to realize a certain goal like image classification or segmentation. Such an end-to-end deep learning paradigm has achieved remarkable success in computer vision, especially with discriminative DNNs for supervised learning tasks [15]. However, although introducing DNNs for deep outlier removal seems to be pretty straightforward, a both *effective* and *end-to-end* DNN based OD solution still requires exploration. The major impediment to developing such a solution lies in the unsupervised nature of the OD task, i.e. the absence of data labels results in a lack of supervision signal. Consequently, as several recent surveys point out [2], [16], [17], [18], auto-encoder (AE) still plays a dominant role in deep OD, while other widely-used DNNs like discriminative ResNet [19] are not directly applicable for deep OD without any given labels.

To bridge those gaps in deep OD, we propose the first self-supervised framework termed $E^3Outlier$, which aims to realize both *effective* and *end-to-end* deep outlier removal. Specifically, our core idea is to remedy the label absence in OD by introducing self-supervision. To this end, our major solution is to create multiple pseudo classes from given unlabeled data by imposing certain data operations like rotation and patch re-arranging. With labels of those pseudo classes, powerful discriminative DNNs that have been thoroughly studied can be exploited in OD and enable more effective representation learning. Second, in order to further conduct end-to-end OD, we unveil a property named "inlier priority": Even though inliers and outliers are indiscriminately fed into the DNN during self-supervised learning, the DNN tends to prioritize inliers' loss reduction. We provide both theoretical and empirical demonstration to this property. Third, instead of commonly-used outlierness measure (e.g. density and proximity), we point out that the DNN uncertainty in self-supervised learning can be leveraged to design highly effective outlier scores. Meanwhile, inspired by the inlier priority and network uncertainty, we develop two practical strategies and fuse them into a score refinement stage to yield performance enhancement. Finally, in addition to the aforementioned discriminative learning paradigm, we further design the solution to leverage generative/contrastive learning paradigm to perform self-supervised learning for the proposed $E^3Outlier$ framework. With the extendibility to different learning paradigms, $E^3Outlier$ is not only able to be flexibly applied to other OD applications like video abnormal event detection, but also yield further performance gain on relatively difficult datasets. Our main contributions can be summarized below:

- We for the first time design a self-supervised learning framework for DNN based OD. It not only eases the lack of supervision, but also enables discriminative DNNs to be directly applied to the deep OD problem.
- We unveil a property named inlier priority during self-supervised learning, and theoretical and empirical demonstration are presented to justify this property. It lays the foundation to perform end-to-end OD with the proposed $E^3Outlier$ framework.
- We point out that the uncertainty of discriminative DNN can be exploited as a novel outlierness measure in deep OD, and develop several highly effective uncertainty based outlier scores for end-to-end OD. Moreover, we propose joint score refinement with two practical strategies to boost the OD performance.
- We further design solutions that incorporates generative learning and contrastive learning paradigm into the $E^3Outlier$ framework to provide self-supervision, which endows the proposed framework with more flexibility and better OD performance.

An earlier version of this paper is reported in [20], and this paper is mainly extended in terms of the following aspects: (1) This paper explicitly points out that DNN uncertainty can be used as a new outlierness measure, and intuitively unveils the connection among OD, self-supervised learning and network uncertainty. Compared with this paper, [20] just reported empirical comparison of different outlier scores and did not provide in-depth analysis into the underlying principle of score design. (2) We design several practical strategies to conduct outlier score refinement, which enables the model to achieve consistent performance enhancement against the performance reported in [20] on all benchmark datasets. (3) Unlike [20] that only exploited discriminative learning paradigm for deep OD, this paper further validates the applicability of generative learning or contrastive learning paradigm to $E^3Outlier$. (4) Apart from the image outlier removal task in [20], this paper shows that the proposed $E^3Outlier$ framework is also able to achieve

superior performance in other deep OD application like unsupervised video abnormal event detection.

2 Related Work

2.1 Shallow Model based Outlier Detection

A vast number of shallow methods have been proposed to handle OD, and they usually fall into the following categories: (1) Proximity based methods, which measure the outlierness of a datum by its relation to its neighboring data. Early methods of this type simply assume the data density to be homogeneous, and define some intuitive quantities as outlier scores, such as the distance to the k-th neareast neighbors (k-nn) [21] and the number of neighbors within a pre-defined radius [22]. To this end, Local Outlier Factor (LoF) [23] is the first work that considers local outliers using the average ratio of one datum's neighbor's local reachability density to its own reachability density, which inspires numerous subsequent works, e.g. Connectivity-based Outlier Factor (CoF) [24] considers the degree of connectivity among data when computing outlier scores, while Local Outlier Probability (LoOP) [25] estimates the probability of being an outlier by assuming a half-Guassian distribution on a datum's distance to its k-nn. As computing k-nn can be time-consuming, recent works [26], [27] propose to leverage subsampling and achieve linear time complexity. (2) Statistics based methods, which view data endowed with low likelihood as outliers. The likelihood can be estimated by several statistical models, including parametric and non-parametric statistical models. As to parametric models, the most representative model is Gaussian Mixture Model (GMM) [28], and recently a more robust GMM based OD approach is proposed by Tang et al. [29] by incorporating subspace learning. Meanwhile, as to non-parametric models, kernel density estimation (KDE) [30] is frequently used for OD, while and its its recent variants like [31], [32], [33] are developed to improve its efficiency of OD. (3) Clustering based methods, which view data that do not belong to any major data cluster as outliers. For example, Jiang et al. [34] perform OD by a modified k-means algorithm and constructing a minimal spanning tree from cluster centers. He et al. [35] combine LoF and clustering into CBLOF, which quantitatively distinguishes small and large clusters. To avoid specifying the number of clusters, a recent work by Yan et al. [36] propose to leverage Gibbs Sampling of Dirichlet Process Multinomial Mixture (GSDPMM) for OD. Chenaghlou et al. [37] extends the clustering based OD to online streaming data by considering the evolve of clusters. (4) Projection based methods, which project the original data into a new space to manifest outlierness. Concretely, data can be projected into a low-dimensional embedding by dimension reduction techniques like principal component analysis (PCA) [38] or neural networks like shallow autoencoders [39], and outliers are viewed to be those data that are poorly recovered from the embeddings. In particular, Liu et al. [40] propose Isolation Forest (IF), which projects input data into the tree nodes of random binary trees, and then discriminate outliers by the depth of tree nodes. IF proves to be a both effective and efficient OD method, while recent works by Hariri [41] propose to further improve IF by using random hyperplane cut. Besides, projection techniques like local sensitivity hashing [42] and random projection [43] are also used to reduce complexity of OD models. A more comprehensive review on shallow OD methods can be found in recent survey papers [2], [16], [17], [18]

2.2 DNN based Outlier Detection

As a newly-emerging topic, DNN based OD is highly challenging as it requires to learn suitable data representations for OD. To our best knowledge, only few attempts have been made in the literature. A straightforward idea is to exploit a two-stage solution, which performs representation learning by DNNs first, and then feeds learned features into a separated module that is implemented by some classic OD model (reviewed in [44]). However, such two-stage approaches may suffer from the incompatibility between learned features and the OD module, which can lead to sub-optimal performance. By contrast, state-of-the-art methods usually conduct a joint learning of data representations and outlier scores, and we review each existing solution to our best knowledge below: Xia et al. [9] design a new loss function that encourages a better separation of inliers and outliers by minimizing intra-class variance for multi-layer AE, and propose an adaptive thresholding technique to discriminate outliers; Zhai et al. [45] connect an energy based model with a regularized AE, and develop an energy based score for OD; Zhou et al. [46] utilize a combination of deep AE and Robust Principal Component Analysis (RPCA), which decomposes the matrice of unlabeled data into a low-rank part and a sparse part to represent inliers and outliers respectively, while Chalapathy et al. [47] also adopt a similar idea; Chen et al. [39] propose to generate a set of AEs that possess randomly varied connectivity architecture to perform OD, while adaptive sampling is leveraged to make the approach more efficient and effective. Inspired by Gaussian Mixture Model (GMM), Zong et al. [48] focus on developing an end-to-end OD solution that embeds a GMM density estimation network into the deep AE, and both components are optimized simultaneously; Unlike other methods that rely on AEs, Pang et al. [49] propose a ranking-model based framework named RAMODO, which can be readily incorporated into random distance based OD approach to perform efficient OD with tabular data; Liu et al. [50] convert OD into a binary classification problem via generative adversarial networks (GANs) [51], which are modified to generate simulated outliers; The most recent work [52] exploits the latent low-dimensional subspace structure in data by adding a Robust Subspace Recovery (RSR) regularizer into AE, and two variants, RSRAE and RSRAE+, are proposed for deep outlier removal. As several recent surveys point out [2], [17], [18], AE still plays a center role in existing deep OD solutions due to its unsupervised nature, which motivates us to develop $E^3Outlier$.

2.3 Self-supervised Learning and Network Uncertainty

Self-supervised learning, which is also known as surrogate supervision [53] based learning or pseudo supervision [54] based learning, enjoys a swift growth of popularity in recent research. Its core idea is to construct additional supervision signals from given data by introducing a pretext task. The learning targets of pretext task can be obtained by numerous

ways, such as clustering [55], geometric transformations [56], [57], masking [58], image patch permutation [59], time sequence shuffling [60], contrastive learning [61], etc. As a highly effective pre-training technique or auxiliary task to improve the performance of high-level downstream tasks, self-supervised learning has been explored in many application scenarios, such as image classification, semantic segmentation, object detection and action recognition [62]. To our best knowledge, this is the first work that connects self-supervised learning to unsupervised outlier analysis.

DNN's uncertainty reflects its confidence to a certain prediction, which usually makes it a concept for inductive learning. Several methods have been proposed to quantify network uncertainty, such as Bayesian Neural Networks (BNN) [63], Monte Carlo dropout (MC-Dropout) [64], model ensemble [65], maximum softmax probability [66], information entropy [67], etc. Despite that network uncertainty has drawn increasing attention, its application is typically limited to knowing whether DNN makes trustworthy predictions or detecting the dataset shift. In this paper, we for the first time discuss network uncertainty under a transductive setup, and demonstrate that it can serve as a fairly effective outlierness measure for DNN based OD.

3 The Proposed Framework

3.1 Problem Formulation

Suppose that the data space spanned by all images is denoted by $\mathcal{X}$. DNN based OD deals with a completely unlabeled image data collection $X \subseteq \mathcal{X}$ that is contaminated by outlier images. In other words, X consists of an inlier set X_{in} and an outlier set X_{out}, while $X = X_{in} \cup X_{out}$ and $X_{in} \cap X_{out} = \emptyset$. By the definition of outliers [68], image data of the inlier set are from the same underlying distribution that shares close semantics, but outliers originate from different distributions. Given any image $\mathbf{x} \in \mathcal{X}$, DNN based OD intends to build a scoring model $S(\cdot)$, which takes raw $\mathbf{x}$ as the input and does not perform any prior feature extraction. The goal of $S(\cdot)$ is to output $S(\mathbf{x}) = 1$ for any inlier $\mathbf{x} \in X_{in}$, while $S(\mathbf{x}) = 0$ for any outlier $\mathbf{x} \in X_{out}$. In practice, a larger output $S(\mathbf{x})$ signifies a lower likelihood to be an outlier for $\mathbf{x}$. Besides, within the domain of DNN based OD, *end-to-end* OD refers to the case where both representation learning and OD can be carried out by the same DNN, and no separated classic OD method is involved. In this paper, the proposed *E^{3}Outlier* framework aims to achieve both effective and end-to-end OD.

3.2 Discriminative *E^{3}Outlier*

3.2.1 Motivation

As reviewed in Sec. 2.2, it is noted that AE based solutions play a center role in the deep OD task due to its unsupervised setup. Specifically, deep AE based solutions typically perform unsupervised representation learning by learning to reconstruct the inputs, which is realized by training the deep AE to reduce pixel-wise reconstruction errors like mean square errors (MSE). However, recent researches like [69], [70] demonstrate that such a pixel-wise reconstruction tends to overemphasize low-level image details, which are of very limited interest to human perception. By contrast, semantics of high-level image structures are ignored, but they are actually pivotal to DNN based OD. Another emerging type of generative DNNs is GANs. Despite of fruitful progress, it is still challenging to integrate them into OD [71]: First, it is actually difficult to generate sufficient realistic image outliers, as potential image outliers are infinite and generating high-quality image outliers by GANs is still an open topic; Second, efficient representation learning with GANs is neither straightforward nor easy. By comparison, the supervised discriminative learning paradigm is still the most effective way to learn image semantics and capture high-level structures so far. As a result, these reasons above motivate us to introduce *self-supervision*, so as to enable the use of discriminative learning paradigm in OD.

3.2.2 Self-supervised Discriminative Network (SSD)

The availablity of supervision signals is the key to introduce discriminative DNNs like ResNet [19] and Wide ResNet (WRN) [72] to OD. As image classification is the most fundamental task in supervised learning, creating several pseudo classes from given unlabeled data is a natural idea. Instead of generating a pseudo outlier class like [50], which is a straightforward but difficult task, we propose to build self-supervision by exerting some frequently-seen data operations on given images. Those new data produced by a certain operation are viewed as one pseudo class. Afterwards, we can readily realize representation learning with a discriminative DNN by training it to classify those created pseudo classes. As the discriminative DNN is guided by self-supervision, we term it *self-supervised discriminative network (SSD)* here. Formally, supposing a set of K operations $\mathcal{O} = \{O(\cdot|y)\}_{y=1}^{K}$ is designed to create pseudo classes, we impose the y-th operation $O(\cdot|y)$ on an unlabeled image $\mathbf{x}$ (regardless of an inlier or outlier) and produce a new image $\mathbf{x}^{(y)} = O(\mathbf{x}|y)$. In this way, we can create the y-th pseudo class $X^{(y)} = \{\mathbf{x}^{(y)}|\mathbf{x} \in X\}$, with the pseudo label y assigned to all data in this class. Then, given all data $X' = \{X^{(1)}, \cdots, X^{(K)}\}$ and their label set Y, an SSD with a K-node Softmax layer is trained to perform classification. Like the standard classification process, the SSD is supposed to classify a datum $\mathbf{x}^{(y')}$ into the y'-th pseudo class. The probability vector of $\mathbf{x}^{(y')}$ output by SSD's Softmax layer is denoted as $\mathbf{P}(\mathbf{x}^{(y')}|\boldsymbol{\theta}) = [P^{(y)}(\mathbf{x}^{(y')}|\boldsymbol{\theta})]_{y=1}^{K}$, where $P^{(y)}(\cdot)$ and $\boldsymbol{\theta}$ indicate the probability from the y-th node of Softmax layer and DNN's leanable parameters respectively. To train the SSD, we can minimize the following objective function:

$$\mathcal{L}_{SSD} = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_{SS}(\mathbf{x}_i|\boldsymbol{\theta}) \tag{1}$$

where $\mathcal{L}_{SS}(\mathbf{x}_i|\boldsymbol{\theta})$ represents the loss incurred by $\mathbf{x}_i$ in X during the self-supervised learning. When the standard cross-entropy loss is used, $\mathcal{L}_{SS}(\mathbf{x}_i|\boldsymbol{\theta})$ takes the form below:

$$\mathcal{L}_{DSS}(\mathbf{x}_i|\boldsymbol{\theta}) = -\frac{1}{K}\sum_{y=1}^{K}\log(P^{(y)}(\mathbf{x}_i^{(y)}|\boldsymbol{\theta})) \tag{2}$$

Another key to SSD is the design of data operation. We introduce three sets of operations: Regular affine operation set $\mathcal{O}_{RA}$, irregular affine operation set $\mathcal{O}_{IA}$ and patch rearranging operation set $\mathcal{O}_{PR}$. The general intuition behind

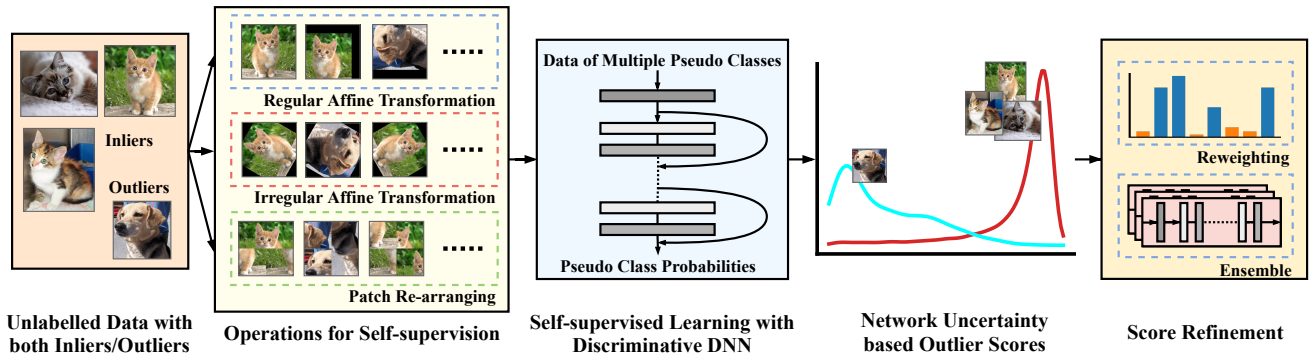


Fig. 2: Overview of the proposed discriminative E^3*Outlier* for deep OD.: Given unlabeled image data polluted by outliers, three operation sets are first imposed on images to create multiple pseudo classes and provide self-supervision. Then, a discriminative DNN is trained to perform the self-supervised learning, i.e. learning to classify those created pseudo classes. Next, the outlierness of each image is measured by the proposed network uncertainty based outlier score. Finally, the joint score refinement with re-weighting and ensemble strategy can be used to further boost the OD performance of E^3*Outlier*.

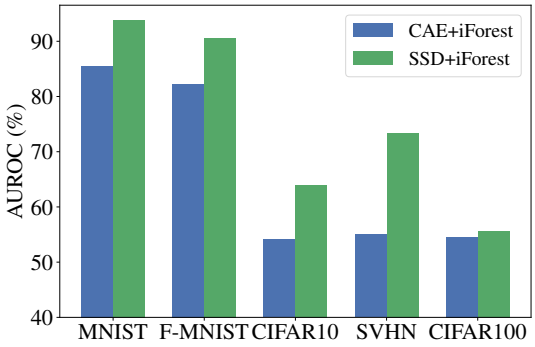


Fig. 3: Comparison of learned image representations.

those operations is to force DNN to capture the semantics of high-level structures in an image when it is required to fulfill such a classification task. For example, to recognize what type of rotation is imposed on the original image, the DNN must learn to localize salient object in images and recognize the orientation of its high-level parts, such as the head and legs of a human. Due to the page limit, we illustrate the details of data operation design in Sec. 1 of the supplementary material. Due to the prevalence of discriminative DNNs, creating pseudo classes by data operations is an intuitive and convenient way to provide self-supervision for deep OD. The overview of discriminative E^3*Outlier* is presented in Fig. 2. However, we will show other learning paradigms are also applicable to the proposed E^3*Outlier* later.

3.2.3 Comparison between SSD and AE

To verify whether SSD can learn better image representations, we conduct a simple experiment that compares SSD with Convolutional AE (CAE). We select WRN-28-10 [72] as SSD and adopt the CAE architecture in [57], which has a close depth to the SSD.Then, we extract the outputs of SSD's penultimate layer as learned representations, while the outputs of CAE's intermediate layer are extracted for comparison (note that they share the same dimension). With the protocol described in Sec. 4.1 to evaluate the OD performance on image datasets, learned representations of SSD and CAE are both fed into an Isolation Forest (IF) model with the same parameterization to conduct OD. The comparison is shown in Fig. 3: On those image benchmarks, learned representations of SSD are always able to improve IF's OD performance, which justifies SSD's effectiveness.

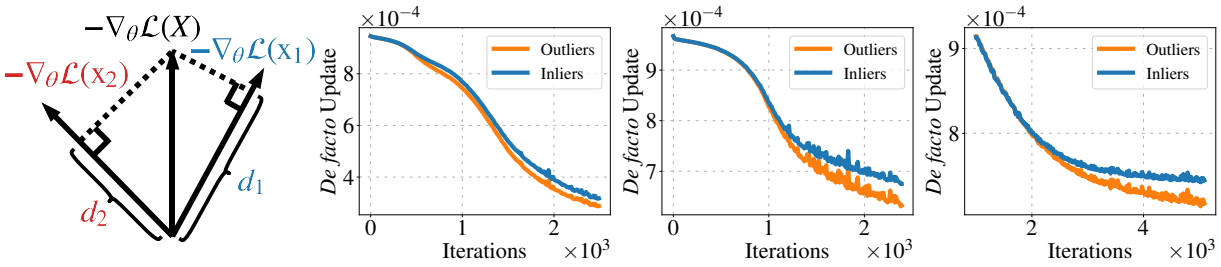
3.3 Inlier Priority: Foundation of End-to-end OD

3.3.1 Motivation

Although the proposed SSD achieves more effective representation learning than CAE, there are still some problems: First, without using a specialized OD network like [48], the proposed paradigm actually learns a pre-text task (i.e. classification) instead of OD, so by now we cannot draw OD results directly from SSD alone; Second, although we can resort to a classic OD model like we did in Sec. 3.2.3, such a two-stage solution can be sub-optimal as learned representations and the OD model are not jointly optimized. In fact, the OD performance of SSD+IF solution in Sec. 3.2.3 indeed has room for improvement (60%-70% AUROC) on relatively difficult benchmarks, i.e. CIFAR10/SVHN/CIFAR100. Therefore, an end-to-end solution is favorable for deep OD. However, for the proposed SSD, data operations are equally imposed on both inliers and outliers to create a pseudo class, and they are indiscriminately fed into DNN for training. Thus, it is still not sure whether inliers and outliers will behave differently during the self-supervised learning. This motivates us to explore this issue below from both theoretical and empirical view.

3.3.2 The Theoretical View

First of all, we approach this issue from a theoretical view. Since the theoretical analysis of DNNs remains particularly difficult, we consider a simplified case that is analyzable: We choose a feed-forward network with a single hidden layer and sigmoid activation to be SSD. Suppose that the hidden layer and Softmax layer have $(L+1)$ and K nodes respectively. Parameters of the simple SSD is randomly initialized by an i.i.d uniform distribution on $[-1, 1]$. Since neural networks are usually optimized by gradient descent, the influence of inliers and outliers imposed on the SSD can be reflected by the gradients that they back-propagate to update the network parameters. Hence, we



(a) *De facto* update (b) MNIST ("3") (c) Fashion-MNIST ("bag") (d) CIFAR10 ("horse")

Fig. 4: An illustration of *de facto* update and the average *de facto* update of inliers/outliers during the network training. The class used as inliers is in brackets.

analyze gradients w.r.t the weights associated with the c-th class ($1 \leq c \leq K$) between the hidden layer (it is also the penultimate layer in this case) and the final Softmax layer, $\mathbf{w}_c = [w_{s,c}]_{s=1}^{(L+1)}$ ($w_{L+1,c}$ is the bias), which are directly responsible for making SSD's predictions. We discuss the case of iniers (X_{in}) first: For the cross-entropy loss $\mathcal{L}$ that is used in our case, only those data yielded by imposing the c-th operation on X_{in} are used to update $\mathbf{w}_c$, i.e. $X_{in}^{(c)} = \{\mathbf{x}^{(c)} = O(\mathbf{x}|c)|\mathbf{x} \in X_{in}\}$. The gradient vector incurred by $X_{in}^{(c)}$ is denoted by $\nabla_{\mathbf{w}_c}\mathcal{L} = [\nabla_{w_{s,c}}\mathcal{L}]_{s=1}^{(L+1)}$, and each element of $\nabla_{w_{s,c}}\mathcal{L}$ is given by:

$$\nabla_{w_{s,c}}\mathcal{L} = \sum_{i=1}^{N_{in}} \nabla_{w_{s,c}}\mathcal{L}(\mathbf{x}_i) = \sum_{i=1}^{N_{in}} (P^{(c)}(\mathbf{x}_i) - 1)h^{(s)}(\mathbf{x}_i) \tag{3}$$

where $N_{in} = |X_{in}^{(c)}| = |X_{in}|$ is the number of inliers. For $\mathbf{x}_i \in X_{in}^{(c)}$, $P^{(c)}(\mathbf{x}_i)$ is the output of c-th node in the Softmax layer, and $h^{(s)}(\mathbf{x}_i)$ is the output of s-th node in the penultimate layer. To quantify inliers' influence on a randomly initialized SSD, a direct indicator can be the expectation of inliers' gradient magnitude to update $\mathbf{w}_c$, $E^{(in)}(||\nabla_{\mathbf{w}_c}\mathcal{L}||_2^2)$. Thus, our goal is to obtain:

$$E^{(in)}(||\nabla_{\mathbf{w}_c}\mathcal{L}||_2^2) = E\big(\sum_{s=1}^{L+1}(\nabla_{w_{s,c}}\mathcal{L})^2\big) = \sum_{s=1}^{L+1} E\big((\nabla_{w_{s,c}}\mathcal{L})^2\big) \tag{4}$$

By addition in (3), computing (4) requires the term below:

$$\begin{aligned} E\big((\nabla_{w_{s,c}}\mathcal{L})^2\big) &= E\big((\sum_{i=1}^{N_{in}} \nabla_{w_{s,c}}\mathcal{L}(\mathbf{x}_i))^2\big) \\ &= \sum_{i=1}^{N_{in}}\sum_{j=1}^{N_{in}} E(\nabla_{w_{s,c}}\mathcal{L}(\mathbf{x}_i)\nabla_{w_{s,c}}\mathcal{L}(\mathbf{x}_j)) \end{aligned} \tag{5}$$

To compute (5), in our case we can resort to the second-order Taylor series expansion to derive the approximation below (detailed in Sec. 2 of the supplementary material):

$$\begin{aligned} &E(\nabla_{w_{s,c}}\mathcal{L}(\mathbf{x}_i)\nabla_{w_{s,c}}\mathcal{L}(\mathbf{x}_j)) \approx \\ &h^{(s)}(\mathbf{x}_i)h^{(s)}(\mathbf{x}_j)\Big[\frac{(K-1)^2}{K^2} + \frac{K-1}{3K^3}\sum_{t=1}^{L+1} h^{(t)}(\mathbf{x}_i)h^{(t)}(\mathbf{x}_j)\Big] \end{aligned} \tag{6}$$

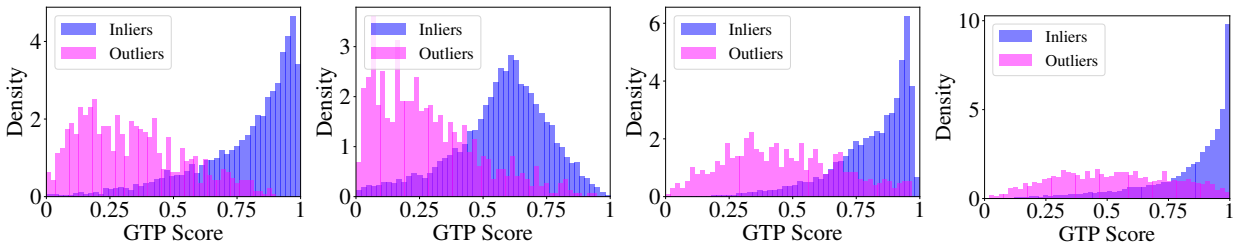
There remains to calculate $h^{(t)}(\mathbf{x}_i)h^{(t)}(\mathbf{x}_j)$ in (6). In this case, [73, Lemma 3.b] has proved that the expectation of $h^{(s)}(\mathbf{x}_i)h^{(s)}(\mathbf{x}_j)$ w.r.t the randomly initialized weights between the input and hidden layer satisfies $E(h^{(s)}(\mathbf{x}_i)h^{(s)}(\mathbf{x}_j)) \approx \frac{1}{4}$ and $E(h^{(s)}(\mathbf{x}_i)^2 h^{(s)}(\mathbf{x}_j)^2) \approx \frac{1}{16}$. Thus, by definition of $||\nabla_{\mathbf{w}_c}\mathcal{L}||_2^2$ in (4) and (5), we yield:

$$\begin{aligned} E^{(in)}(||\nabla_{\mathbf{w}_c}\mathcal{L}||_2^2) &\approx N_{in}^2\Big[(L+1)(\frac{(K-1)^2}{4K^2} + \frac{(K-1)(L+1)}{48K^3})\Big] \\ &\triangleq N_{in}^2 \cdot Q \end{aligned} \tag{7}$$

Since L, K above are both fixed, Q is a constant. As a result, (7) shows that for the self-supervised learning of SSD, $E^{(in)}(||\nabla_{\mathbf{w}_c}\mathcal{L}||_2^2)$ is roughly proportional to N_{in}^2. Likewise, we can also derive that the expectation of outliers' gradient magnitude is $E^{(out)}(||\nabla_{\mathbf{w}_c}\mathcal{L}||_2^2) = N_{out}^2 \cdot Q$. Since $N_{in} \gg N_{out}$ is a indispensable premise for the OD task, we have $E^{(in)}(||\nabla_{\mathbf{w}_c}\mathcal{L}||_2^2) \gg E^{(out)}(||\nabla_{\mathbf{w}_c}\mathcal{L}||_2^2)$, which leads to an interesting conclusion: Although inliers and outliers are equally used for the self-supervised learning of SSD, the gradients contributed by inliers are much more important than outliers. Since those back-propagated gradients are used to train SSD, the theoretical analysis leads to an underlying property: *SSD is inclined to prioritize inliers during self-supervised learning*, which is named *inlier priority* in this paper. Such a property implies that inliers and outliers behave differently in self-supervised learning, which makes it possible to establish an end-to-end OD solution. Since it is intractable to compute $E(h^{(t)}(\mathbf{x}_i)h^{(t)}(\mathbf{x}_j))$ for more complex SSD, we will further validate inlier priority by empirical validations in the next section.

3.3.3 Empirical Validations

To further validate the property of inlier priority empirically, we propose to calculate a more direct indicator named "*de facto* update" for inliers and outliers respectively: In addition to gradient magnitude that we have considered in previous theoretical analysis, another important attribute of gradient vectors is gradient direction. As illustrated by Fig. 4a, consider $\mathbf{x}_i$ from a batch of data X (we slightly abuse the notation of X here). The negative gradient $-\nabla_{\boldsymbol{\theta}}\mathcal{L}(\mathbf{x}_i)$ is the fastest network updating direction to reduce $\mathbf{x}_i$'s loss. However, the network weights $\boldsymbol{\theta}$ are actually updated by the averaged negative gradient of the entire batch X, $-\nabla_{\boldsymbol{\theta}}\mathcal{L}(X) = -\frac{1}{N}\sum_i \nabla_{\boldsymbol{\theta}}\mathcal{L}(\mathbf{x}_i)$. Thus, the actual updating direction at each iteration is usually different from the best updating direction for each individual datum. To measure the actual gradient magnitude that $\mathbf{x}_i$ obtains along its best direction



(a) MNIST ("3") (b) Fashion-MNIST ("bag") (c) CIFAR10 ("horse") (d) SVHN ("3")

Fig. 5: Normalized histograms of inliers/outliers' $S_{gtp}(\mathbf{x})$. The class used as inliers is in brackets.

for loss reduction from $-\nabla_{\boldsymbol{\theta}}\mathcal{L}(X)$, we introduce the concept *de facto* update, which is computed by projecting $\nabla_{\boldsymbol{\theta}}\mathcal{L}(X)$ onto the direction of $\nabla_{\boldsymbol{\theta}}\mathcal{L}(\mathbf{x}_i)$: $d_i = \nabla_{\boldsymbol{\theta}}\mathcal{L}(X) \cdot \frac{\nabla_{\boldsymbol{\theta}}\mathcal{L}(\mathbf{x}_i)}{||\nabla_{\boldsymbol{\theta}}\mathcal{L}(\mathbf{x}_i)||}$. For example, as shown in Fig. 4a, the *de facto* update d_1 and d_2 reflect how much effort the network will devote to reduce the training loss of $\mathbf{x}_1$ and $\mathbf{x}_2$ respectively. *De facto* update can be viewed as an even more direct indicator of data's priority during training. In our case, we still take the gradients w.r.t. the weights between SSD's penultimate and softmax layer as an example. Under the setup in Sec. 4.1, we calculate the average *de facto* update for inliers and outliers respectively, and visualize typical results of *de facto* update on several image benchmarks in Fig. 4b-4d: As can be seen from the results, despite being close at the beginning, the average *de facto* update of inliers becomes evidently higher than outliers as the training continues, which justifies that SSD will bias towards inliers' best updating directions.

3.3.4 Baseline Outlier Score and Additional Remarks

Having illustrated inlier priority both theoretically and empirically, it can be expected that inliers are likely to achieve better training performance than outliers on a SSD after the self-supervised learning. In other words, SSD will prioritize reducing inliers' loss, which suggests that it is possible to discriminate outliers directly by each datum's loss value after training. To be more specific, for an image $\mathbf{x}^{(y)}$, we note that the calculation of its cross entropy loss only depends on its ground truth class probability $P^{(y)}(\mathbf{x}^{(y)}|\boldsymbol{\theta})$ that corresponds to its pseudo class label y. Thus, we propose Ground Truth Probability (GTP) score $S_{gtp}(\mathbf{x})$ that averages $P^{(y)}(\mathbf{x}^{(y)}|\boldsymbol{\theta})$ for all K operations to measure outlierness:

$$S_{gtp}(\mathbf{x}) = \frac{1}{K}\sum_{y=1}^{K} \mathbf{1}_y^\top \cdot \mathbf{P}(\mathbf{x}^{(y)}|\boldsymbol{\theta}) = \frac{1}{K}\sum_{y=1}^{K} P^{(y)}(\mathbf{x}^{(y)}|\boldsymbol{\theta}) \quad (8)$$

where $\mathbf{1}_y$ denotes the one-hot vector with the y-th element to be 1. To validate whether GTP score is a plausible way to measure outlierness, we calculate the $S_{gtp}(\mathbf{x})$ on image benchmarks and visualize the accumulated histograms for inliers and outliers respectively (note that histograms are normalized for better visualization). Representative results are shown in Fig. 5a-5d, and the score distributions of inliers and outliers are observed to be readily separable. Thus, GTP score can be a feasible baseline score for end-to-end OD. In addition, we would also like to point out the relation between inlier priority and representation learning: In deep OD task like outlier image removal, the difference between outliers and inliers lie in their semantics, e.g. high-level structure and appearance. To encourage the semantic similarity within inliers and maximize the semantic difference between inliers and outliers, it is necessary to learn good representations with rich semantics in the first place. Thus, a learning task that can yield semantically meaningful representations is the foundation for inliers to be semantically similar and joint their efforts into a priority against outliers.

3.4 Network Uncertainty As an Outlierness Measure

3.4.1 Motivation

SSD+GTP score provides a baseline end-to-end OD solution. However, it is imperfect and still has room for improvement, especially considering that the proposed self-supervised learning is not as precise as the classic supervised learning with human annotations: The data operation sometimes may not be able to transform the original image into an actual new one, e.g. a digit "8" is still itself after flipping is performed. Therefore, labels assigned to pseudo classes can be inaccurate. Since the calculation of GTP score in (8) relies on the pseudo class label y, such inaccurate labeling may undermine the GTP score's effectiveness to discriminate outliers. Motivated by this problem, we intend to design a new outlierness measure that is independent of pseudo class labels, so as to exploit the possibility to further improve end-to-end OD performance. Besides, when compared with other outlierness measures like density or proximity, uncertainty is usually directly optimized during the training of DNN, while other measures are not an explicit goal of the optimization. Therefore, we believe that network uncertainty can be a more direct indicator of inlier priority than other traditional measures. To this end, network uncertainty comes into our sight, since it is exactly an orthogonal attribute to DNN's classification accuracy [74]. As previous works basically discuss this concept in the context of DNN's prediction confidence, it is interesting to explore whether network uncertainty can be used for end-to-end OD.

3.4.2 A Demonstration Experiment

We carry out a simple demonstration experiment to shed light on this issue. For visualization, we generate 2D data with different degree of outlierness (detailed in Sec. 3 in supplementary material): The generated data (dots in Fig. 6) exhibit a larger dispersion as their coordinate on x-axis, x_i, gets more distant from the origin of x-axis, which enables data on two ends to show larger outlierness. To calculate

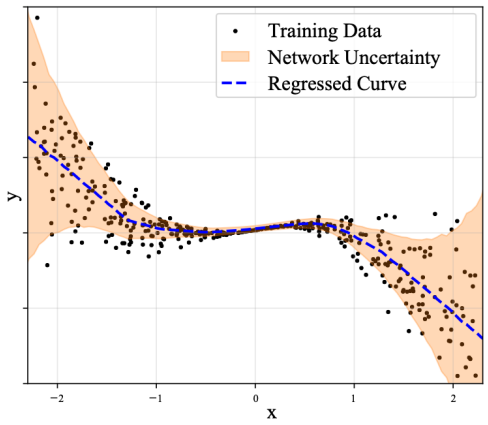


Fig. 6: The uncertainty of a regression network.

network uncertainty, we introduce a regression task that predicts y-axis coordinate y_i by corresponding x_i. Note that the regression task can be viewed as a self-supervised learning task, since we actually intend to infer the missing coordinate y_i by the incomplete data $\tilde{\mathbf{x}}_i = [x_i]$ like the masking mechanism [58]. The regression task is performed by training a simple neural network with the generated 2D data, and we estimate the uncertainty of neural network by the popular MC-Dropout method [64]. As it is shown in Fig. 6, it is easy to discover that the network uncertainty (highlighted orange region) is positively correlated to the outlierness of data. In other words, the experiment demonstrates some interesting connections among network uncertainty, OD and self-supervised learning: *The uncertainty of a neural network, which is trained to accomplish a self-supervised learning task (not OD itself), actually serves as a fairly effective way to measure data's outlierness.* Besides, it is also worth noting that network uncertainty is not relevant to the label y_i. This facilitates it to be more robust to label noises in self-supervised learning, just as we discussed in Sec. 3.4.1.

3.4.3 Network Uncertainty based Outlier Scores

As reviewed in Sec. 2.3, the uncertainty of DNN can be estimated by several ways, which can be categorized into Bayesian methods and non-Bayesian methods. Since Bayesian methods are usually more complicated and require more modifications to DNN itself, we focus on non-Bayesian methods when designing outlier scores. The following network uncertainty based scores are designed: **(1)** Maximum Probability (MP) score $S_{mp}(\mathbf{x})$. $S_{mp}(\mathbf{x})$ utilizes the maximum probability (i.e. prediction probability) output by the Softmax layer of SSD, which has proved to be a simple but strong baseline for uncertainty estimation [66], [67]:

$$S_{mp}(\mathbf{x}) = \frac{1}{K}\sum_{y=1}^{K} \max \mathbf{P}(\mathbf{x}^{(y)}|\boldsymbol{\theta}) = \frac{1}{K}\sum_{y=1}^{K} \max_t P^{(t)}(\mathbf{x}^{(y)}|\boldsymbol{\theta}) \quad (9)$$

(2) MC-Dropout (MCD) score $S_{mcd}(\mathbf{x})$. MC-Dropout keeps the dropout layers functional during inference, and calculates the first and second-order moment of DNN's outputs by several forward passes [64]. Since the maximum output probability and variance in DNN's outputs are both able to reflect DNN's uncertainty, we devise $S_{mcd}(\mathbf{x})$ as follows, so as to adapt it to OD task ($Mean(\cdot)$ and $Var(\cdot)$ refers to the mean and variance of multiple forward passes):

$$S_{mcd}(\mathbf{x}) = \frac{1}{K}\sum_{y=1}^{K} -Var(\max \mathbf{P}(\mathbf{x}^{(y)}|\boldsymbol{\theta})) + Mean(\max \mathbf{P}(\mathbf{x}^{(y)}|\boldsymbol{\theta})) \quad (10)$$

(3) Negative Entropy (NE) based score $S_{ne}(\mathbf{x})$. Information entropy (i.e. Shannon entropy) has constantly been used for measuring information and uncertainty embedded in data. Thus, we design $S_{ne}(\mathbf{x})$ to be computing the negative entropy of SSD's output probability distribution $\mathbf{P}(\mathbf{x}^{(y)}|\boldsymbol{\theta})$:

$$S_{ne}(\mathbf{x}) = \frac{1}{K}\sum_{y=1}^{K}\sum_{t=1}^{K} P^{(t)}(\mathbf{x}^{(y)}|\boldsymbol{\theta}) \log(P^{(t)}(\mathbf{x}^{(y)}|\boldsymbol{\theta})) \quad (11)$$

In addition to scores above, other network uncertainty based scores can also be explored. Our later evaluations show that network uncertainty based scores typically work better than the baseline outlier score S_{gtp}.

3.5 Score Refinement of Discriminative *E*^{3}*Outlier*

3.5.1 Motivation

Although components presented above have constituted a fully-functional end-to-end OD solution, it is still possible to improve discriminative *E*^{3}*Outlier*'s performance. As we have demonstrated how inlier priority and network uncertainty enable end-to-end OD, they should also be considered as the origin for performance improvement. Intuitively, a better OD performance essentially suggests that the priority of inliers is magnified, while it can also be accomplished by better uncertainty estimation. Inspired by such instincts, we propose two types of strategies to refine outlier scores.

3.5.2 Re-weighting Strategy

Our first instinct is to make SSD further prioritize inliers during training. Nevertheless, it is noted that inliers and outliers are indiscriminately fed into SSD at the very beginning of training, i.e. inliers and outliers are equally weighted by 1. Having revealed the role of inlier priority in OD, it is undoubted that this default initialization is not optimal: We can assign inliers with larger weights right before the beginning of SSD's training, which justifies the introduction of a re-weighting scheme. Since given data are completely unlabeled in OD, how and when to re-weight those unlabeled data for OD are key issues that we have to answer. As to how to re-weight, our solution is to utilize scores yielded by the proposed outlierness measure as weights, which have already achieved far better OD performance than existing methods. To be more specific, we can normalize scores into non-negative weights $w_1, \cdots w_N$ that satisfy $\sum_{i=1}^{N} w_i = 1$, and modify the objective function in (1) into the form below:

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} w_i \mathcal{L}_{SS}(\mathbf{x}_i|\boldsymbol{\theta}) \quad (12)$$

As for when to re-weight, since scores are only accessible after self-supervised learning begins, we can perform re-weighting during or after SSD's training. Accordingly, we propose *online* re-weighting and *reboot* re-weighting strategy: Online re-weighting strategy will update the weights

at the end of every epoch, and only one SSD is trained. By contrast, reboot re-weighting trains two SSD models: The first SSD is trained by a standard procedure, while the scores yielded by the first SSD are used as fixed weights to train the second SSD. The full algorithms are detailed in Algorithm 1 and Algorithm 2 in Sec. 4 of supplementary material. Our evaluations show that both algorithms can improve E^3*Outlier*'s performance.

3.5.3 Ensemble Strategy

In addition to the re-weighting strategy, another instinct is to improve uncertainty estimation for better OD performance. Since a generic strategy that can be easily embedded into the model is always preferred, we introduce the ensemble strategy into the score refinement stage. Ensemble is a widely-used technique in machine learning that combines multiple models into a stronger one. It is shown to be a powerful tool to improve the predictive performance [75], and recent works also demonstrate that an ensemble of DNNs can be highly efficient for producing good model uncertainty estimates [65], [67]. Specifically, we first create multiple SSD models $M_1, \cdots, M_e$ in a certain way, where $e > 1$ is the number of SSD models. For example, we can initialize SSD models with different random seeds, or adopt several different network architectures as different SSD models. After self-supervised learning, we simply average the outputs of different SSD models by $\bar{\mathbf{P}}(\mathbf{x}_i^{(y)}|\boldsymbol{\theta}) = \frac{1}{e}\sum_{j=1}^{e}\mathbf{P}_j(\mathbf{x}_i^{(y)}|\boldsymbol{\theta})$, where $\mathbf{P}_j(\mathbf{x}_i^{(y)}|\boldsymbol{\theta})$ is the outputs of j_{th} SSD model. Afterwards, we can calculate any network uncertainty based score with $\bar{\mathbf{P}}(\mathbf{x}_i^{(y)}|\boldsymbol{\theta})$. Note that the ensemble process can be readily paralleled for potential acceleration. Our later empirical evaluations show that such simple ensemble technique almost consistently improves the OD performance when compared with the case where a single SSD model is used.

3.5.4 Joint Score Refinement

Two aforementioned strategies are both able to yield better outlier scores, but it should be noted that they actually refine outlier scores from different views: The re-weighting strategy strengthens the inlier priority during self-supervised learning, while the ensemble strategy aims to improve the estimation of network uncertainty. In other words, two strategies exploit non-overlapping facets for score refinement. Thus, using a joint strategy of the re-weighting and ensemble to achieve even better OD performance is natural. In this paper, we devise the final score refinement stage by combining the reboot re-weighting strategy with the ensemble strategy (shown in Algorithm 3 in Sec. 4 of the supplementary material). Note that this is not the only form to combine re-weighting and ensemble, e.g. combining online re-weighting with the ensemble is also possible.

3.6 Other Learning Paradigms for E^3*Outlier*

In previous sections, we have demonstrated the way to leverage discriminative self-supervised learning to perform deep OD. As the way to introduce self-supervision is not limited to the discriminative learning paradigm, it is natural for us to explore other learning paradigms for E^3*Outlier*, which brings two benefits: First, more available learning paradigms enable E^3*Outlier* to be more flexible when dealing with different application scenarios. Second, emerging self-supervised learning paradigms like contrastive learning also facilitate E^3*Outlier* to further exploit its potential for deep OD. Thus, this section will detail our solution to apply generative and contrastive learning paradigms to E^3*Outlier*.

3.6.1 Generative E^3 Outlier

Generative learning paradigm is not new, because AE based reconstruction is exactly the most frequently-used method in existing deep OD solutions so far. However, as illustrated in Sec. 3.2.3, existing generative solutions often perform unsatisfactorily. As self-supervision is shown to be surprisingly effective in discriminative E^3*Outlier*, it is instinctive for us to explore *whether self-supervision can also improve the performance of generative deep OD*. Specifically, our solution is to add richer self-supervision information into the generation process to avoid simple reconstruction of the inputs. Inspired by the fact that data operations can provide rich self-supervision signal in SSD, we propose the generative self-supervised learning (GSS) paradigm below: Consider a data operation set with K_g operations $\mathcal{O}_g = \{O_g(\cdot|y)\}_{y=1}^{K_g}$. The data operations in $\mathcal{O}_g$ can be defined by various ways, such as certain transformations or fetching a specific part or modality of the input data. Then, we draw two different operations $O_g(\cdot|y_1)$ and $O_g(\cdot|y_2)$ from $\mathcal{O}_g$. Given an input data $\mathbf{x}$, two operations are required to satisfy:

$$O_g(\mathbf{x}|y_1) \neq O_g(\mathbf{x}|y_2), \qquad y_1 \neq y_2 \tag{13}$$

Then, a generative DNN $\mathcal{G}$ (e.g. AE, UNet [76] or GANs) is trained to generate $O_g(\mathbf{x}|y_2)$ by taking $O_g(\mathbf{x}|y_1)$ as the input, which is equivalent to minimizing the objective below:

$$\mathcal{L}_{GSS}(y_1, y_2) = \frac{1}{N}\sum_{i=1}^{N}||\mathcal{G}(O_g(\mathbf{x}_i|y_1)) - O_g(\mathbf{x}_i|y_2)||_2^2 \tag{14}$$

It is easy to note that when Eq. (13) is not satisfied, Eq. (14) will degrade into plain reconstruction. When $\mathcal{G}$ has been trained, one can simply obtain an outlier score of $\mathbf{x}$ based on the MSE loss of generation:

$$S_g(\mathbf{x}|y_1, y_2) = -||\mathcal{G}(O_g(\mathbf{x}|y_1)) - O_g(\mathbf{x}|y_2)||_2^2 \tag{15}$$

Since there exist different ways to select operations, it is natural to train the model and compute final outlier score by a combination of different y_1, y_2 configurations:

$$\begin{aligned} \mathcal{L}_{GSS} &= \sum_{y_1}\sum_{y_2}\mathcal{L}_{GSS}(y_1, y_2), \\ S_g(\mathbf{x}) &= \sum_{y_1}\sum_{y_2}S_g(\mathbf{x}|y_1, y_2) \end{aligned} \tag{16}$$

Compared with the plain reconstruction adopted by AE based deep OD methods, the key to our generative E^3*Outlier* is to make DNN generate a different datum obtained by a non-identical operation, which makes the learning task more challenging for DNNs. This not only avoids the DNN to simply memorize the low-level details, but also encourages the DNN to consider high-level semantics by learning the correlations of two different data, which can be

viewed as valuable self-supervision information. Our later evaluations show that generative E^3*Outlier* can produce tangible performance improvement when it shares the same generative DNN with other reconstruction based deep OD solutions. More importantly, generative E^3*Outlier* can be readily applied to some important scenarios where the input data can be decomposed into multiple views or modalities. For example, video data are usually considered from the view of both appearance and motion. In those cases, the correspondence between different data views/modalities is valuable self-supervision signal in itself, and generative E^3*Outlier* provides a convenient and straightforward way to exploit such semantics. As a demonstration, we will show how to design a new unsupervised video abnormal event detection solution by generative E^3*Outlier* in Sec. 4.3.2.

3.6.2 Contrastive E^3 Outlier

It is easy to notice that the performance of current deep OD solutions, including the proposed discriminative E^3*Outlier*, suffers from evidently inferior performance on colored image datasets (e.g. CIFAR10) when compared with comparatively simple gray-scale image datasets (e.g. MNIST). Meanwhile, we also note that color based operations (e.g. color jittering and RGB-to-gray transformation) play an important role in many vision tasks. To further exploit color information and enhance the capability to handle more ubiquitous colored images in practical applications, we leverage the emerging contrastive learning paradigm, which is shown to be highly effective in unsupervised representation learning of real-world colored images, to provide self-supervision in deep OD and design contrastive E^3*Outlier*. The core idea of contrastive learning is to learn meaningful representations by making DNNs compare a pair of data drawn from the unlabeled dataset. We choose one of the most representative contrastive learning method, SimCLR [77], as the foundation for the proposed contrastive E^3*Outlier*. Specifically, a contrastive loss for a datum $\mathbf{x}$ is defined as follows:

$$\mathcal{L}_{cl}(\mathbf{x}, X^+, X^-) = \\ -\frac{1}{|X^+|}\log\frac{\sum_{\mathbf{x}'\in X^+}\exp(sim(z(\mathbf{x}), z(\mathbf{x}'))/\tau)}{\sum_{\mathbf{x}'\in X^+\cup X^-}\exp(sim(z(\mathbf{x}), z(\mathbf{x}'))/\tau)} \tag{17}$$

where X^+/X^- denote the set with data that can form a positive/negative pair with $\mathbf{x}$, and $sim(\cdot,\cdot)$ is a similarity measure like cosine similarity. $|\cdot|$ is the cardinality of the set, and $z(\mathbf{x})$ is the projection yielded by feeding DNN's learned representation $f(\mathbf{x})$ into a projection layer $g(\cdot)$: $z(\mathbf{x}) = g(f(\mathbf{x}))$. τ is a hyperparameter. Next, the issue is to construct positive and negative data pairs to enable the calculation of Eq. (17). To this end, we introduce a random augmentation set $\mathcal{A}$, which contains augmentation operations that is composed of color jittering, RGB-to-gray transformation and image crop with random parameterization. Each time two independent random augmentation A_1 and A_2 are drawn from $\mathcal{A}$. After that, the data pair of augmented data $A_1(\mathbf{x})$ and $A_2(\mathbf{x})$ are viewed as a positive pair, while any other pair is viewed as negative. The goal of contrastive loss defined in Eq. (17) is to yield similar representations for a positive data pair, and make representations of a negative pair dissimilar. Given a mini-batch data set B drawn from the unlabeled dataset, SimCLR defined the following training objective to perform contrastive learning:

$$\mathcal{L}_{scl}(B, A_1, A_2) = \frac{1}{2|B|}\sum_{i=1}^{|B|}(\mathcal{L}_{cl}(A_1(\mathbf{x}_i), \{A_2(\mathbf{x}_i)\}, \hat{B}_{-i}) + \\ \mathcal{L}_{cl}(A_2(\mathbf{x}_i), \{A_1(\mathbf{x}_i)\}, \hat{B}_{-i})) \tag{18}$$

where we define $\hat{B}_{-i} = \{A_1(\mathbf{x}_j)\}_{j\neq i} \cup \{A_2(\mathbf{x}_j)\}_{j\neq i}$. Some recent works [77], [78] point out that some data operations (e.g. 90 degree rotation) can be used to generate negative pairs as they produce very different data from the original one. This is also verified in discriminative E^3*Outlier*, since those data operations are often likely to produce pseudo classes that are readily separable. Following such an observation, we collect an operation set $\mathcal{O}_c = \{O_c(\cdot|y)\}_{y=1}^{K_c}$ with K_c operations (including one identity transformation), and expand the mini-batch B into $B' = O_c(B|1) \cup \cdots \cup O_c(B|K_c)$, where the data set $O_c(B|y) = \{O_c(\mathbf{x}|y)|\mathbf{x}\in B\}$. Since B' can be viewed as a data set with K_c pseudo classes and discriminative E^3*Outlier* works well in deep OD, we substitute B by B' into Eq. (18) for training, and make DNN learn to classify those pseudo classes by an additional discriminative module and the cross-entropy loss $\mathcal{L}_{cls}(B')$, so as to produce more meaningful representations. In this way, the contrastive self-supervised learning (CSS) of E^3*Outlier* can be performed by the joint loss below:

$$\mathcal{L}_{CSS} = \mathcal{L}_{scl}(B', A_1, A_2) + \mathcal{L}_{cls}(B') \tag{19}$$

After training, we design a simple but effective outlier score based on inner product of learned representations: For the datum $\mathbf{x}_i^{(y)} = O_c(\mathbf{x}_i|y)$ obtained by imposing the y-th operation in $\mathcal{O}_c$ on $\mathbf{x}_i$, its outlier score $S_c(\mathbf{x}_i^{(y)})$ is given by:

$$S_c(\mathbf{x}_i^{(y)}) = \frac{1}{Z_{scl}^{(y)}}\max_{j\neq i} f^\top(\mathbf{x}_i^{(y)})\cdot f(\mathbf{x}_j^{(y)}) \tag{20}$$

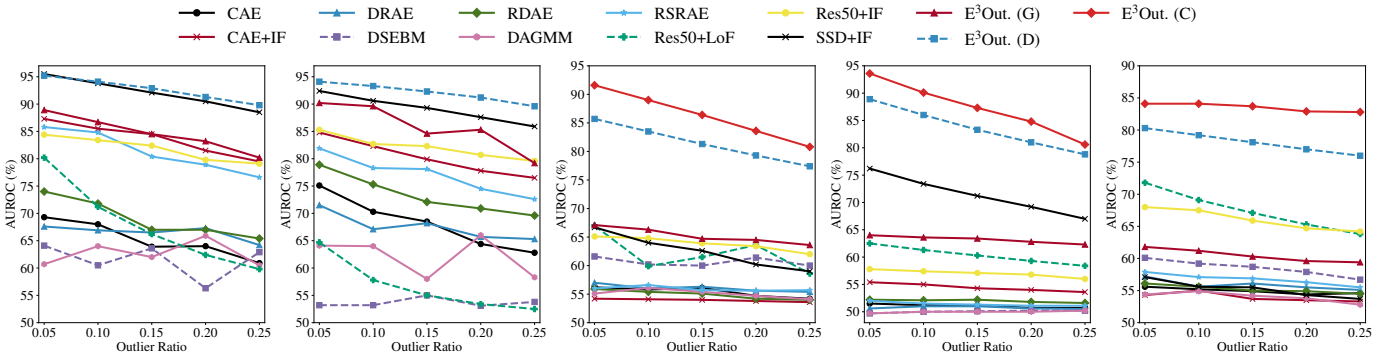
where $Z_{scl}^{(y)}$ is the normalization term computed as follows:

$$Z_{scl}^{(y)} = (\frac{1}{N}\sum_{i=1}^{N}||f(\mathbf{x}_i^{(y)})||)^{-1} \tag{21}$$

In Eq. (20), the score actually computes the maximum inner product between the learned representations of $\mathbf{x}_i^{(y)}$ and other data yielded by operation $O(\cdot|y)$, so as to measure how similar $\mathbf{x}_i^{(y)}$ is to the rest of data. With multiple operations in $\mathcal{O}_c$, the final outlier score can be computed by:

$$S_c(\mathbf{x}_i) = \sum_{y=1}^{K_c} S_c(\mathbf{x}_i^{(y)}) \tag{22}$$

Just like that contrastive learning paradigm significantly improves the performance of self-supervised learning, our later empirical evaluations show that contrastive E^3*Outlier* also advances the deep OD performance by a notable margin on those colored datasets that are relatively difficult for previous generative and discriminative E^3*Outlier*. As a summary, by designing generative learning and contrastive learning based solutions, we enable E^3*Outlier* to be a more flexible and stronger deep OD framework.



(a) MNIST (b) Fashion-MNIST (c) CIFAR10 (d) SVHN (e) CIFAR100

Fig. 7: AUROC comparison of OD methods under different outlier ratios.

4 Experiments

4.1 Experimental Settings

4.1.1 Benchmark Datasets and Evaluation

To validate the effectiveness of the proposed framework, we conduct extensive experiments on five frequently-used public image benchmarks: MNIST (MST) [79], Fashion-MNIST (FMST) [80], CIFAR10 (C10) [81], SVHN (SH) [82], CIFAR100 (C100) [81]. We follow the standard procedure, which is shared by previous image outlier removal works like [8], [9], [46], to construct a noisy image set with outliers: Given a standard image benchmark, all images from a class with one common semantic concept (e.g. "horse", "bag") are retrieved as inliers, while outliers are randomly sampled from the rest of classes by an outlier ratio ρ. We vary ρ from 5% to 25% by a step of 5%. The assigned inlier/outlier labels are strictly unknown to OD methods and only used for evaluation. Each class of a benchmark is used as inliers in turn, and the performance on all classes is averaged as the overall OD performance on this benchmark dataset. Since all images are viewed as unlabeled in OD, we do not use the split of train/test set and merge them for experiments. Note that for CIFAR100 dataset, we uses 20 superclasses instead of the original 100 classes to ensure that the constructed noisy image set contains sufficient data for DNN's training, and it can also test the OD performance when inliers have multiple subclasses (each superclass in CIFAR100 contains 5 classes). All experiments are repeated for 5 times with different random seeds, so as to yield the average results. Raw pixels are directly used as inputs with their intensity normalized into $[-1, 1]$. As for evaluation, we adopt the commonly-used Area under the Receiver Operating Characteristic curve (AUROC) and Area under the Precision-Recall curve (AUPR) as threshold-independent metrics [83].

4.1.2 Compared Methods

We extensively compare generative *E^3Outlier* (E^3Out. (G)), discriminative E^3Outlier (*E^3Out.* (D)) and contrastive *E^3Outlier* (E^3Out. (C)) with baselines and existing state-of-the-art DNN based OD methods in literature: **(1)** Convolutional Auto-Encoder (CAE) [84]. CAE is the most prevalent DNN type to deal with image data in many unsupervised learning tasks. Here it serves as an end-to-end baseline, which directly uses CAE's reconstruction loss to perform deep outlier removal. **(2)** CAE+Isolation Forest (CAE+IF). IF [40] is a classic OD method with wide popularity, so we combine it with CAE as the baseline of two-stage OD approaches. Specifically, CAE+IF feeds CAE's learned representations from its intermediate hidden layer into IF to perform OD. **(3)** SSD+IF. It shares *E^3Outlier*'s SSD part but feeds SSD's learned representations into an IF model to perform OD. SSD+IF serves as a two-stage baseline to compare against the proposed end-to-end *E^3Outlier*. **(4)** Discriminative Reconstruction based Auto-Encoder (DRAE) [9]. DRAE discriminates outliers by thresholding CAE's reconstruction loss with a self-adaptive scheme, which is in turn integrated into the loss function to refine the outlier removal performance. **(5)** Deep Structured Energy based Models (DSEBM) [45]. DSEBM uses an energy based function and score matching technique to estimate the probability that a datum fits the data distribution. **(6)** Robust Deep Auto-Encoder (RDAE) [46]. RDAE synthesizes CAE and RPCA, and it iteratively decomposes unlabeled data into a low-rank part and a sparse error part for outlier removal. **(7)** Deep Auto-encoding Gaussian Mixture Model (DAGMM) [48]. DAGMM embeds a GMM parameter estimation network into CAE, which realizes end-to-end OD by performing representation learning and fitting a GMM simultaneously. **(8)** Multiple-Objective Generative Adversarial Active Learning (MOGAAL) [50]. MOGAAL attempts to generate pseudo outliers that are distributed around given unlabeled data with modified GANs and active learning, so as to transform OD into a supervised binary classification problem. **(9)** Robust Subspace Recovery based AE (RSRAE) [52]. RSRAE is the latest method that improves OD performance by learning to recover the underlying data manifold in a subspace while performing AE's reconstruction. For RSRAE, the reconstruction loss and RSR loss are optimized in a seprated manner. In addition to deep solutions, we also include the following baseline solutions for a more comprehensive comparison: **(10)** Two-stage solutions based on pre-trained DNN and the classic OD model. DNN models pre-trained on large-scale generic datasets prove to be an effective tool for feature extraction. Thus, to design a two-stage solution, we use a ResNet50 model pre-trained on ImageNet dataset as feature extractor, and the extracted features are then fed into a classic OD model. IF and the classic Local Outlier Factor (LoF) are exploited here. Due to

TABLE 1: OD performance comparison (in %) in terms of AUROC (Area Under ROC curve, shorted as ROC), AUPR-In (Area under PR curve with inliers to be the positive class, shorted as PR-I) and AUPR-Out (Area under PR curve with outliers to be the positive class, shorted as PR-O). Each benchmark shows the case where $\rho = 10\%$ and $\rho = 20\%$. Note that contrastive *E^3Outlier* is only used for benchmark datasets with colored images (CIFAR10/SVHN/CIFAR100), and the raw performance without score refinement is compared for fairness. The best performer is shown in bold font.

Dataset	MNIST			Fashion-MNIST			CIFAR10			SVHN			CIFAR100		
	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O
$\rho = 10\%$															
CAE	68.0	92.0	32.9	70.3	94.3	29.3	55.8	91.0	14.4	51.2	90.3	10.6	55.2	91.0	14.5
CAE+IF	85.5	97.8	49.0	82.3	97.2	40.3	54.1	90.2	13.7	55.0	91.4	11.9	55.0	90.7	13.8
DRAE	66.9	93.0	30.5	67.1	93.9	25.5	56.0	90.7	14.7	51.0	90.3	10.5	55.6	90.9	15.0
DSEBM	60.5	91.6	23.0	53.2	88.9	19.7	60.2	92.3	14.7	50.0	90.0	10.1	59.2	92.2	16.2
RDAE	71.8	93.1	35.8	75.3	95.8	31.7	55.4	90.7	14.9	52.1	90.6	10.8	55.6	90.9	15.0
DAGMM	64.0	92.9	26.6	64.0	92.7	30.3	56.1	91.3	15.6	50.0	90.0	19.3	54.9	91.1	14.2
MOGAAL	30.9	78.8	15.2	22.8	74.8	14.8	56.2	91.1	13.6	49.0	89.7	9.8	53.2	90.4	12.6
RSRAE	84.8	97.4	45.4	78.3	96.2	37.0	56.6	91.4	14.0	51.5	90.3	10.6	57.1	91.6	14.1
Res50+LoF	71.2	97.5	26.6	57.8	96.2	16.9	59.9	91.4	17.4	61.3	90.3	14.0	69.1	94.6	22.2
Res50+IF	83.4	97.5	43.3	82.7	97.3	43.8	64.8	93.8	17.9	57.4	92.0	12.8	67.5	94.3	21.0
SSD+IF	93.8	99.2	**68.7**	90.6	98.5	68.6	64.0	93.5	18.3	73.4	95.9	22.0	55.6	91.5	13.0
E^3Out. (G)	86.7	96.4	60.3	89.6	98.5	61.6	66.3	93.5	20.0	63.6	93.9	15.0	61.2	92.4	16.7
E^3Out. (D)	**94.1**	**99.3**	67.5	**93.3**	**99.0**	**75.9**	83.5	97.5	43.4	86.0	98.0	36.7	79.2	96.8	33.3
E^3Out. (C)	-	-	-	-	-	-	**89.0**	**98.5**	**53.2**	**90.1**	**98.5**	**51.3**	**84.1**	**97.8**	**38.0**
$\rho = 20\%$															
CAE	64.0	82.7	40.7	64.4	85.3	36.8	54.7	81.6	25.5	50.7	80.2	20.7	54.4	81.7	25.6
CAE+IF	81.5	93.6	57.2	77.8	92.2	49.0	53.8	80.7	25.3	54.0	82.0	22.4	53.5	80.9	25.1
DRAE	67.3	86.6	42.5	65.7	86.9	36.6	55.6	81.7	26.8	50.6	80.4	20.5	55.5	81.8	27.0
DSEBM	56.3	81.2	32.3	53.1	79.6	31.7	61.4	85.2	27.8	50.2	80.3	20.2	57.9	83.7	27.8
RDAE	67.0	89.2	43.2	70.9	89.2	41.4	54.2	81.0	25.7	51.8	80.9	21.1	54.9	81.5	26.5
DAGMM	65.9	86.7	41.3	66.0	86.7	43.5	54.7	81.8	26.3	50.0	79.9	29.6	53.8	81.5	24.7
MOGAAL	37.8	70.6	28.0	34.0	66.6	28.3	55.7	82.0	25.0	49.6	79.8	19.8	53.1	80.9	24.4
RSRAE	78.9	91.3	53.0	74.5	90.4	46.3	55.6	82.1	25.8	51.1	80.3	21.0	56.3	82.7	25.2
Res50+LoF	62.4	84.9	31.0	53.4	80.3	24.9	63.6	84.9	27.9	59.3	85.0	25.2	65.3	87.5	32.6
Res50+IF	79.8	93.6	52.1	80.7	93.5	55.0	63.4	86.6	30.4	56.8	83.3	24.2	64.7	87.1	32.4
SSD+IF	90.5	97.3	71.0	87.6	95.6	71.4	60.2	85.0	28.3	69.2	89.5	33.7	54.3	82.1	23.4
E^3Out. (G)	83.2	90.4	67.9	85.3	95.2	66.4	64.5	85.7	33.0	62.8	86.8	27.9	59.6	83.8	28.6
E^3Out. (D)	**91.3**	**97.6**	**72.3**	**91.2**	**97.1**	**78.9**	79.3	93.1	52.7	81.0	93.4	47.0	77.0	92.4	46.5
E^3Out. (C)	-	-	-	-	-	-	**83.6**	**94.8**	**59.0**	**84.8**	**94.9**	**57.6**	**82.9**	**95.1**	**53.0**

page limit, implementation details are provided in Sec. 5 of the supplementary material. All of our codes and results can be verified at https://github.com/demonzyj56/E3Outlier.

4.2 Experimental Results

4.2.1 Raw OD Performance Comparison

Due to the space limit, we report numerical results under $\rho = 10\%$ and 20% in Table 1, while the AUROC comparison under different outlier ratios are shown in Fig. 7. From those results, we can obtain the following observations: **(1)** First of all, the proposed *E^3Outlier* framework possesses an evident advantage against existing state-of-the-art DNN based OD methods and baselines in terms of all evaluation metrics. Taking discriminative *E^3Outlier* as an example, it outperforms the best performer among state-of-the-art DNN based OD methods and baselines by a considerable 8%-20% AUROC on different benchmark datasets. In particular, it has realized a performance leap on CIFAR10, SVHN and CIFAR100, which are generally acknowledged to be challenging benchmarks for unsupervised learning tasks like deep outlier removal or clustering. Meanwhile, with the same CAE as backbone, the proposed generative *E^3Outlier* is able to achieve evidently superior performance to existing CAE based deep OD solutions. Specifically, although it is inferior to its discriminative and contrastive counterparts, generative *E^3Outlier* consistently outperforms all AE based deep OD solutions in terms of AUROC, while it also yields comparable or better AUPR-In and AUPR-Out performance. Such improvement further justifies the effectiveness of introducing richer self-supervision information, and in later sections we show that generative *E^3Outlier* also enables us to flexibly handle other deep OD applications. Next, the proposed contrastive *E^3Outlier* is able to produce a significant performance gain (about 4%-6% AUROC) on colored datasets (CIFAR10/SVHN/CIFAR100) that are relatively difficult for its discriminative and generative counterparts, and it suggests that the potential of *E^3Outlier* can be further exploited by introducing more advanced self-supervised learning paradigms. Thus, the above observations have justified *E^3Outlier* as a highly effective framework for DNN based OD. **(2)** Second, we notice that the baseline OD solutions that combine the classic OD model and features extracted from pre-trained ResNet50 model (Res50+LoF and Res50+IF) can indeed produce better performance than previous end-to-end OD solutions in many cases, which verifies the importance of the good representation. However, there is still a large performance gap between such two-stage solutions and the proposed deep OD framework, especially discriminative and contrastive *E^3Outlier*. Thus, it further

TABLE 2: Performance of discriminative *E*^{3}*Outlier* (in %) before and after joint score refinement (JSR) in terms of Area Under ROC curve, PR curve with inliers to be the positive class (PR-I) and PR curve with outliers to be the positive class (PR-O). Each benchmark shows the case where $\rho = 10\%$ and $\rho = 20\%$ due to the space limit.

Dataset	MNIST			Fashion-MNIST			CIFAR10			SVHN			CIFAR100		
	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O
$\rho = 10\%$															
E^{3}Out.	94.1	99.3	67.5	93.3	**99.0**	75.9	83.5	97.5	43.4	86.0	98.0	36.7	79.2	96.8	33.3
E^{3}Out.+JSR	**94.9**	**99.4**	**71.0**	**93.5**	**99.0**	**77.2**	**84.7**	**97.7**	**45.7**	**87.1**	**98.2**	**37.7**	**81.3**	**97.2**	**37.0**
$\rho = 20\%$															
E^{3}Out.	91.3	97.6	72.3	91.2	97.1	78.9	79.3	93.1	52.7	81.0	93.4	47.0	77.0	92.4	46.5
E^{3}Out.+JSR	**92.9**	**98.1**	**76.3**	**92.1**	**97.4**	**81.9**	**80.3**	**93.5**	**54.5**	**82.0**	**94.2**	**47.9**	**79.1**	**93.1**	**49.9**

demonstrates the effectiveness of the proposed deep OD framework. **(3)** Third, it is interesting to note that two-stage OD approaches can be more effective than previous end-to-end OD approaches. Specifically, the two-stage counterpart of discriminative *E*^{3}*Outlier* SSD+IF achieves fairly close performance to discriminative *E*^{3}*Outlier* on relatively simple gray-scale image datasets (MNIST/Fashion-MNIST). Meanwhile, CAE based end-to-end OD solutions (DRAE/DSEBM/DAGMM/RSRAE) cannot constantly outperform their two-stage counterparts (CAE+IF/RDAE), and CAE+IF even performs much better than some CAE based end-to-end solutions on MNIST/Fashion-MNIST. Nevertheless, as shown in Fig. 7a-Fig. 7e, the proposed discriminative *E*^{3}*Outlier* almost defeats its two-stage baseline SSD+IF in all experiments, and it suffers from evidently worse performance (i.e. over 10% AUROC loss) on difficult datasets like CIFAR10/SVHN/CIFAR100. **(4)** Among existing end-to-end OD methods, we notice that although recent end-to-end DNN based OD methods (RSRAE) are indeed making progress on relatively simple benchmarks like MNIST and Fashion-MNIST, their performance on difficult datasets like CIFAR10 is still as unsatisfactory as previous counterparts. Besides, MOGAAL performs poorly in almost all cases, which suggests that generating proper pseudo outliers are s till very difficult for deep OD by now.

4.2.2 Score Refinement

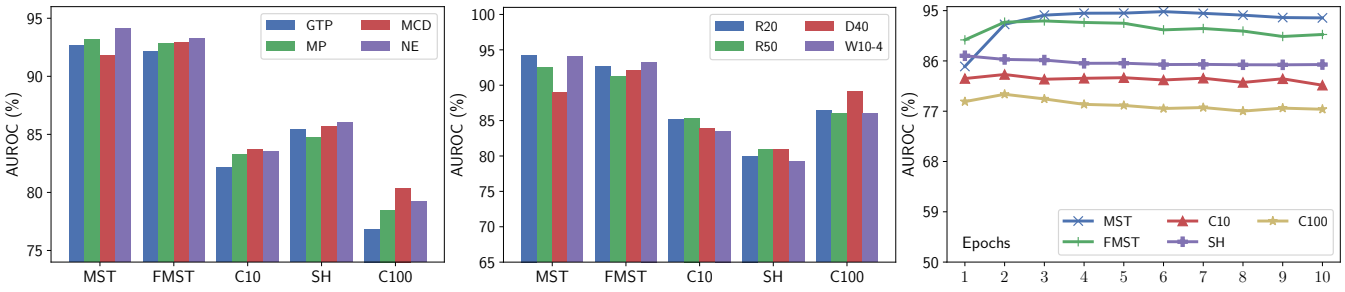
In this section, we validate the effectiveness of score refinement for discriminative *E*^{3}*Outlier*. As shown in Table 2, JSR enables consistent performance improvement under different outlier ratios and all evaluation metrics. To show the effect of each score refinement strategy, we further compare the OD performance of five cases in terms of AUROC: Baseline using no score refinement (BAS), using the online re-weighting strategy only (ORW), with the reboot re-weighting strategy only (RRW), using the ensemble strategy only (ENS) and using the joint score refinement (JSR), under $\rho = 10\%$ with default NE score for discriminative *E*^{3}*Outlier*. We report the results in Table 3, from which the following facts are drawn: First, when compared with the baseline (BAS), score refinement strategies are able to produce performance gain on all benchmarks by up to 2.1% AUROC gain. The improvement tends to be more tangible on comparatively difficult benchmarks like CIFAR100. Besides, under other outlier ratios, using score refinement also produces stable performance improvement (1% to 2% AUROC) on difficult benchmarks. Second, RRW tends to be slightly better than ORW, while ORW enjoys lower computational cost. Finally, the joint score refinement (JSR) with both reboot re-weighting and ensemble is typically better than a single score refinement strategy, except for the case Fashion-MNIST where JSR performs comparably to other refinement strategies. We also discuss the parameters in score refinement in Sec. 4 of supplementary material.

TABLE 3: comparison of score refinement strategies (in %).

CONFIG.	MST	FMST	C10	SH	C100
BAS	94.1	93.3	83.5	86.0	79.2
BAS+ORW	94.4	93.6	84.1	86.7	80.3
BAS+RRW	94.6	93.6	84.4	86.5	80.5
BAS+ENS	94.3	93.4	84.1	86.7	80.7
BAS+JSR	94.9	93.5	84.7	87.1	81.3

4.2.3 Discussion

In this section, we discuss several key factors in *E*^{3}*Outlier*. Similarly, we conduct experiments under $\rho = 10\%$ to show the general trends. We investigate the following factors of discriminative *E*^{3}*Outlier*: **(1)** Outlier scores: We compare four different outlier scores for discriminative *E*^{3}*Outlier*, i.e. GTP/MP/MCD/NE. As shown by Fig. 8a, uncertainty based scores (MP/MCD/NE) basically prevail over the baseline GTP score, which validates the advantages of exploring network uncertainty as outlierness measure for *E*^{3}*Outlier*. Among uncertainty based outlier scores, MCD and NE are prone to outperform the simplest MP. Although MCD achieves the best performance on some benchmarks, it requires multiple forward passes and tends to be less efficient than NE. By contrast, NE consistently outperforms the baseline by a notable margin, and it realizes a good trade-off between performance and efficiency. **(2)** The network architecture of SSD: With other settings fixed, we additionally explore ResNet20/ResNet50 [19] and DenseNet40 [85] as the backbone architecture for SSD (shown in Fig. 8b). Despite of some differences, those frequently-used architectures basically perform satisfactorily. Interestingly, we note that a more complex architecture (ResNet50/DenseNet40) tends to be more effective on relatively complex datasets (CIFAR10, SVHN and CIFAR100), but its performance is inferior on simpler datasets. **(3)** Training epochs (see Fig. 8c): We measure the OD performance when the SSD is trained by different epoch numbers to evaluate its impact on self-supervised learning. In general, we notice that the OD performance is



(a) Different outlier scores. (b) Different network architectures. (c) Performance with different training epochs.

Fig. 8: Different factors' influence on E^3*Outlier*'s performance under $\rho = 10\%$.

inclined to be improved at the initial stage of training (less than $\lceil \frac{250}{K} \rceil$ training epochs) and then reach a plateau. No drastic performance changes are observed as the training epochs continue to increase. **(4)** Pseudo label design. Since the operation set is often constructed by a composite of multiple types of transformations, it is natural to consider a multi-label way to assign pseudo labels. To explore its possibility, we assign each transformed datum with 5 labels based on the performed transformations: Simple rotation label (4 classes in total), translation label ($3 \times 3 = 9$ classes in total), irregular rotation label (8+1=9 classes in total), flip label (2 classes in total) and patch re-arranging label (23+1=24 classes in total). The DNN is equiped with 5 classification heads to predict 5 labels, while the outlier score is computed by averaging the outlier scores yielded by 5 heads. We report the performance of such a multi-label setup in Table 4, and the results suggest that it can yield slightly better performance on most benchmark datasets. Thus, it is possible to explore a more effective design of pseudo labels for E^3*Outlier*. For generative and contrastive E^3*Outlier*, we investigate two major factors: **(1)** Backbone architecture for generative E^3*Outlier*. In fact, one can explore different backbone architecture to implement the generative DNN $\mathcal{G}$ for generative E^3*Outlier*, and we test UNet as an example. As shown in Table 5, the results suggest that UNet is also able to yield fairly satisfactory OD performance, and we notice that UNet performs evidently better than CAE on relatively difficult datasets CIFAR10/SVHN/CIFAR100, while CAE tends to be better on simpler MNIST/Fashion-MNIST. **(2)** Classification loss $\mathcal{L}_{cls}$ for contrastive E^3*Outlier*. It is noted that the loss of classification $\mathcal{L}_{cls}$ when training the DNN model of contrastive E^3*Outlier*, and we also discuss the case where only the contrastive loss $\mathcal{L}_{scl}$ is applied. Interestingly, contrastive E^3*Outlier* without $\mathcal{L}_{cls}$ yields sigificantly worse performance on CIFAR10/CIFAR100 (77.3%/76.6% AUROC under $\rho = 10\%$), but the performance is better on SVHN (91.7% AUROC under $\rho = 10\%$). The reason is that the performance on "0" class of SVHN suffers from a drastic degradation when classification is performed, as "0" is still a "0" aften a rotation of 90, 180 or 270 degrees. Thus, the classification task is completely invalid in this case.

4.3 E^3*Outlier* based Video Abnormal Event Detection

4.3.1 Unsupervised Video Abnormal Event Detection

Inspired by E^3*Outlier*'s success with images, it is natural to explore E^3*Outlier* for other type of visual data, e.g. videos. To this end, unsupervised video abnormal event detection (UVAD) [10] is exactly an application of deep OD to videos. UVAD is an emerging task that aims to detect those unusual events that divert from other frequently-encountered routine in completely unlabeled video sequences. As it does not require labeling and enumerating normal video events to construct a training set, UVAD is more challenging than semi-supervised VAD that has been thoroughly studied [86]. Most existing UVAD solutions approach UVAD by change detection and its variants [10], [87], [88], while the recent work [89] also proposes a different solution that first initializes the detection results based on IF and pre-trained DNNs, and then refines the detection iteratively. However, existing UVAD solutions typically perform unsatisfactorily.

4.3.2 Design of E^3Outlier based UVAD Solution

Before we tailor the E^3*Outlier* for UVAD, we notice two important differences between UVAD and previous outlier image removal task: First, despite that discriminative and contrastive E^3*Outlier* are shown to be highly effective in detecting outlier images by appearance information (e.g. structure and texture), normal and abnormal video events are often conducted by the same type of subjects in UVAD (For example, humans in Fig. 9). In other words, appearance differences are less important to UVAD. Second, unlike static images, videos are described by both appearance and motion information. As motion is the key to detecting many abnormal events, optical flow maps of video frames are often computed to describe the motion in videos. Therefore, both raw video frames and optical flow maps are supposed to be exploited for providing self-supervision. Due to those differences, we naturally turn to generative E^3*Outlier* to connect both appearance and motion view. Based on generative E^3*Outlier*, the designed UVAD solution is presented below:

First of all, we follow our previous work [90] to extract and represent video events: Foreground objects in each video frame are first localized by a series of regions of interest (RoIs). Then, 5 rectangular patches are extracted from current and 4 neighboring frames by the location of each RoI. Afterwards, they are normalized into 32×32 and stacked into a $5 \times 32 \times 32$ spatio-temporal cube (STC) $\mathbf{x} = [p_1; \cdots; p_5]$, where p_i is a normalized patch ($i = 1, \cdots, 5$). Note that a STC $\mathbf{x}$ serves as the basic representation of a video event, because it not only describes the foreground object but also contains its motion in a time interval. To apply generative E^3*Outlier*, we then design the operation

TABLE 4: Performance comparison (in %) of discriminative *E*^{3}*Outlier* with single-label (SL) and multi-label (ML) learning.

Dataset	MNIST			Fashion-MNIST			CIFAR10			SVHN			CIFAR100		
	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O
E^{3}Out. (SL)	94.1	99.3	67.5	**93.3**	**99.0**	**75.9**	83.5	97.5	43.4	86.0	98.0	36.7	79.2	96.8	33.3
E^{3}Out. (ML)	**95.4**	**99.5**	**71.1**	92.7	98.9	72.9	**84.1**	**97.6**	**45.1**	**86.9**	**98.1**	**38.5**	**80.0**	**97.0**	**34.9**

TABLE 5: Performance comparison (in %) of different DNN models for generative *E*^{3}*Outlier*.

Dataset	MNIST			Fashion-MNIST			CIFAR10			SVHN			CIFAR100		
	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O
CAE	**86.7**	**96.4**	**60.3**	**89.6**	**98.5**	**61.6**	66.3	93.5	20.0	63.6	93.9	15.0	61.2	92.4	16.7
UNet	82.0	95.0	56.5	86.4	98.0	52.8	**72.2**	**92.0**	**26.1**	**68.5**	**94.7**	**18.6**	**65.5**	**93.5**	**20.5**

(a) A person riding in the crowd. (b) A skater and a riding person. (c) A student throwing his backpack.

Fig. 9: Examples of abnormal events on UCSDped1, UCSDped2 and Avenue datasets (walking pedestrians are normal).

TABLE 6: Performance comparison of state-of-the-art UVAD methods with our *E*^{3}*Outlier* based UVAD solution in terms of frame-level AUC ("-" indicates that the performance is not reported).

	UCSDped1	UCSDped2	Avenue
SCD [10]	59.6%	63.0%	78.3%
UM [87]	68.4%	82.2%	80.6%
MC2ST [88]	71.8%	87.5%	84.4%
DOR [89]	71.7%	83.2%	-
E^{3}Out.	**79.5%**	**92.6%**	**89.2%**

$O(\cdot|y_1)$ and $O(\cdot|y_2)$ as follows: Given an input STC, $O(\cdot|y_1)$ is defined by $O(\mathbf{x}|y_1) = [p_1; p_2; p_4; p_5]$, which means deleting the middle patch in the STC $\mathbf{x}$. Meanwhile, we devise two types of $O(\cdot|y_2)$: (1) $O(\mathbf{x}|y_2) = p_3$, which suggests fetching the middle patch of $\mathbf{x}$. (2) $O(\mathbf{x}|y_2) = OF(p_3)$, which means transforming p_3 into its corresponding optical flow map. In this way, we actually define a self-supervised learning task that aims to infer p_3 and its optical flow map based on $\mathbf{x}$'s remaining patches p_1, p_2, p_4, p_5. We simple use CAE to carry out this generative task. As described in Sec. 3.6.1, we can train the models by the objective in Eq. (14) and score each STC by Eq. (15). The scores yielded by two types of $O(\cdot|y_2)$ operations are normalized and then summed to obtain the final score of each STC. The minimum of all STCs' scores on a frame is viewed as the frame score. More details are provided in Sec. 5 of supplementary material.

4.3.3 Performance Evaluation and Comparison

To evaluate the performance of our UVAD solution, we conduct experiments on three most commonly-used VAD benchmark datasets: UCSDped1 [91], UCSDped2 [91] and Avenue [92]. Following the standard practice in VAD, we compute frame-level AUC [91] as the quantitative performance measure, and compare our method with latest state-of-the-art UVAD approaches: Shuffled change detection (SCD) [10], Unmasking (UM) [87], Multiple Classifier Two Sample Test (MC2ST) [88], and Deep Ordinal Regression (DOR) [89]. The results are displayed in Table 6, and we can discover that the proposed *E*^{3}*Outlier* based UVAD solution outperforms existing UVAD solutions by by a 4% to 10% frame-level AUROC, which justifies *E*^{3}*Outlier* as a flexible and effective solution to different OD applications. Besides, unlike SCD, UM and MC2ST that require feature extraction based on hand-crafted descriptors, the proposed *E*^{3}*Outlier* based solution achieves end-to-end UVAD, while it also leads the other deep UVAD solution DOR by a huge margin.

5 Conclusion

In this paper, we propose a self-supervised deep OD framework named *E*^{3}*Outlier*. *E*^{3}*Outlier* for the first time leverages discriminative self-supervised learning for deep OD, which facilitates more effective representation learning from raw images. Then we demonstrate inlier priority, a property that lays the foundation for end-to-end OD, by both theory and empirical validations. Afterwards, we illustrate how the network uncertainty of discriminative DNNs can be utilized as a new outlierness measure, and present three specific outlier scores that can outperform the baseline. Then, the joint score refinement that fuses two types of strategies can be used to further boost OD performance. Finally, we demonstrate the applicability of *E*^{3}*Outlier* to different learning paradigms and other deep OD applications.

REFERENCES

[1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.

[2] A. Boukerche, L. Zheng, and O. Alfandi, "Outlier detection: Methods, models, and classification," *ACM Computing Surveys (CSUR)*, vol. 53, no. 3, pp. 1–37, 2020.

[3] U. Fiore, A. De Santis, F. Perla, P. Zanetti, and F. Palmieri, "Using generative adversarial networks for improving classification effectiveness in credit card fraud detection," *Information Sciences*, vol. 479, pp. 448–455, 2019.

[4] H. Soleimani and D. J. Miller, "Atd: Anomalous topic discovery in high dimensional discrete data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 9, pp. 2267–2280, 2016.

[5] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *International Conference on Information Processing in Medical Imaging*. Springer, 2017, pp. 146–157.

[6] J. Mao, T. Wang, C. Jin, and A. Zhou, "Feature grouping-based outlier detection upon streaming trajectories," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 12, pp. 2696–2709, 2017.

[7] D. M. J. Tax, "One-class classification," *Applied Sciences*, 2001.

[8] W. Liu, G. Hua, and J. R. Smith, "Unsupervised one-class learning for automatic outlier removal," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3826–3833.

[9] Y. Xia, X. Cao, F. Wen, G. Hua, and J. Sun, "Learning discriminative reconstructions for unsupervised outlier removal," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1511–1519.

[10] A. Del Giorno, J. A. Bagnell, and M. Hebert, "A discriminative framework for anomaly detection in large videos," in *European Conference on Computer Vision*. Springer, 2016, pp. 334–349.

[11] S. Wang, Y. Zeng, Q. Liu, C. Zhu, E. Zhu, and J. Yin, "Detecting abnormality without knowing normality: A two-stage approach for unsupervised video abnormal event detection," in *2018 ACM Multimedia Conference on Multimedia Conference*. ACM, 2018, pp. 636–644.

[12] S. Wang, E. Zhu, X. Hu, X. Liu, Q. Liu, J. Yin, and F. Wang, "Robustness can be cheap: A highly efficient approach to discover outliers under high outlier ratios," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 5313–5320.

[13] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2. Ieee, 1999, pp. 1150–1157.

[14] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 3360–3367.

[15] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[16] C. C. Aggarwal, "An introduction to outlier analysis," in *Outlier analysis*. Springer, 2017, pp. 1–34.

[17] H. Wang, M. J. Bah, and M. Hammad, "Progress in outlier detection techniques: A survey," *Ieee Access*, vol. 7, pp. 107 964–108 000, 2019.

[18] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM Computing Surveys (CSUR)*, vol. 54, no. 2, pp. 1–38, 2021.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[20] S. Wang, Y. Zeng, X. Liu, E. Zhu, J. Yin, C. Xu, and M. Kloft, "Effective end-to-end unsupervised outlier detection via inlier priority of discriminative network," in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2019, pp. 5962–5975.

[21] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," in *ACM Sigmod Record*, vol. 29, no. 2. ACM, 2000, pp. 427–438.

[22] E. M. Knox and R. T. Ng, "Algorithms for mining distancebased outliers in large datasets," in *Proceedings of the international conference on very large data bases*. Citeseer, 1998, pp. 392–403.

[23] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *ACM sigmod record*, vol. 29, no. 2. ACM, 2000, pp. 93–104.

[24] J. Tang, Z. Chen, A. W.-C. Fu, and D. W. Cheung, "Enhancing effectiveness of outlier detections for low density patterns," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2002, pp. 535–548.

[25] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "Loop: local outlier probabilities," in *Proceedings of the 18th ACM conference on Information and knowledge management*, 2009, pp. 1649–1652.

[26] T. R. Bandaragoda, K. M. Ting, D. Albrecht, F. T. Liu, and J. R. Wells, "Efficient anomaly detection by isolation using nearest neighbour ensemble," in *2014 IEEE International Conference on Data Mining Workshop*. IEEE, 2014, pp. 698–705.

[27] G. Pang, K. M. Ting, and D. Albrecht, "Lesinn: Detecting anomalies by identifying least similar nearest neighbours," in *2015 IEEE international conference on data mining workshop (ICDMW)*. IEEE, 2015, pp. 623–630.

[28] X. Yang, L. J. Latecki, and D. Pokrajac, "Outlier detection with globally optimal exemplar-based gmm," in *Proceedings of the 2009 SIAM International Conference on Data Mining*. SIAM, 2009, pp. 145–154.

[29] X.-m. Tang, R.-x. Yuan, and J. Chen, "Outlier detection in energy disaggregation using subspace learning and gaussian mixture model," *International Journal of Control and Automation*, vol. 8, no. 8, pp. 161–170, 2015.

[30] L. J. Latecki, A. Lazarevic, and D. Pokrajac, "Outlier detection with kernel density functions," in *International Workshop on Machine Learning and Data Mining in Pattern Recognition*. Springer, 2007, pp. 61–75.

[31] A. P. Boedihardjo, C.-T. Lu, and F. Chen, "Fast adaptive kernel density estimator for data streams," *Knowledge and Information Systems*, vol. 42, no. 2, pp. 285–317, 2015.

[32] L. Zhang, J. Lin, and R. Karim, "Adaptive kernel density-based anomaly detection for nonlinear systems," *Knowledge-Based Systems*, vol. 139, pp. 50–63, 2018.

[33] X. Qin, L. Cao, E. A. Rundensteiner, and S. Madden, "Scalable kernel density estimation-based local outlier detection over large data streams." in *EDBT*, 2019, pp. 421–432.

[34] M.-F. Jiang, S.-S. Tseng, and C.-M. Su, "Two-phase clustering process for outliers detection," *Pattern recognition letters*, vol. 22, no. 6-7, pp. 691–700, 2001.

[35] Z. He, X. Xu, and S. Deng, "Discovering cluster-based local outliers," *Pattern Recognition Letters*, vol. 24, no. 9-10, pp. 1641–1650, 2003.

[36] J. Yin and J. Wang, "A model-based approach for text clustering with outlier detection," in *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*. IEEE, 2016, pp. 625–636.

[37] M. Chenaghlou, M. Moshtaghi, C. Leckie, and M. Salehi, "Online clustering for evolving data streams with online anomaly detection," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2018, pp. 508–521.

[38] B. Schölkopf, A. Smola, and K.-R. Müller, "Kernel principal component analysis," in *International Conference on Artificial Neural Networks*. Springer, 1997, pp. 583–588.

[39] J. Chen, S. Sathe, C. Aggarwal, and D. Turaga, "Outlier detection with autoencoder ensembles," in *Proceedings of the 2017 SIAM international conference on data mining*. SIAM, 2017, pp. 90–98.

[40] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 2008, pp. 413–422.

[41] S. Hariri, M. C. Kind, and R. J. Brunner, "Extended isolation forest," *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[42] Y. Wang, S. Parthasarathy, and S. Tatikonda, "Locality sensitive outlier detection: A ranking driven approach," in *2011 IEEE 27th International Conference on Data Engineering*. IEEE, 2011, pp. 410–421.

[43] T. Pevnỳ, "Loda: Lightweight on-line detector of anomalies," *Machine Learning*, vol. 102, no. 2, pp. 275–304, 2016.

[44] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," *arXiv preprint arXiv:1901.03407*, 2019.

[45] S. Zhai, Y. Cheng, W. Lu, and Z. Zhang, "Deep structured energy based models for anomaly detection," *international conference on machine learning*, pp. 1100–1109, 2016.

[46] C. Zhou and R. C. Paffenroth, "Anomaly detection with robust deep autoencoders," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017, pp. 665–674.

[47] R. Chalapathy, A. K. Menon, and S. Chawla, "Robust, deep and inductive anomaly detection," in *Joint European Conference on Ma-*

chine Learning and Knowledge Discovery in Databases. Springer, 2017, pp. 36–51.

[48] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," in *International Conference on Learning Representations (ICLR)*, 2018.

[49] G. Pang, L. Cao, L. Chen, and H. Liu, "Learning representations of ultrahigh-dimensional data for random distance-based outlier detection," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 2041–2050.

[50] Y. Liu, Z. Li, C. Zhou, Y. Jiang, J. Sun, M. Wang, and X. He, "Generative adversarial active learning for unsupervised outlier detection," *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[51] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Advances in neural information processing systems*, vol. 3, no. 06, 2014.

[52] C. Lai, D. Zou, and G. Lerman, "Robust subspace recovery layer for unsupervised anomaly detection," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2020.

[53] N. Tajbakhsh, Y. Hu, J. Cao, X. Yan, Y. Xiao, Y. Lu, J. Liang, D. Terzopoulos, and X. Ding, "Surrogate supervision for medical image analysis: Effective deep learning from limited quantities of labeled data," in *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*. IEEE, 2019, pp. 1251–1255.

[54] D. Zhang, J. Han, and Y. Zhang, "Supervision by fusion: Towards unsupervised learning of deep salient object detector," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4048–4056.

[55] X. Ji, J. F. Henriques, and A. Vedaldi, "Invariant information clustering for unsupervised image classification and segmentation," *arXiv: Computer Vision and Pattern Recognition*, 2018.

[56] N. Komodakis and S. Gidaris, "Unsupervised representation learning by predicting image rotations," in *International Conference on Learning Representations (ICLR)*, 2018.

[57] I. Golan and R. El-Yaniv, "Deep anomaly detection using geometric transformations," in *Advances in Neural Information Processing Systems*, 2018, pp. 9758–9769.

[58] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "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)*, 2019, pp. 4171–4186.

[59] R. Santa Cruz, B. Fernando, A. Cherian, and S. Gould, "Visual permutation learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[60] I. Misra, C. L. Zitnick, and M. Hebert, "Shuffle and learn: unsupervised learning using temporal order verification," in *European Conference on Computer Vision*. Springer, 2016, pp. 527–544.

[61] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3733–3742.

[62] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey." *arXiv: Computer Vision and Pattern Recognition*, 2019.

[63] A. Malinin and M. Gales, "Predictive uncertainty estimation via prior networks," in *Advances in Neural Information Processing Systems*, 2018, pp. 7047–7058.

[64] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*, 2016, pp. 1050–1059.

[65] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in neural information processing systems*, 2017, pp. 6402–6413.

[66] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," 2017.

[67] J. Snoek, Y. Ovadia, E. Fertig, B. Lakshminarayanan, S. Nowozin, D. Sculley, J. Dillon, J. Ren, and Z. Nado, "Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift," in *Advances in Neural Information Processing Systems*, 2019, pp. 13 969–13 980.

[68] D. M. Hawkins, *Identification of outliers*. Springer, vol. 11.

[69] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *International Conference on Machine Learning*, 2016, pp. 1558–1566.

[70] A. Dosovitskiy and T. Brox, "Generating images with perceptual similarity metrics based on deep networks," in *Advances in neural information processing systems*, 2016, pp. 658–666.

[71] J. Gui, Z. Sun, Y. Wen, D. Tao, and J. Ye, "A review on generative adversarial networks: Algorithms, theory, and applications," *arXiv preprint arXiv:2001.06937*, 2020.

[72] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *British Machine Vision Conference 2016*. British Machine Vision Association, 2016.

[73] R. Anand, K. G. Mehrotra, C. K. Mohan, and S. Ranka, "An improved algorithm for neural network classification of imbalanced training sets," *IEEE Transactions on Neural Networks*, vol. 4, no. 6, pp. 962–969, 1993.

[74] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1321–1330.

[75] T. G. Dietterich, "Ensemble methods in machine learning," in *International workshop on multiple classifier systems*. Springer, 2000, pp. 1–15.

[76] O. Ronneberger, P. Fischer, and T. Brox, *U-Net: Convolutional Networks for Biomedical Image Segmentation*. Springer, Cham, 2015.

[77] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[78] J. Tack, S. Mo, J. Jeong, and J. Shin, "Csi: Novelty detection via contrastive learning on distributionally shifted instances," *Advances in Neural Information Processing Systems*, vol. 33, pp. 11 839–11 852, 2020.

[79] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[80] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[81] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Citeseer, Tech. Rep., 2009.

[82] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," 2011.

[83] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 233–240.

[84] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *International Conference on Artificial Neural Networks*. Springer, 2011, pp. 52–59.

[85] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[86] B. Ramachandra, M. Jones, and R. R. Vatsavai, "A survey of single-scene video anomaly detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[87] R. Tudor Ionescu, S. Smeureanu, B. Alexe, and M. Popescu, "Unmasking the abnormal events in video," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2895–2903.

[88] Y. Liu, C.-L. Li, and B. Póczos, "Classifier two sample test for video anomaly detections." in *BMVC*, 2018, p. 71.

[89] G. Pang, C. Yan, C. Shen, A. V. D. Hengel, and X. Bai, "Self-trained deep ordinal regression for end-to-end video anomaly detection," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12 170–12 179, 2020.

[90] G. Yu, S. Wang, Z. Cai, E. Zhu, C. Xu, J. Yin, and M. Kloft, "Cloze test helps: Effective video anomaly detection via learning to complete video events," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 583–591.

[91] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos, "Anomaly detection in crowded scenes," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 1975–1981.

[92] C. Lu, J. Shi, and J. Jia, "Abnormal event detection at 150 fps in matlab," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 2720–2727.

Siqi Wang received the Ph.D. degree in computer science and technology from the National University of Defense Technology (NUDT), China. He is currently an assistant research professor in College of Computer, NUDT. His main research include outlier/anomaly detection and unsupervised learning. His works have been published on leading conferences and journals, such as NeurIPS, AAAI, IJCAI, ACM MM, TPAMI, TIP, PR, TCYB and Neurocomputing. He serves as a PC member and reviewer for top-tier conference like NeurIPS and AAAI and several prestigious journals.

Yijie Zeng received the B.Sc. in computational mathematics from University of Science and Technology of China in 2015, and the Ph.D. degree in the School of Electrical and Electronic Engineering from Nanyang Technological University, Singapore in 2020. His research interests include machine learning, computer vision, and pattern recognition.

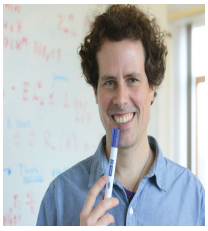
Guang Yu received the bachelor's degree in computer science and technology from Sichuan University, Chengdu, China, in 2018. He is currently working toward the Ph.D. degree at the College of Computer, National University of Defense Technology, Changsha, China. His main research interests include anomaly/outlier detection and self-supervised/unsupervised learning.

Zhen Cheng is currently pursuing the Ph.D. degree with the National University of Defense Technology (NUDT), China. His current research interests include transfer learning, outlier detection, and deep neural networks.

Xinwang Liu received his PhD degree from National University of Defense Technology (NUDT), China. He is now Professor of School of Computer, NUDT. His current research interests include kernel learning and unsupervised feature learning. Dr. Liu has published 60+ peer-reviewed papers, including those in highly regarded journals and conferences such as TPAMI, TKDE, TIP, TNNLS, TMM, TIFS, NeurIPS, ICCV, CVPR, AAAI, IJCAI, etc.

Sihang Zhou received his PhD degree from National University of Defense Technology (NUDT), China. He is now lecturer at College of Intelligence Science and Technology, NUDT. His current research interests include machine learning and medical image analysis. Dr. Zhou has published 20+ peer-reviewed papers, including IEEE T-IP, IEEE T-NNLS, IEEE T-MI, Information Fusion, Medical Image Analysis, AAAI, MICCAI.

En Zhu received his PhD degree from National University of Defense Technology (NUDT), China. He is now Professor at School of Computer Science, NUDT, China. His main research interests are pattern recognition, image processing, machine vision and machine learning. Dr. Zhu has published 60+ peer-reviewed papers, including IEEE T-CSVT, IEEE T-NNLS, PR, AAAI, IJCAI, etc. He was awarded China National Excellence Doctoral Dissertation.



Marius Kloft is a professor of computer science at TU Kaiserslautern and an adjunct faculty member of the University of Southern California. Previously he was a junior professor at HU Berlin and a joint postdoctoral fellow at the Courant Institute of Mathematical Sciences and Memorial Sloan-Kettering Cancer Center, New York. He earned his PhD at TU Berlin and UC Berkeley.

Jianping Yin received his PhD degree from National University of Defense Technology (NUDT), China. He is now the distinguished Professor at Dongguan University of Technology. His research interests include pattern recognition and machine learning. Dr. Yin has published 150+ peer-reviewed papers, including IEEE T-CSVT, IEEE T-NNLS, PR, AAAI, IJCAI, etc. He was awarded China National Excellence Doctoral Dissertation' Supervisor and National Excellence Teacher. He served on the Technical Program Committees of 30+ international conferences and workshops.

Qing Liao received her Ph.D. degree in computer science and engineering in 2016 supervised by Prof. Qian Zhang from the Department of Computer Science and Engineering of the Hong Kong University of Science and Technology. She is currently an assistant professor with School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), China. Her research interests include artificial intelligence and bioinformatics.

E^3*Outlier*: A Self-supervised Framework for Unsupervised Deep Outlier Detection

Siqi Wang, Yijie Zeng, Guang Yu, Zhen Cheng, Xinwang Liu, Sihang Zhou,
En Zhu, Marius Kloft, Jianping Yin, Qing Liao

Abstract—Existing unsupervised outlier detection (OD) solutions face a grave challenge with surging visual data like images. Although deep neural networks (DNNs) proves successful for visual data, deep OD remains difficult due to OD's unsupervised nature. This paper proposes a novel framework named *E^3Outlier* that can performs **e**ffective and **e**nd-to-**e**nd deep outlier removal. Its core idea is to introduce *self-supervision* into deep OD. Specifically, our major solution is to adopt a discriminative learning paradigm that creates multiple pseudo classes from given unlabeled data by various data operations, which enables us to apply prevalent discriminative DNNs (e.g. ResNet) to the unsupervised OD problem. Then, with theoretical and empirical demonstration, we argue that inlier priority, a property that encourages DNN to prioritize inliers during self-supervised learning, makes it possible to perform end-to-end OD. Meanwhile, unlike frequently-used outlierness measures (e.g. density, proximity) in previous OD methods, we explore network uncertainty and validate it as a highly effective outlierness measure, while two practical score refinement strategies are also designed to improve OD performance. Finally, in addition to the discriminative learning paradigm above, we also explore the solutions that exploit other learning paradigms (i.e. generative learning and contrastive learning) to introduce self-supervision for *E^3Outlier*. Such extendibility not only brings further performance gain on relatively difficult datasets, but also enables *E^3Outlier* to be applied to other OD applications like video abnormal event detection. Extensive experiments demonstrate that *E^3Outlier* can considerably outperform state-of-the-art counterparts by 10%-30% AUROC. All codes are available at https://github.com/demonzyj56/E3Outlier.

Index Terms—outlier detection, deep neural networks, unsupervised learning, self-supervised learning

1 Introduction

In realms like machine learning and data science, outliers, which are also called novelties, anomalies, deviants, exceptions, irregularities, etc [1], have a pervasive existence. Outlier detection (OD), which may also be referred as unsupervised anomaly/outlier detection, is a long-standing problem that draws continuous attention from the research community. To provide a clear and strict formulation of OD problem, this paper follows the definition used in the recent OD survey paper [2]: Given a set of data instances, OD is an unsupervised task that aims to identify those instances that deviate significantly from the rest of data. Thus, outliers are discerned from given unlabeled data by a *transductive* learning setup. OD is of great importance in practice: First, as data labeling is usually expensive and time-consuming, it is often required to deal with massive unlabeled data. As a result, OD has been a frequently-encountered unsupervised task when handling prevalent unlabeled data. Second, even for supervised/semi-supervised tasks, OD plays a vital role in the data cleansing stage (e.g. removing wrongly-labeled data or noise when building a data set), which is the foundation for obtaining high-quality models. OD enjoys a variety of real-world applications, such as financial fraud detection [3], emerging topic detection [4], computer-aided medical diagnosis [5], motion trajectory analysis [6], etc. Since the only prior knowledge is that outliers have rare occurrence when compared with inliers, no supervision information is available for OD here. Due to its unsupervised nature, OD is usually addressed by exploiting some intrinsic properties of data, e.g. density, proximity, cluster membership, etc. A more detailed review of classic OD is given in Sec. 2.1. In particular, we distinguish OD in this paper from the *(semi-supervised) anomaly detection* or *one-class classification* [7], which builds a normality model from a pure set of labeled normal data and detects deviants in a separated test set by an *inductive* learning setup. To avoid any confusion, a detailed clarification of terms is also provided in Sec. 6 of the supplementary material, so as to differentiate OD here from other relevant but different realms like (semi-supervised) anomaly detection and out-of-distribution detection.

With the widespread use of photographic equipment (e.g. cameras, smart phones), visual data like images and videos have undergone an explosive growth in these years. In this context, a marriage of OD and visual data is pretty natural, and it gives birth to many novel applications, such as the refinement of web image search results [8], [9] and video abnormal event detection [10], [11]. Among various forms of visual data, images have constantly played

- *S. Wang, G. Yu, Z. Cheng, X. Liu and E. Zhu are with College of Computer, National University of Defense Technology (NUDT), Changsha, 410073, China. E-mail: {wangsiqi10c, xinwangliu, enzhu}@nudt.edu.cn.*
- *Y. Zeng is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798. E-mail: yzeng004@e.ntu.edu.sg.*
- *S. Zhou is with College of Intelligent Science and Technology, NUDT, Changsha, 410073, China. E-mail: sihangjoe@gmail.com.*
- *Marius Kloft is with Department of Computer Science, TU Kaiserslautern, Germany. E-mail: kloft@cs.uni-kl.de.*
- *J. Yin is with Dongguan University of Technology, Dongguan, 523808, China. E-mail: jpyin@dgut.edu.cn.*
- *Q. Liao is with School of Computer Science and Technology, Harbin Institue of Technology (Shenzhen), China. E-mail: liaoqing@hit.edu.cn.*

Manuscript received June 30, 2020.

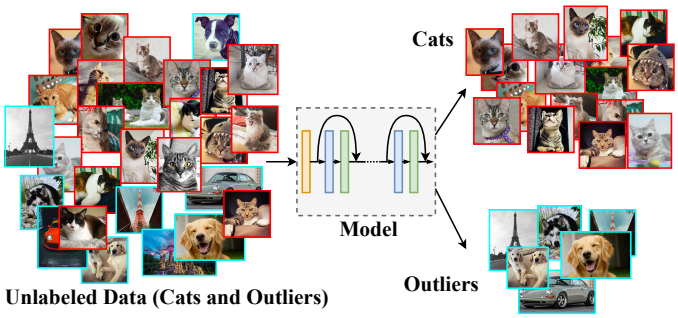


Fig. 1: An example of deep outlier image removal task.

a fundamental role in all sorts of visual analysis. Therefore, this paper will focus on OD for image data, i.e. the image outlier removal task. For an intuitive illustration, we show an example that aims to remove outliers from images of cats (inliers) in Fig. 1. Compared with frequently-seen tabular data (or vectorized data), image data exhibit evidently different characteristics: They possess a variety of high-level spatial structures that are endowed with rich semantics, and low-level details (i.e. image pixels) alone are much less meaningful to perception. As a consequence, a direct application of those classic OD methods to image data usually leads to poor performance, and proper image representations will be a prerequisite for successful outlier removal. As a simple solution, some works [8], [12] extract the image representations by hand-crafted feature descriptors (e.g. SIFT [13], sparsity-constrained linear coding [14]), and then feed the extracted feature vectors into a classic OD method. However, such solutions bring about complex feature engineering issues, and they often suffer from sub-optimal image representations and poor transferability. To this end, an emerging trend is to learn good representations automatically via deep neural networks (DNNs) during the learning process, so as to realize a certain goal like image classification or segmentation. Such an end-to-end deep learning paradigm has achieved remarkable success in computer vision, especially with discriminative DNNs for supervised learning tasks [15]. However, although introducing DNNs for deep outlier removal seems to be pretty straightforward, a both *effective* and *end-to-end* DNN based OD solution still requires exploration. The major impediment to developing such a solution lies in the unsupervised nature of the OD task, i.e. the absence of data labels results in a lack of supervision signal. Consequently, as several recent surveys point out [2], [16], [17], [18], auto-encoder (AE) still plays a dominant role in deep OD, while other widely-used DNNs like discriminative ResNet [19] are not directly applicable for deep OD without any given labels.

To bridge those gaps in deep OD, we propose the first self-supervised framework termed $E^3Outlier$, which aims to realize both *effective* and *end-to-end* deep outlier removal. Specifically, our core idea is to remedy the label absence in OD by introducing self-supervision. To this end, our major solution is to create multiple pseudo classes from given unlabeled data by imposing certain data operations like rotation and patch re-arranging. With labels of those pseudo classes, powerful discriminative DNNs that have been thoroughly studied can be exploited in OD and enable more effective representation learning. Second, in order to further conduct end-to-end OD, we unveil a property named "inlier priority": Even though inliers and outliers are indiscriminately fed into the DNN during self-supervised learning, the DNN tends to prioritize inliers' loss reduction. We provide both theoretical and empirical demonstration to this property. Third, instead of commonly-used outlierness measure (e.g. density and proximity), we point out that the DNN uncertainty in self-supervised learning can be leveraged to design highly effective outlier scores. Meanwhile, inspired by the inlier priority and network uncertainty, we develop two practical strategies and fuse them into a score refinement stage to yield performance enhancement. Finally, in addition to the aforementioned discriminative learning paradigm, we further design the solution to leverage generative/contrastive learning paradigm to perform self-supervised learning for the proposed $E^3Outlier$ framework. With the extendibility to different learning paradigms, $E^3Outlier$ is not only able to be flexibly applied to other OD applications like video abnormal event detection, but also yield further performance gain on relatively difficult datasets. Our main contributions can be summarized below:

- We for the first time design a self-supervised learning framework for DNN based OD. It not only eases the lack of supervision, but also enables discriminative DNNs to be directly applied to the deep OD problem.
- We unveil a property named inlier priority during self-supervised learning, and theoretical and empirical demonstration are presented to justify this property. It lays the foundation to perform end-to-end OD with the proposed $E^3Outlier$ framework.
- We point out that the uncertainty of discriminative DNN can be exploited as a novel outlierness measure in deep OD, and develop several highly effective uncertainty based outlier scores for end-to-end OD. Moreover, we propose joint score refinement with two practical strategies to boost the OD performance.
- We further design solutions that incorporates generative learning and contrastive learning paradigm into the $E^3Outlier$ framework to provide self-supervision, which endows the proposed framework with more flexibility and better OD performance.

An earlier version of this paper is reported in [20], and this paper is mainly extended in terms of the following aspects: (1) This paper explicitly points out that DNN uncertainty can be used as a new outlierness measure, and intuitively unveils the connection among OD, self-supervised learning and network uncertainty. Compared with this paper, [20] just reported empirical comparison of different outlier scores and did not provide in-depth analysis into the underlying principle of score design. (2) We design several practical strategies to conduct outlier score refinement, which enables the model to achieve consistent performance enhancement against the performance reported in [20] on all benchmark datasets. (3) Unlike [20] that only exploited discriminative learning paradigm for deep OD, this paper further validates the applicability of generative learning or contrastive learning paradigm to $E^3Outlier$. (4) Apart from the image outlier removal task in [20], this paper shows that the proposed $E^3Outlier$ framework is also able to achieve

superior performance in other deep OD application like unsupervised video abnormal event detection.

2 Related Work

2.1 Shallow Model based Outlier Detection

A vast number of shallow methods have been proposed to handle OD, and they usually fall into the following categories: (1) Proximity based methods, which measure the outlierness of a datum by its relation to its neighboring data. Early methods of this type simply assume the data density to be homogeneous, and define some intuitive quantities as outlier scores, such as the distance to the k-th neareast neighbors (k-nn) [21] and the number of neighbors within a pre-defined radius [22]. To this end, Local Outlier Factor (LoF) [23] is the first work that considers local outliers using the average ratio of one datum's neighbor's local reachability density to its own reachability density, which inspires numerous subsequent works, e.g. Connectivity-based Outlier Factor (CoF) [24] considers the degree of connectivity among data when computing outlier scores, while Local Outlier Probability (LoOP) [25] estimates the probability of being an outlier by assuming a half-Guassian distribution on a datum's distance to its k-nn. As computing k-nn can be time-consuming, recent works [26], [27] propose to leverage subsampling and achieve linear time complexity. (2) Statistics based methods, which view data endowed with low likelihood as outliers. The likelihood can be estimated by several statistical models, including parametric and non-parametric statistical models. As to parametric models, the most representative model is Gaussian Mixture Model (GMM) [28], and recently a more robust GMM based OD approach is proposed by Tang et al. [29] by incorporating subspace learning. Meanwhile, as to non-parametric models, kernel density estimation (KDE) [30] is frequently used for OD, while and its its recent variants like [31], [32], [33] are developed to improve its efficiency of OD. (3) Clustering based methods, which view data that do not belong to any major data cluster as outliers. For example, Jiang et al. [34] perform OD by a modified k-means algorithm and constructing a minimal spanning tree from cluster centers. He et al. [35] combine LoF and clustering into CBLOF, which quantitatively distinguishes small and large clusters. To avoid specifying the number of clusters, a recent work by Yan et al. [36] propose to leverage Gibbs Sampling of Dirichlet Process Multinomial Mixture (GSDPMM) for OD. Chenaghlou et al. [37] extends the clustering based OD to online streaming data by considering the evolve of clusters. (4) Projection based methods, which project the original data into a new space to manifest outlierness. Concretely, data can be projected into a low-dimensional embedding by dimension reduction techniques like principal component analysis (PCA) [38] or neural networks like shallow autoencoders [39], and outliers are viewed to be those data that are poorly recovered from the embeddings. In particular, Liu et al. [40] propose Isolation Forest (IF), which projects input data into the tree nodes of random binary trees, and then discriminate outliers by the depth of tree nodes. IF proves to be a both effective and efficient OD method, while recent works by Hariri [41] propose to further improve IF by using random hyperplane cut. Besides, projection techniques like local sensitivity hashing [42] and random projection [43] are also used to reduce complexity of OD models. A more comprehensive review on shallow OD methods can be found in recent survey papers [2], [16], [17], [18]

2.2 DNN based Outlier Detection

As a newly-emerging topic, DNN based OD is highly challenging as it requires to learn suitable data representations for OD. To our best knowledge, only few attempts have been made in the literature. A straightforward idea is to exploit a two-stage solution, which performs representation learning by DNNs first, and then feeds learned features into a separated module that is implemented by some classic OD model (reviewed in [44]). However, such two-stage approaches may suffer from the incompatibility between learned features and the OD module, which can lead to sub-optimal performance. By contrast, state-of-the-art methods usually conduct a joint learning of data representations and outlier scores, and we review each existing solution to our best knowledge below: Xia et al. [9] design a new loss function that encourages a better separation of inliers and outliers by minimizing intra-class variance for multi-layer AE, and propose an adaptive thresholding technique to discriminate outliers; Zhai et al. [45] connect an energy based model with a regularized AE, and develop an energy based score for OD; Zhou et al. [46] utilize a combination of deep AE and Robust Principal Component Analysis (RPCA), which decomposes the matrice of unlabeled data into a low-rank part and a sparse part to represent inliers and outliers respectively, while Chalapathy et al. [47] also adopt a similar idea; Chen et al. [39] propose to generate a set of AEs that possess randomly varied connectivity architecture to perform OD, while adaptive sampling is leveraged to make the approach more efficient and effective. Inspired by Gaussian Mixture Model (GMM), Zong et al. [48] focus on developing an end-to-end OD solution that embeds a GMM density estimation network into the deep AE, and both components are optimized simultaneously; Unlike other methods that rely on AEs, Pang et al. [49] propose a ranking-model based framework named RAMODO, which can be readily incorporated into random distance based OD approach to perform efficient OD with tabular data; Liu et al. [50] convert OD into a binary classification problem via generative adversarial networks (GANs) [51], which are modified to generate simulated outliers; The most recent work [52] exploits the latent low-dimensional subspace structure in data by adding a Robust Subspace Recovery (RSR) regularizer into AE, and two variants, RSRAE and RSRAE+, are proposed for deep outlier removal. As several recent surveys point out [2], [17], [18], AE still plays a center role in existing deep OD solutions due to its unsupervised nature, which motivates us to develop $E^3Outlier$.

2.3 Self-supervised Learning and Network Uncertainty

Self-supervised learning, which is also known as surrogate supervision [53] based learning or pseudo supervision [54] based learning, enjoys a swift growth of popularity in recent research. Its core idea is to construct additional supervision signals from given data by introducing a pretext task. The learning targets of pretext task can be obtained by numerous

ways, such as clustering [55], geometric transformations [56], [57], masking [58], image patch permutation [59], time sequence shuffling [60], contrastive learning [61], etc. As a highly effective pre-training technique or auxiliary task to improve the performance of high-level downstream tasks, self-supervised learning has been explored in many application scenarios, such as image classification, semantic segmentation, object detection and action recognition [62]. To our best knowledge, this is the first work that connects self-supervised learning to unsupervised outlier analysis.

DNN's uncertainty reflects its confidence to a certain prediction, which usually makes it a concept for inductive learning. Several methods have been proposed to quantify network uncertainty, such as Bayesian Neural Networks (BNN) [63], Monte Carlo dropout (MC-Dropout) [64], model ensemble [65], maximum softmax probability [66], information entropy [67], etc. Despite that network uncertainty has drawn increasing attention, its application is typically limited to knowing whether DNN makes trustworthy predictions or detecting the dataset shift. In this paper, we for the first time discuss network uncertainty under a transductive setup, and demonstrate that it can serve as a fairly effective outlierness measure for DNN based OD.

3 The Proposed Framework

3.1 Problem Formulation

Suppose that the data space spanned by all images is denoted by $\mathcal{X}$. DNN based OD deals with a completely unlabeled image data collection $X \subseteq \mathcal{X}$ that is contaminated by outlier images. In other words, X consists of an inlier set X_{in} and an outlier set X_{out}, while $X = X_{in} \cup X_{out}$ and $X_{in} \cap X_{out} = \emptyset$. By the definition of outliers [68], image data of the inlier set are from the same underlying distribution that shares close semantics, but outliers originate from different distributions. Given any image $\mathbf{x} \in \mathcal{X}$, DNN based OD intends to build a scoring model $S(\cdot)$, which takes raw $\mathbf{x}$ as the input and does not perform any prior feature extraction. The goal of $S(\cdot)$ is to output $S(\mathbf{x}) = 1$ for any inlier $\mathbf{x} \in X_{in}$, while $S(\mathbf{x}) = 0$ for any outlier $\mathbf{x} \in X_{out}$. In practice, a larger output $S(\mathbf{x})$ signifies a lower likelihood to be an outlier for $\mathbf{x}$. Besides, within the domain of DNN based OD, *end-to-end* OD refers to the case where both representation learning and OD can be carried out by the same DNN, and no separated classic OD method is involved. In this paper, the proposed *E^{3}Outlier* framework aims to achieve both effective and end-to-end OD.

3.2 Discriminative *E^{3}Outlier*

3.2.1 Motivation

As reviewed in Sec. 2.2, it is noted that AE based solutions play a center role in the deep OD task due to its unsupervised setup. Specifically, deep AE based solutions typically perform unsupervised representation learning by learning to reconstruct the inputs, which is realized by training the deep AE to reduce pixel-wise reconstruction errors like mean square errors (MSE). However, recent researches like [69], [70] demonstrate that such a pixel-wise reconstruction tends to overemphasize low-level image details, which are of very limited interest to human perception. By contrast, semantics of high-level image structures are ignored, but they are actually pivotal to DNN based OD. Another emerging type of generative DNNs is GANs. Despite of fruitful progress, it is still challenging to integrate them into OD [71]: First, it is actually difficult to generate sufficient realistic image outliers, as potential image outliers are infinite and generating high-quality image outliers by GANs is still an open topic; Second, efficient representation learning with GANs is neither straightforward nor easy. By comparison, the supervised discriminative learning paradigm is still the most effective way to learn image semantics and capture high-level structures so far. As a result, these reasons above motivate us to introduce *self-supervision*, so as to enable the use of discriminative learning paradigm in OD.

3.2.2 Self-supervised Discriminative Network (SSD)

The availablity of supervision signals is the key to introduce discriminative DNNs like ResNet [19] and Wide ResNet (WRN) [72] to OD. As image classification is the most fundamental task in supervised learning, creating several pseudo classes from given unlabeled data is a natural idea. Instead of generating a pseudo outlier class like [50], which is a straightforward but difficult task, we propose to build self-supervision by exerting some frequently-seen data operations on given images. Those new data produced by a certain operation are viewed as one pseudo class. Afterwards, we can readily realize representation learning with a discriminative DNN by training it to classify those created pseudo classes. As the discriminative DNN is guided by self-supervision, we term it *self-supervised discriminative network (SSD)* here. Formally, supposing a set of K operations $\mathcal{O} = \{O(\cdot|y)\}_{y=1}^{K}$ is designed to create pseudo classes, we impose the y-th operation $O(\cdot|y)$ on an unlabeled image $\mathbf{x}$ (regardless of an inlier or outlier) and produce a new image $\mathbf{x}^{(y)} = O(\mathbf{x}|y)$. In this way, we can create the y-th pseudo class $X^{(y)} = \{\mathbf{x}^{(y)}|\mathbf{x} \in X\}$, with the pseudo label y assigned to all data in this class. Then, given all data $X' = \{X^{(1)}, \cdots, X^{(K)}\}$ and their label set Y, an SSD with a K-node Softmax layer is trained to perform classification. Like the standard classification process, the SSD is supposed to classify a datum $\mathbf{x}^{(y')}$ into the y'-th pseudo class. The probability vector of $\mathbf{x}^{(y')}$ output by SSD's Softmax layer is denoted as $\mathbf{P}(\mathbf{x}^{(y')}|\boldsymbol{\theta}) = [P^{(y)}(\mathbf{x}^{(y')}|\boldsymbol{\theta})]_{y=1}^{K}$, where $P^{(y)}(\cdot)$ and $\boldsymbol{\theta}$ indicate the probability from the y-th node of Softmax layer and DNN's leanable parameters respectively. To train the SSD, we can minimize the following objective function:

$$\mathcal{L}_{SSD} = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_{SS}(\mathbf{x}_i|\boldsymbol{\theta}) \tag{1}$$

where $\mathcal{L}_{SS}(\mathbf{x}_i|\boldsymbol{\theta})$ represents the loss incurred by $\mathbf{x}_i$ in X during the self-supervised learning. When the standard cross-entropy loss is used, $\mathcal{L}_{SS}(\mathbf{x}_i|\boldsymbol{\theta})$ takes the form below:

$$\mathcal{L}_{DSS}(\mathbf{x}_i|\boldsymbol{\theta}) = -\frac{1}{K}\sum_{y=1}^{K}\log(P^{(y)}(\mathbf{x}_i^{(y)}|\boldsymbol{\theta})) \tag{2}$$

Another key to SSD is the design of data operation. We introduce three sets of operations: Regular affine operation set $\mathcal{O}_{RA}$, irregular affine operation set $\mathcal{O}_{IA}$ and patch rearranging operation set $\mathcal{O}_{PR}$. The general intuition behind

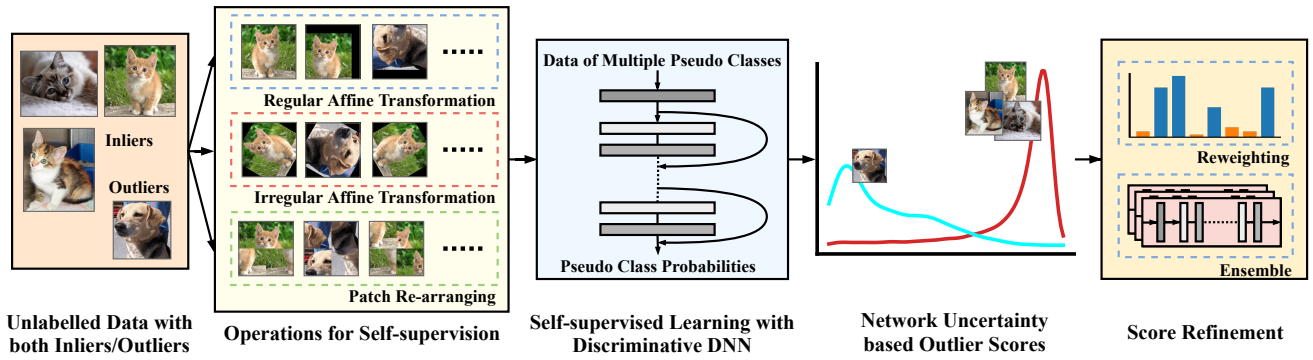


Fig. 2: Overview of the proposed discriminative E^3*Outlier* for deep OD.: Given unlabeled image data polluted by outliers, three operation sets are first imposed on images to create multiple pseudo classes and provide self-supervision. Then, a discriminative DNN is trained to perform the self-supervised learning, i.e. learning to classify those created pseudo classes. Next, the outlierness of each image is measured by the proposed network uncertainty based outlier score. Finally, the joint score refinement with re-weighting and ensemble strategy can be used to further boost the OD performance of E^3*Outlier*.

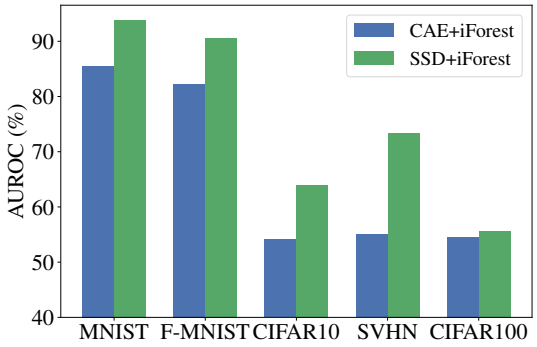


Fig. 3: Comparison of learned image representations.

those operations is to force DNN to capture the semantics of high-level structures in an image when it is required to fulfill such a classification task. For example, to recognize what type of rotation is imposed on the original image, the DNN must learn to localize salient object in images and recognize the orientation of its high-level parts, such as the head and legs of a human. Due to the page limit, we illustrate the details of data operation design in Sec. 1 of the supplementary material. Due to the prevalence of discriminative DNNs, creating pseudo classes by data operations is an intuitive and convenient way to provide self-supervision for deep OD. The overview of discriminative E^3*Outlier* is presented in Fig. 2. However, we will show other learning paradigms are also applicable to the proposed E^3*Outlier* later.

3.2.3 Comparison between SSD and AE

To verify whether SSD can learn better image representations, we conduct a simple experiment that compares SSD with Convolutional AE (CAE). We select WRN-28-10 [72] as SSD and adopt the CAE architecture in [57], which has a close depth to the SSD.Then, we extract the outputs of SSD's penultimate layer as learned representations, while the outputs of CAE's intermediate layer are extracted for comparison (note that they share the same dimension). With the protocol described in Sec. 4.1 to evaluate the OD performance on image datasets, learned representations of SSD and CAE are both fed into an Isolation Forest (IF) model with the same parameterization to conduct OD. The comparison is shown in Fig. 3: On those image benchmarks, learned representations of SSD are always able to improve IF's OD performance, which justifies SSD's effectiveness.

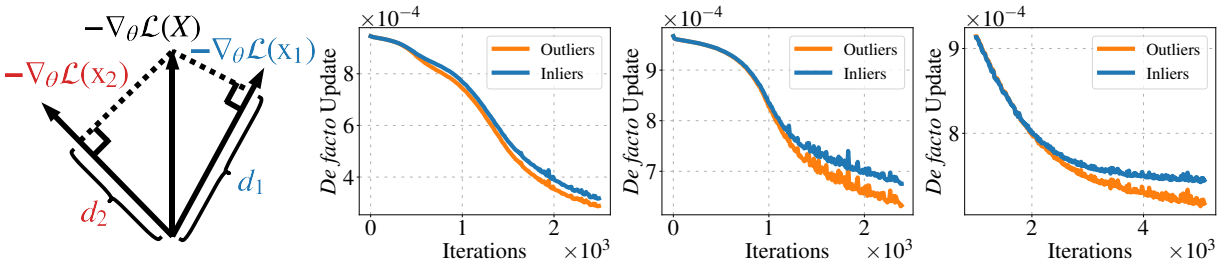
3.3 Inlier Priority: Foundation of End-to-end OD

3.3.1 Motivation

Although the proposed SSD achieves more effective representation learning than CAE, there are still some problems: First, without using a specialized OD network like [48], the proposed paradigm actually learns a pre-text task (i.e. classification) instead of OD, so by now we cannot draw OD results directly from SSD alone; Second, although we can resort to a classic OD model like we did in Sec. 3.2.3, such a two-stage solution can be sub-optimal as learned representations and the OD model are not jointly optimized. In fact, the OD performance of SSD+IF solution in Sec. 3.2.3 indeed has room for improvement (60%-70% AUROC) on relatively difficult benchmarks, i.e. CIFAR10/SVHN/CIFAR100. Therefore, an end-to-end solution is favorable for deep OD. However, for the proposed SSD, data operations are equally imposed on both inliers and outliers to create a pseudo class, and they are indiscriminately fed into DNN for training. Thus, it is still not sure whether inliers and outliers will behave differently during the self-supervised learning. This motivates us to explore this issue below from both theoretical and empirical view.

3.3.2 The Theoretical View

First of all, we approach this issue from a theoretical view. Since the theoretical analysis of DNNs remains particularly difficult, we consider a simplified case that is analyzable: We choose a feed-forward network with a single hidden layer and sigmoid activation to be SSD. Suppose that the hidden layer and Softmax layer have $(L+1)$ and K nodes respectively. Parameters of the simple SSD is randomly initialized by an i.i.d uniform distribution on $[-1, 1]$. Since neural networks are usually optimized by gradient descent, the influence of inliers and outliers imposed on the SSD can be reflected by the gradients that they back-propagate to update the network parameters. Hence, we



(a) *De facto* update (b) MNIST ("3") (c) Fashion-MNIST ("bag") (d) CIFAR10 ("horse")

Fig. 4: An illustration of *de facto* update and the average *de facto* update of inliers/outliers during the network training. The class used as inliers is in brackets.

analyze gradients w.r.t the weights associated with the c-th class ($1 \leq c \leq K$) between the hidden layer (it is also the penultimate layer in this case) and the final Softmax layer, $\mathbf{w}_c = [w_{s,c}]_{s=1}^{(L+1)}$ ($w_{L+1,c}$ is the bias), which are directly responsible for making SSD's predictions. We discuss the case of iniers (X_{in}) first: For the cross-entropy loss $\mathcal{L}$ that is used in our case, only those data yielded by imposing the c-th operation on X_{in} are used to update $\mathbf{w}_c$, i.e. $X_{in}^{(c)} = \{\mathbf{x}^{(c)} = O(\mathbf{x}|c)|\mathbf{x} \in X_{in}\}$. The gradient vector incurred by $X_{in}^{(c)}$ is denoted by $\nabla_{\mathbf{w}_c}\mathcal{L} = [\nabla_{w_{s,c}}\mathcal{L}]_{s=1}^{(L+1)}$, and each element of $\nabla_{w_{s,c}}\mathcal{L}$ is given by:

$$\nabla_{w_{s,c}}\mathcal{L} = \sum_{i=1}^{N_{in}} \nabla_{w_{s,c}}\mathcal{L}(\mathbf{x}_i) = \sum_{i=1}^{N_{in}} (P^{(c)}(\mathbf{x}_i) - 1)h^{(s)}(\mathbf{x}_i) \tag{3}$$

where $N_{in} = |X_{in}^{(c)}| = |X_{in}|$ is the number of inliers. For $\mathbf{x}_i \in X_{in}^{(c)}$, $P^{(c)}(\mathbf{x}_i)$ is the output of c-th node in the Softmax layer, and $h^{(s)}(\mathbf{x}_i)$ is the output of s-th node in the penultimate layer. To quantify inliers' influence on a randomly initialized SSD, a direct indicator can be the expectation of inliers' gradient magnitude to update $\mathbf{w}_c$, $E^{(in)}(||\nabla_{\mathbf{w}_c}\mathcal{L}||_2^2)$. Thus, our goal is to obtain:

$$E^{(in)}(||\nabla_{\mathbf{w}_c}\mathcal{L}||_2^2) = E\big(\sum_{s=1}^{L+1}(\nabla_{w_{s,c}}\mathcal{L})^2\big) = \sum_{s=1}^{L+1} E\big((\nabla_{w_{s,c}}\mathcal{L})^2\big) \tag{4}$$

By addition in (3), computing (4) requires the term below:

$$\begin{aligned} E\big((\nabla_{w_{s,c}}\mathcal{L})^2\big) &= E\big((\sum_{i=1}^{N_{in}} \nabla_{w_{s,c}}\mathcal{L}(\mathbf{x}_i))^2\big) \\ &= \sum_{i=1}^{N_{in}}\sum_{j=1}^{N_{in}} E(\nabla_{w_{s,c}}\mathcal{L}(\mathbf{x}_i)\nabla_{w_{s,c}}\mathcal{L}(\mathbf{x}_j)) \end{aligned} \tag{5}$$

To compute (5), in our case we can resort to the second-order Taylor series expansion to derive the approximation below (detailed in Sec. 2 of the supplementary material):

$$\begin{aligned} &E(\nabla_{w_{s,c}}\mathcal{L}(\mathbf{x}_i)\nabla_{w_{s,c}}\mathcal{L}(\mathbf{x}_j)) \approx \\ &h^{(s)}(\mathbf{x}_i)h^{(s)}(\mathbf{x}_j)\Big[\frac{(K-1)^2}{K^2} + \frac{K-1}{3K^3}\sum_{t=1}^{L+1} h^{(t)}(\mathbf{x}_i)h^{(t)}(\mathbf{x}_j)\Big] \end{aligned} \tag{6}$$

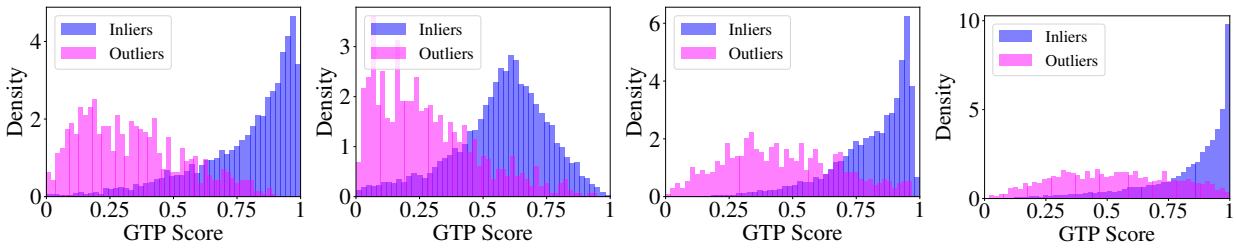
There remains to calculate $h^{(t)}(\mathbf{x}_i)h^{(t)}(\mathbf{x}_j)$ in (6). In this case, [73, Lemma 3.b] has proved that the expectation of $h^{(s)}(\mathbf{x}_i)h^{(s)}(\mathbf{x}_j)$ w.r.t the randomly initialized weights between the input and hidden layer satisfies $E(h^{(s)}(\mathbf{x}_i)h^{(s)}(\mathbf{x}_j)) \approx \frac{1}{4}$ and $E(h^{(s)}(\mathbf{x}_i)^2 h^{(s)}(\mathbf{x}_j)^2) \approx \frac{1}{16}$. Thus, by definition of $||\nabla_{\mathbf{w}_c}\mathcal{L}||_2^2$ in (4) and (5), we yield:

$$\begin{aligned} E^{(in)}(||\nabla_{\mathbf{w}_c}\mathcal{L}||_2^2) &\approx N_{in}^2\Big[(L+1)\big(\frac{(K-1)^2}{4K^2} + \frac{(K-1)(L+1)}{48K^3}\big)\Big] \\ &\triangleq N_{in}^2 \cdot Q \end{aligned} \tag{7}$$

Since L, K above are both fixed, Q is a constant. As a result, (7) shows that for the self-supervised learning of SSD, $E^{(in)}(||\nabla_{\mathbf{w}_c}\mathcal{L}||_2^2)$ is roughly proportional to N_{in}^2. Likewise, we can also derive that the expectation of outliers' gradient magnitude is $E^{(out)}(||\nabla_{\mathbf{w}_c}\mathcal{L}||_2^2) = N_{out}^2 \cdot Q$. Since $N_{in} \gg N_{out}$ is a indispensable premise for the OD task, we have $E^{(in)}(||\nabla_{\mathbf{w}_c}\mathcal{L}||_2^2) \gg E^{(out)}(||\nabla_{\mathbf{w}_c}\mathcal{L}||_2^2)$, which leads to an interesting conclusion: Although inliers and outliers are equally used for the self-supervised learning of SSD, the gradients contributed by inliers are much more important than outliers. Since those back-propagated gradients are used to train SSD, the theoretical analysis leads to an underlying property: *SSD is inclined to prioritize inliers during self-supervised learning*, which is named *inlier priority* in this paper. Such a property implies that inliers and outliers behave differently in self-supervised learning, which makes it possible to establish an end-to-end OD solution. Since it is intractable to compute $E(h^{(t)}(\mathbf{x}_i)h^{(t)}(\mathbf{x}_j))$ for more complex SSD, we will further validate inlier priority by empirical validations in the next section.

3.3.3 Empirical Validations

To further validate the property of inlier priority empirically, we propose to calculate a more direct indicator named "*de facto* update" for inliers and outliers respectively: In addition to gradient magnitude that we have considered in previous theoretical analysis, another important attribute of gradient vectors is gradient direction. As illustrated by Fig. 4a, consider $\mathbf{x}_i$ from a batch of data X (we slightly abuse the notation of X here). The negative gradient $-\nabla_{\boldsymbol{\theta}}\mathcal{L}(\mathbf{x}_i)$ is the fastest network updating direction to reduce $\mathbf{x}_i$'s loss. However, the network weights $\boldsymbol{\theta}$ are actually updated by the averaged negative gradient of the entire batch X, $-\nabla_{\boldsymbol{\theta}}\mathcal{L}(X) = -\frac{1}{N}\sum_i \nabla_{\boldsymbol{\theta}}\mathcal{L}(\mathbf{x}_i)$. Thus, the actual updating direction at each iteration is usually different from the best updating direction for each individual datum. To measure the actual gradient magnitude that $\mathbf{x}_i$ obtains along its best direction



(a) MNIST ("3") (b) Fashion-MNIST ("bag") (c) CIFAR10 ("horse") (d) SVHN ("3")

Fig. 5: Normalized histograms of inliers/outliers' $S_{gtp}(\mathbf{x})$. The class used as inliers is in brackets.

for loss reduction from $-\nabla_{\boldsymbol{\theta}}\mathcal{L}(X)$, we introduce the concept *de facto* update, which is computed by projecting $\nabla_{\boldsymbol{\theta}}\mathcal{L}(X)$ onto the direction of $\nabla_{\boldsymbol{\theta}}\mathcal{L}(\mathbf{x}_i)$: $d_i = \nabla_{\boldsymbol{\theta}}\mathcal{L}(X) \cdot \frac{\nabla_{\boldsymbol{\theta}}\mathcal{L}(\mathbf{x}_i)}{||\nabla_{\boldsymbol{\theta}}\mathcal{L}(\mathbf{x}_i)||}$. For example, as shown in Fig. 4a, the *de facto* update d_1 and d_2 reflect how much effort the network will devote to reduce the training loss of $\mathbf{x}_1$ and $\mathbf{x}_2$ respectively. *De facto* update can be viewed as an even more direct indicator of data's priority during training. In our case, we still take the gradients w.r.t. the weights between SSD's penultimate and softmax layer as an example. Under the setup in Sec. 4.1, we calculate the average *de facto* update for inliers and outliers respectively, and visualize typical results of *de facto* update on several image benchmarks in Fig. 4b-4d: As can be seen from the results, despite being close at the beginning, the average *de facto* update of inliers becomes evidently higher than outliers as the training continues, which justifies that SSD will bias towards inliers' best updating directions.

3.3.4 Baseline Outlier Score and Additional Remarks

Having illustrated inlier priority both theoretically and empirically, it can be expected that inliers are likely to achieve better training performance than outliers on a SSD after the self-supervised learning. In other words, SSD will prioritize reducing inliers' loss, which suggests that it is possible to discriminate outliers directly by each datum's loss value after training. To be more specific, for an image $\mathbf{x}^{(y)}$, we note that the calculation of its cross entropy loss only depends on its ground truth class probability $P^{(y)}(\mathbf{x}^{(y)}|\boldsymbol{\theta})$ that corresponds to its pseudo class label y. Thus, we propose Ground Truth Probability (GTP) score $S_{gtp}(\mathbf{x})$ that averages $P^{(y)}(\mathbf{x}^{(y)}|\boldsymbol{\theta})$ for all K operations to measure outlierness:

$$S_{gtp}(\mathbf{x}) = \frac{1}{K}\sum_{y=1}^{K}\mathbf{1}_y^{\top}\cdot\mathbf{P}(\mathbf{x}^{(y)}|\boldsymbol{\theta}) = \frac{1}{K}\sum_{y=1}^{K}P^{(y)}(\mathbf{x}^{(y)}|\boldsymbol{\theta}) \quad (8)$$

where $\mathbf{1}_y$ denotes the one-hot vector with the y-th element to be 1. To validate whether GTP score is a plausible way to measure outlierness, we calculate the $S_{gtp}(\mathbf{x})$ on image benchmarks and visualize the accumulated histograms for inliers and outliers respectively (note that histograms are normalized for better visualization). Representative results are shown in Fig. 5a-5d, and the score distributions of inliers and outliers are observed to be readily separable. Thus, GTP score can be a feasible baseline score for end-to-end OD. In addition, we would also like to point out the relation between inlier priority and representation learning: In deep OD task like outlier image removal, the difference between outliers and inliers lie in their semantics, e.g. high-level structure and appearance. To encourage the semantic similarity within inliers and maximize the semantic difference between inliers and outliers, it is necessary to learn good representations with rich semantics in the first place. Thus, a learning task that can yield semantically meaningful representations is the foundation for inliers to be semantically similar and joint their efforts into a priority against outliers.

3.4 Network Uncertainty As an Outlierness Measure

3.4.1 Motivation

SSD+GTP score provides a baseline end-to-end OD solution. However, it is imperfect and still has room for improvement, especially considering that the proposed self-supervised learning is not as precise as the classic supervised learning with human annotations: The data operation sometimes may not be able to transform the original image into an actual new one, e.g. a digit "8" is still itself after flipping is performed. Therefore, labels assigned to pseudo classes can be inaccurate. Since the calculation of GTP score in (8) relies on the pseudo class label y, such inaccurate labeling may undermine the GTP score's effectiveness to discriminate outliers. Motivated by this problem, we intend to design a new outlierness measure that is independent of pseudo class labels, so as to exploit the possibility to further improve end-to-end OD performance. Besides, when compared with other outlierness measures like density or proximity, uncertainty is usually directly optimized during the training of DNN, while other measures are not an explicit goal of the optimization. Therefore, we believe that network uncertainty can be a more direct indicator of inlier priority than other traditional measures. To this end, network uncertainty comes into our sight, since it is exactly an orthogonal attribute to DNN's classification accuracy [74]. As previous works basically discuss this concept in the context of DNN's prediction confidence, it is interesting to explore whether network uncertainty can be used for end-to-end OD.

3.4.2 A Demonstration Experiment

We carry out a simple demonstration experiment to shed light on this issue. For visualization, we generate 2D data with different degree of outlierness (detailed in Sec. 3 in supplementary material): The generated data (dots in Fig. 6) exhibit a larger dispersion as their coordinate on x-axis, x_i, gets more distant from the origin of x-axis, which enables data on two ends to show larger outlierness. To calculate

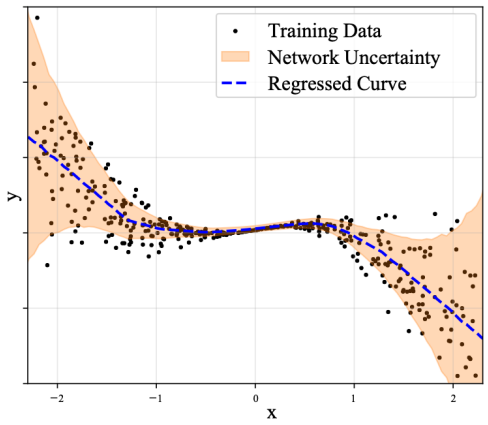


Fig. 6: The uncertainty of a regression network.

network uncertainty, we introduce a regression task that predicts y-axis coordinate y_i by corresponding x_i. Note that the regression task can be viewed as a self-supervised learning task, since we actually intend to infer the missing coordinate y_i by the incomplete data $\tilde{\mathbf{x}}_i = [x_i]$ like the masking mechanism [58]. The regression task is performed by training a simple neural network with the generated 2D data, and we estimate the uncertainty of neural network by the popular MC-Dropout method [64]. As it is shown in Fig. 6, it is easy to discover that the network uncertainty (highlighted orange region) is positively correlated to the outlierness of data. In other words, the experiment demonstrates some interesting connections among network uncertainty, OD and self-supervised learning: *The uncertainty of a neural network, which is trained to accomplish a self-supervised learning task (not OD itself), actually serves as a fairly effective way to measure data's outlierness.* Besides, it is also worth noting that network uncertainty is not relevant to the label y_i. This facilitates it to be more robust to label noises in self-supervised learning, just as we discussed in Sec. 3.4.1.

3.4.3 Network Uncertainty based Outlier Scores

As reviewed in Sec. 2.3, the uncertainty of DNN can be estimated by several ways, which can be categorized into Bayesian methods and non-Bayesian methods. Since Bayesian methods are usually more complicated and require more modifications to DNN itself, we focus on non-Bayesian methods when designing outlier scores. The following network uncertainty based scores are designed: **(1)** Maximum Probability (MP) score $S_{mp}(\mathbf{x})$. $S_{mp}(\mathbf{x})$ utilizes the maximum probability (i.e. prediction probability) output by the Softmax layer of SSD, which has proved to be a simple but strong baseline for uncertainty estimation [66], [67]:

$$S_{mp}(\mathbf{x}) = \frac{1}{K}\sum_{y=1}^{K} \max \mathbf{P}(\mathbf{x}^{(y)}|\boldsymbol{\theta}) = \frac{1}{K}\sum_{y=1}^{K} \max_t P^{(t)}(\mathbf{x}^{(y)}|\boldsymbol{\theta}) \quad (9)$$

(2) MC-Dropout (MCD) score $S_{mcd}(\mathbf{x})$. MC-Dropout keeps the dropout layers functional during inference, and calculates the first and second-order moment of DNN's outputs by several forward passes [64]. Since the maximum output probability and variance in DNN's outputs are both able to reflect DNN's uncertainty, we devise $S_{mcd}(\mathbf{x})$ as follows, so as to adapt it to OD task ($Mean(\cdot)$ and $Var(\cdot)$ refers to the mean and variance of multiple forward passes):

$$S_{mcd}(\mathbf{x}) = \frac{1}{K}\sum_{y=1}^{K} -Var(\max \mathbf{P}(\mathbf{x}^{(y)}|\boldsymbol{\theta})) + Mean(\max \mathbf{P}(\mathbf{x}^{(y)}|\boldsymbol{\theta})) \quad (10)$$

(3) Negative Entropy (NE) based score $S_{ne}(\mathbf{x})$. Information entropy (i.e. Shannon entropy) has constantly been used for measuring information and uncertainty embedded in data. Thus, we design $S_{ne}(\mathbf{x})$ to be computing the negative entropy of SSD's output probability distribution $\mathbf{P}(\mathbf{x}^{(y)}|\boldsymbol{\theta})$:

$$S_{ne}(\mathbf{x}) = \frac{1}{K}\sum_{y=1}^{K}\sum_{t=1}^{K} P^{(t)}(\mathbf{x}^{(y)}|\boldsymbol{\theta}) \log(P^{(t)}(\mathbf{x}^{(y)}|\boldsymbol{\theta})) \quad (11)$$

In addition to scores above, other network uncertainty based scores can also be explored. Our later evaluations show that network uncertainty based scores typically work better than the baseline outlier score S_{gtp}.

3.5 Score Refinement of Discriminative *E*^{3}*Outlier*

3.5.1 Motivation

Although components presented above have constituted a fully-functional end-to-end OD solution, it is still possible to improve discriminative *E*^{3}*Outlier*'s performance. As we have demonstrated how inlier priority and network uncertainty enable end-to-end OD, they should also be considered as the origin for performance improvement. Intuitively, a better OD performance essentially suggests that the priority of inliers is magnified, while it can also be accomplished by better uncertainty estimation. Inspired by such instincts, we propose two types of strategies to refine outlier scores.

3.5.2 Re-weighting Strategy

Our first instinct is to make SSD further prioritize inliers during training. Nevertheless, it is noted that inliers and outliers are indiscriminately fed into SSD at the very beginning of training, i.e. inliers and outliers are equally weighted by 1. Having revealed the role of inlier priority in OD, it is undoubted that this default initialization is not optimal: We can assign inliers with larger weights right before the beginning of SSD's training, which justifies the introduction of a re-weighting scheme. Since given data are completely unlabeled in OD, how and when to re-weight those unlabeled data for OD are key issues that we have to answer. As to how to re-weight, our solution is to utilize scores yielded by the proposed outlierness measure as weights, which have already achieved far better OD performance than existing methods. To be more specific, we can normalize scores into non-negative weights $w_1, \cdots w_N$ that satisfy $\sum_{i=1}^{N} w_i = 1$, and modify the objective function in (1) into the form below:

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} w_i \mathcal{L}_{SS}(\mathbf{x}_i|\boldsymbol{\theta}) \quad (12)$$

As for when to re-weight, since scores are only accessible after self-supervised learning begins, we can perform re-weighting during or after SSD's training. Accordingly, we propose *online* re-weighting and *reboot* re-weighting strategy: Online re-weighting strategy will update the weights

at the end of every epoch, and only one SSD is trained. By contrast, reboot re-weighting trains two SSD models: The first SSD is trained by a standard procedure, while the scores yielded by the first SSD are used as fixed weights to train the second SSD. The full algorithms are detailed in Algorithm 1 and Algorithm 2 in Sec. 4 of supplementary material. Our evaluations show that both algorithms can improve E^3*Outlier*'s performance.

3.5.3 Ensemble Strategy

In addition to the re-weighting strategy, another instinct is to improve uncertainty estimation for better OD performance. Since a generic strategy that can be easily embedded into the model is always preferred, we introduce the ensemble strategy into the score refinement stage. Ensemble is a widely-used technique in machine learning that combines multiple models into a stronger one. It is shown to be a powerful tool to improve the predictive performance [75], and recent works also demonstrate that an ensemble of DNNs can be highly efficient for producing good model uncertainty estimates [65], [67]. Specifically, we first create multiple SSD models $M_1, \cdots, M_e$ in a certain way, where $e > 1$ is the number of SSD models. For example, we can initialize SSD models with different random seeds, or adopt several different network architectures as different SSD models. After self-supervised learning, we simply average the outputs of different SSD models by $\bar{\mathbf{P}}(\mathbf{x}_i^{(y)}|\boldsymbol{\theta}) = \frac{1}{e}\sum_{j=1}^{e}\mathbf{P}_j(\mathbf{x}_i^{(y)}|\boldsymbol{\theta})$, where $\mathbf{P}_j(\mathbf{x}_i^{(y)}|\boldsymbol{\theta})$ is the outputs of j_{th} SSD model. Afterwards, we can calculate any network uncertainty based score with $\bar{\mathbf{P}}(\mathbf{x}_i^{(y)}|\boldsymbol{\theta})$. Note that the ensemble process can be readily paralleled for potential acceleration. Our later empirical evaluations show that such simple ensemble technique almost consistently improves the OD performance when compared with the case where a single SSD model is used.

3.5.4 Joint Score Refinement

Two aforementioned strategies are both able to yield better outlier scores, but it should be noted that they actually refine outlier scores from different views: The re-weighting strategy strengthens the inlier priority during self-supervised learning, while the ensemble strategy aims to improve the estimation of network uncertainty. In other words, two strategies exploit non-overlapping facets for score refinement. Thus, using a joint strategy of the re-weighting and ensemble to achieve even better OD performance is natural. In this paper, we devise the final score refinement stage by combining the reboot re-weighting strategy with the ensemble strategy (shown in Algorithm 3 in Sec. 4 of the supplementary material). Note that this is not the only form to combine re-weighting and ensemble, e.g. combining online re-weighting with the ensemble is also possible.

3.6 Other Learning Paradigms for E^3*Outlier*

In previous sections, we have demonstrated the way to leverage discriminative self-supervised learning to perform deep OD. As the way to introduce self-supervision is not limited to the discriminative learning paradigm, it is natural for us to explore other learning paradigms for E^3*Outlier*, which brings two benefits: First, more available learning paradigms enable E^3*Outlier* to be more flexible when dealing with different application scenarios. Second, emerging self-supervised learning paradigms like contrastive learning also facilitate E^3*Outlier* to further exploit its potential for deep OD. Thus, this section will detail our solution to apply generative and contrastive learning paradigms to E^3*Outlier*.

3.6.1 Generative E^3 Outlier

Generative learning paradigm is not new, because AE based reconstruction is exactly the most frequently-used method in existing deep OD solutions so far. However, as illustrated in Sec. 3.2.3, existing generative solutions often perform unsatisfactorily. As self-supervision is shown to be surprisingly effective in discriminative E^3*Outlier*, it is instinctive for us to explore *whether self-supervision can also improve the performance of generative deep OD*. Specifically, our solution is to add richer self-supervision information into the generation process to avoid simple reconstruction of the inputs. Inspired by the fact that data operations can provide rich self-supervision signal in SSD, we propose the generative self-supervised learning (GSS) paradigm below: Consider a data operation set with K_g operations $\mathcal{O}_g = \{O_g(\cdot|y)\}_{y=1}^{K_g}$. The data operations in $\mathcal{O}_g$ can be defined by various ways, such as certain transformations or fetching a specific part or modality of the input data. Then, we draw two different operations $O_g(\cdot|y_1)$ and $O_g(\cdot|y_2)$ from $\mathcal{O}_g$. Given an input data $\mathbf{x}$, two operations are required to satisfy:

$$O_g(\mathbf{x}|y_1) \neq O_g(\mathbf{x}|y_2), \qquad y_1 \neq y_2 \tag{13}$$

Then, a generative DNN $\mathcal{G}$ (e.g. AE, UNet [76] or GANs) is trained to generate $O_g(\mathbf{x}|y_2)$ by taking $O_g(\mathbf{x}|y_1)$ as the input, which is equivalent to minimizing the objective below:

$$\mathcal{L}_{GSS}(y_1, y_2) = \frac{1}{N}\sum_{i=1}^{N}||\mathcal{G}(O_g(\mathbf{x}_i|y_1)) - O_g(\mathbf{x}_i|y_2)||_2^2 \tag{14}$$

It is easy to note that when Eq. (13) is not satisfied, Eq. (14) will degrade into plain reconstruction. When $\mathcal{G}$ has been trained, one can simply obtain an outlier score of $\mathbf{x}$ based on the MSE loss of generation:

$$S_g(\mathbf{x}|y_1, y_2) = -||\mathcal{G}(O_g(\mathbf{x}|y_1)) - O_g(\mathbf{x}|y_2)||_2^2 \tag{15}$$

Since there exist different ways to select operations, it is natural to train the model and compute final outlier score by a combination of different y_1, y_2 configurations:

$$\begin{aligned} \mathcal{L}_{GSS} &= \sum_{y_1}\sum_{y_2}\mathcal{L}_{GSS}(y_1, y_2), \\ S_g(\mathbf{x}) &= \sum_{y_1}\sum_{y_2}S_g(\mathbf{x}|y_1, y_2) \end{aligned} \tag{16}$$

Compared with the plain reconstruction adopted by AE based deep OD methods, the key to our generative E^3*Outlier* is to make DNN generate a different datum obtained by a non-identical operation, which makes the learning task more challenging for DNNs. This not only avoids the DNN to simply memorize the low-level details, but also encourages the DNN to consider high-level semantics by learning the correlations of two different data, which can be

viewed as valuable self-supervision information. Our later evaluations show that generative E^3*Outlier* can produce tangible performance improvement when it shares the same generative DNN with other reconstruction based deep OD solutions. More importantly, generative E^3*Outlier* can be readily applied to some important scenarios where the input data can be decomposed into multiple views or modalities. For example, video data are usually considered from the view of both appearance and motion. In those cases, the correspondence between different data views/modalities is valuable self-supervision signal in itself, and generative E^3*Outlier* provides a convenient and straightforward way to exploit such semantics. As a demonstration, we will show how to design a new unsupervised video abnormal event detection solution by generative E^3*Outlier* in Sec. 4.3.2.

3.6.2 Contrastive E^3 Outlier

It is easy to notice that the performance of current deep OD solutions, including the proposed discriminative E^3*Outlier*, suffers from evidently inferior performance on colored image datasets (e.g. CIFAR10) when compared with comparatively simple gray-scale image datasets (e.g. MNIST). Meanwhile, we also note that color based operations (e.g. color jittering and RGB-to-gray transformation) play an important role in many vision tasks. To further exploit color information and enhance the capability to handle more ubiquitous colored images in practical applications, we leverage the emerging contrastive learning paradigm, which is shown to be highly effective in unsupervised representation learning of real-world colored images, to provide self-supervision in deep OD and design contrastive E^3*Outlier*. The core idea of contrastive learning is to learn meaningful representations by making DNNs compare a pair of data drawn from the unlabeled dataset. We choose one of the most representative contrastive learning method, SimCLR [77], as the foundation for the proposed contrastive E^3*Outlier*. Specifically, a contrastive loss for a datum $\mathbf{x}$ is defined as follows:

$$\mathcal{L}_{cl}(\mathbf{x}, X^+, X^-) = \\ -\frac{1}{|X^+|}\log\frac{\sum_{\mathbf{x}'\in X^+}\exp(sim(z(\mathbf{x}), z(\mathbf{x}'))/\tau)}{\sum_{\mathbf{x}'\in X^+\cup X^-}\exp(sim(z(\mathbf{x}), z(\mathbf{x}'))/\tau)} \tag{17}$$

where X^+/X^- denote the set with data that can form a positive/negative pair with $\mathbf{x}$, and $sim(\cdot,\cdot)$ is a similarity measure like cosine similarity. $|\cdot|$ is the cardinality of the set, and $z(\mathbf{x})$ is the projection yielded by feeding DNN's learned representation $f(\mathbf{x})$ into a projection layer $g(\cdot)$: $z(\mathbf{x}) = g(f(\mathbf{x}))$. τ is a hyperparameter. Next, the issue is to construct positive and negative data pairs to enable the calculation of Eq. (17). To this end, we introduce a random augmentation set $\mathcal{A}$, which contains augmentation operations that is composed of color jittering, RGB-to-gray transformation and image crop with random parameterization. Each time two independent random augmentation A_1 and A_2 are drawn from $\mathcal{A}$. After that, the data pair of augmented data $A_1(\mathbf{x})$ and $A_2(\mathbf{x})$ are viewed as a positive pair, while any other pair is viewed as negative. The goal of contrastive loss defined in Eq. (17) is to yield similar representations for a positive data pair, and make representations of a negative pair dissimilar. Given a mini-batch data set B drawn from the unlabeled dataset, SimCLR defined the following training objective to perform contrastive learning:

$$\mathcal{L}_{scl}(B, A_1, A_2) = \frac{1}{2|B|}\sum_{i=1}^{|B|}(\mathcal{L}_{cl}(A_1(\mathbf{x}_i), \{A_2(\mathbf{x}_i)\}, \hat{B}_{-i}) + \\ \mathcal{L}_{cl}(A_2(\mathbf{x}_i), \{A_1(\mathbf{x}_i)\}, \hat{B}_{-i})) \tag{18}$$

where we define $\hat{B}_{-i} = \{A_1(\mathbf{x}_j)\}_{j\neq i}\cup\{A_2(\mathbf{x}_j)\}_{j\neq i}$. Some recent works [77], [78] point out that some data operations (e.g. 90 degree rotation) can be used to generate negative pairs as they produce very different data from the original one. This is also verified in discriminative E^3*Outlier*, since those data operations are often likely to produce pseudo classes that are readily separable. Following such an observation, we collect an operation set $\mathcal{O}_c = \{O_c(\cdot|y)\}_{y=1}^{K_c}$ with K_c operations (including one identity transformation), and expand the mini-batch B into $B' = O_c(B|1)\cup\cdots\cup O_c(B|K_c)$, where the data set $O_c(B|y) = \{O_c(\mathbf{x}|y)|\mathbf{x}\in B\}$. Since B' can be viewed as a data set with K_c pseudo classes and discriminative E^3*Outlier* works well in deep OD, we substitute B by B' into Eq. (18) for training, and make DNN learn to classify those pseudo classes by an additional discriminative module and the cross-entropy loss $\mathcal{L}_{cls}(B')$, so as to produce more meaningful representations. In this way, the contrastive self-supervised learning (CSS) of E^3*Outlier* can be performed by the joint loss below:

$$\mathcal{L}_{CSS} = \mathcal{L}_{scl}(B', A_1, A_2) + \mathcal{L}_{cls}(B') \tag{19}$$

After training, we design a simple but effective outlier score based on inner product of learned representations: For the datum $\mathbf{x}_i^{(y)} = O_c(\mathbf{x}_i|y)$ obtained by imposing the y-th operation in $\mathcal{O}_c$ on $\mathbf{x}_i$, its outlier score $S_c(\mathbf{x}_i^{(y)})$ is given by:

$$S_c(\mathbf{x}_i^{(y)}) = \frac{1}{Z_{scl}^{(y)}}\max_{j\neq i} f^\top(\mathbf{x}_i^{(y)})\cdot f(\mathbf{x}_j^{(y)}) \tag{20}$$

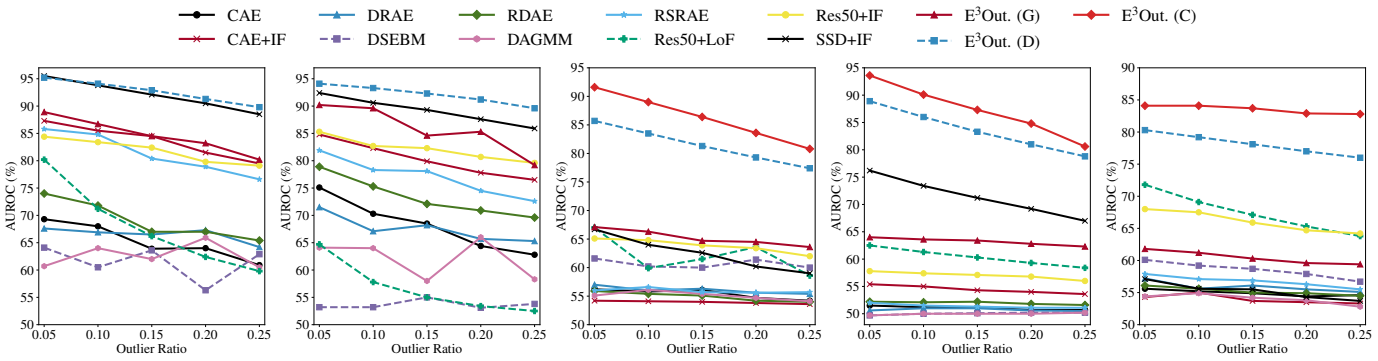
where $Z_{scl}^{(y)}$ is the normalization term computed as follows:

$$Z_{scl}^{(y)} = (\frac{1}{N}\sum_{i=1}^{N}||f(\mathbf{x}_i^{(y)})||)^{-1} \tag{21}$$

In Eq. (20), the score actually computes the maximum inner product between the learned representations of $\mathbf{x}_i^{(y)}$ and other data yielded by operation $O(\cdot|y)$, so as to measure how similar $\mathbf{x}_i^{(y)}$ is to the rest of data. With multiple operations in $\mathcal{O}_c$, the final outlier score can be computed by:

$$S_c(\mathbf{x}_i) = \sum_{y=1}^{K_c} S_c(\mathbf{x}_i^{(y)}) \tag{22}$$

Just like that contrastive learning paradigm significantly improves the performance of self-supervised learning, our later empirical evaluations show that contrastive E^3*Outlier* also advances the deep OD performance by a notable margin on those colored datasets that are relatively difficult for previous generative and discriminative E^3*Outlier*. As a summary, by designing generative learning and contrastive learning based solutions, we enable E^3*Outlier* to be a more flexible and stronger deep OD framework.



(a) MNIST (b) Fashion-MNIST (c) CIFAR10 (d) SVHN (e) CIFAR100

Fig. 7: AUROC comparison of OD methods under different outlier ratios.

4 Experiments

4.1 Experimental Settings

4.1.1 Benchmark Datasets and Evaluation

To validate the effectiveness of the proposed framework, we conduct extensive experiments on five frequently-used public image benchmarks: MNIST (MST) [79], Fashion-MNIST (FMST) [80], CIFAR10 (C10) [81], SVHN (SH) [82], CIFAR100 (C100) [81]. We follow the standard procedure, which is shared by previous image outlier removal works like [8], [9], [46], to construct a noisy image set with outliers: Given a standard image benchmark, all images from a class with one common semantic concept (e.g. "horse", "bag") are retrieved as inliers, while outliers are randomly sampled from the rest of classes by an outlier ratio ρ. We vary ρ from 5% to 25% by a step of 5%. The assigned inlier/outlier labels are strictly unknown to OD methods and only used for evaluation. Each class of a benchmark is used as inliers in turn, and the performance on all classes is averaged as the overall OD performance on this benchmark dataset. Since all images are viewed as unlabeled in OD, we do not use the split of train/test set and merge them for experiments. Note that for CIFAR100 dataset, we uses 20 superclasses instead of the original 100 classes to ensure that the constructed noisy image set contains sufficient data for DNN's training, and it can also test the OD performance when inliers have multiple subclasses (each superclass in CIFAR100 contains 5 classes). All experiments are repeated for 5 times with different random seeds, so as to yield the average results. Raw pixels are directly used as inputs with their intensity normalized into $[-1, 1]$. As for evaluation, we adopt the commonly-used Area under the Receiver Operating Characteristic curve (AUROC) and Area under the Precision-Recall curve (AUPR) as threshold-independent metrics [83].

4.1.2 Compared Methods

We extensively compare generative *E^3Outlier* (E^3Out. (G)), discriminative E^3Outlier (*E^3Out.* (D)) and contrastive *E^3Outlier* (E^3Out. (C)) with baselines and existing state-of-the-art DNN based OD methods in literature: **(1)** Convolutional Auto-Encoder (CAE) [84]. CAE is the most prevalent DNN type to deal with image data in many unsupervised learning tasks. Here it serves as an end-to-end baseline, which directly uses CAE's reconstruction loss to perform deep outlier removal. **(2)** CAE+Isolation Forest (CAE+IF). IF [40] is a classic OD method with wide popularity, so we combine it with CAE as the baseline of two-stage OD approaches. Specifically, CAE+IF feeds CAE's learned representations from its intermediate hidden layer into IF to perform OD. **(3)** SSD+IF. It shares *E^3Outlier*'s SSD part but feeds SSD's learned representations into an IF model to perform OD. SSD+IF serves as a two-stage baseline to compare against the proposed end-to-end *E^3Outlier*. **(4)** Discriminative Reconstruction based Auto-Encoder (DRAE) [9]. DRAE discriminates outliers by thresholding CAE's reconstruction loss with a self-adaptive scheme, which is in turn integrated into the loss function to refine the outlier removal performance. **(5)** Deep Structured Energy based Models (DSEBM) [45]. DSEBM uses an energy based function and score matching technique to estimate the probability that a datum fits the data distribution. **(6)** Robust Deep Auto-Encoder (RDAE) [46]. RDAE synthesizes CAE and RPCA, and it iteratively decomposes unlabeled data into a low-rank part and a sparse error part for outlier removal. **(7)** Deep Auto-encoding Gaussian Mixture Model (DAGMM) [48]. DAGMM embeds a GMM parameter estimation network into CAE, which realizes end-to-end OD by performing representation learning and fitting a GMM simultaneously. **(8)** Multiple-Objective Generative Adversarial Active Learning (MOGAAL) [50]. MOGAAL attempts to generate pseudo outliers that are distributed around given unlabeled data with modified GANs and active learning, so as to transform OD into a supervised binary classification problem. **(9)** Robust Subspace Recovery based AE (RSRAE) [52]. RSRAE is the latest method that improves OD performance by learning to recover the underlying data manifold in a subspace while performing AE's reconstruction. For RSRAE, the reconstruction loss and RSR loss are optimized in a seprated manner. In addition to deep solutions, we also include the following baseline solutions for a more comprehensive comparison: **(10)** Two-stage solutions based on pre-trained DNN and the classic OD model. DNN models pre-trained on large-scale generic datasets prove to be an effective tool for feature extraction. Thus, to design a two-stage solution, we use a ResNet50 model pre-trained on ImageNet dataset as feature extractor, and the extracted features are then fed into a classic OD model. IF and the classic Local Outlier Factor (LoF) are exploited here. Due to

TABLE 1: OD performance comparison (in %) in terms of AUROC (Area Under ROC curve, shorted as ROC), AUPR-In (Area under PR curve with inliers to be the positive class, shorted as PR-I) and AUPR-Out (Area under PR curve with outliers to be the positive class, shorted as PR-O). Each benchmark shows the case where $\rho = 10\%$ and $\rho = 20\%$. Note that contrastive E^3*Outlier* is only used for benchmark datasets with colored images (CIFAR10/SVHN/CIFAR100), and the raw performance without score refinement is compared for fairness. The best performer is shown in bold font.

Dataset	MNIST			Fashion-MNIST			CIFAR10			SVHN			CIFAR100		
	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O
$\rho = 10\%$															
CAE	68.0	92.0	32.9	70.3	94.3	29.3	55.8	91.0	14.4	51.2	90.3	10.6	55.2	91.0	14.5
CAE+IF	85.5	97.8	49.0	82.3	97.2	40.3	54.1	90.2	13.7	55.0	91.4	11.9	55.0	90.7	13.8
DRAE	66.9	93.0	30.5	67.1	93.9	25.5	56.0	90.7	14.7	51.0	90.3	10.5	55.6	90.9	15.0
DSEBM	60.5	91.6	23.0	53.2	88.9	19.7	60.2	92.3	14.7	50.0	90.0	10.1	59.2	92.2	16.2
RDAE	71.8	93.1	35.8	75.3	95.8	31.7	55.4	90.7	14.9	52.1	90.6	10.8	55.6	90.9	15.0
DAGMM	64.0	92.9	26.6	64.0	92.7	30.3	56.1	91.3	15.6	50.0	90.0	19.3	54.9	91.1	14.2
MOGAAL	30.9	78.8	15.2	22.8	74.8	14.8	56.2	91.1	13.6	49.0	89.7	9.8	53.2	90.4	12.6
RSRAE	84.8	97.4	45.4	78.3	96.2	37.0	56.6	91.4	14.0	51.5	90.3	10.6	57.1	91.6	14.1
Res50+LoF	71.2	97.5	26.6	57.8	96.2	16.9	59.9	91.4	17.4	61.3	90.3	14.0	69.1	94.6	22.2
Res50+IF	83.4	97.5	43.3	82.7	97.3	43.8	64.8	93.8	17.9	57.4	92.0	12.8	67.5	94.3	21.0
SSD+IF	93.8	99.2	**68.7**	90.6	98.5	68.6	64.0	93.5	18.3	73.4	95.9	22.0	55.6	91.5	13.0
E^3Out. (G)	86.7	96.4	60.3	89.6	98.5	61.6	66.3	93.5	20.0	63.6	93.9	15.0	61.2	92.4	16.7
E^3Out. (D)	**94.1**	**99.3**	67.5	**93.3**	**99.0**	**75.9**	83.5	97.5	43.4	86.0	98.0	36.7	79.2	96.8	33.3
E^3Out. (C)	-	-	-	-	-	-	**89.0**	**98.5**	**53.2**	**90.1**	**98.5**	**51.3**	**84.1**	**97.8**	**38.0**
$\rho = 20\%$															
CAE	64.0	82.7	40.7	64.4	85.3	36.8	54.7	81.6	25.5	50.7	80.2	20.7	54.4	81.7	25.6
CAE+IF	81.5	93.6	57.2	77.8	92.2	49.0	53.8	80.7	25.3	54.0	82.0	22.4	53.5	80.9	25.1
DRAE	67.3	86.6	42.5	65.7	86.9	36.6	55.6	81.7	26.8	50.6	80.4	20.5	55.5	81.8	27.0
DSEBM	56.3	81.2	32.3	53.1	79.6	31.7	61.4	85.2	27.8	50.2	80.3	20.2	57.9	83.7	27.8
RDAE	67.0	89.2	43.2	70.9	89.2	41.4	54.2	81.0	25.7	51.8	80.9	21.1	54.9	81.5	26.5
DAGMM	65.9	86.7	41.3	66.0	86.7	43.5	54.7	81.8	26.3	50.0	79.9	29.6	53.8	81.5	24.7
MOGAAL	37.8	70.6	28.0	34.0	66.6	28.3	55.7	82.0	25.0	49.6	79.8	19.8	53.1	80.9	24.4
RSRAE	78.9	91.3	53.0	74.5	90.4	46.3	55.6	82.1	25.8	51.1	80.3	21.0	56.3	82.7	25.2
Res50+LoF	62.4	84.9	31.0	53.4	80.3	24.9	63.6	84.9	27.9	59.3	85.0	25.2	65.3	87.5	32.6
Res50+IF	79.8	93.6	52.1	80.7	93.5	55.0	63.4	86.6	30.4	56.8	83.3	24.2	64.7	87.1	32.4
SSD+IF	90.5	97.3	71.0	87.6	95.6	71.4	60.2	85.0	28.3	69.2	89.5	33.7	54.3	82.1	23.4
E^3Out. (G)	83.2	90.4	67.9	85.3	95.2	66.4	64.5	85.7	33.0	62.8	86.8	27.9	59.6	83.8	28.6
E^3Out. (D)	**91.3**	**97.6**	**72.3**	**91.2**	**97.1**	**78.9**	79.3	93.1	52.7	81.0	93.4	47.0	77.0	92.4	46.5
E^3Out. (C)	-	-	-	-	-	-	**83.6**	**94.8**	**59.0**	**84.8**	**94.9**	**57.6**	**82.9**	**95.1**	**53.0**

page limit, implementation details are provided in Sec. 5 of the supplementary material. All of our codes and results can be verified at https://github.com/demonzyj56/E3Outlier.

4.2 Experimental Results

4.2.1 *Raw OD Performance Comparison*

Due to the space limit, we report numerical results under $\rho = 10\%$ and 20% in Table 1, while the AUROC comparison under different outlier ratios are shown in Fig. 7. From those results, we can obtain the following observations: **(1)** First of all, the proposed E^3*Outlier* framework possesses an evident advantage against existing state-of-the-art DNN based OD methods and baselines in terms of all evaluation metrics. Taking discriminative E^3*Outlier* as an example, it outperforms the best performer among state-of-the-art DNN based OD methods and baselines by a considerable 8%-20% AUROC on different benchmark datasets. In particular, it has realized a performance leap on CIFAR10, SVHN and CIFAR100, which are generally acknowledged to be challenging benchmarks for unsupervised learning tasks like deep outlier removal or clustering. Meanwhile, with the same CAE as backbone, the proposed generative E^3*Outlier* is able to achieve evidently superior performance to existing CAE based deep OD solutions. Specifically, although it is inferior to its discriminative and contrastive counterparts, generative E^3*Outlier* consistently outperforms all AE based deep OD solutions in terms of AUROC, while it also yields comparable or better AUPR-In and AUPR-Out performance. Such improvement further justifies the effectiveness of introducing richer self-supervision information, and in later sections we show that generative E^3*Outlier* also enables us to flexibly handle other deep OD applications. Next, the proposed contrastive E^3*Outlier* is able to produce a significant performance gain (about 4%-6% AUROC) on colored datasets (CIFAR10/SVHN/CIFAR100) that are relatively difficult for its discriminative and generative counterparts, and it suggests that the potential of E^3*Outlier* can be further exploited by introducing more advanced self-supervised learning paradigms. Thus, the above observations have justified E^3*Outlier* as a highly effective framework for DNN based OD. **(2)** Second, we notice that the baseline OD solutions that combine the classic OD model and features extracted from pre-trained ResNet50 model (Res50+LoF and Res50+IF) can indeed produce better performance than previous end-to-end OD solutions in many cases, which verifies the importance of the good representation. However, there is still a large performance gap between such two-stage solutions and the proposed deep OD framework, especially discriminative and contrastive E^3*Outlier*. Thus, it further

TABLE 2: Performance of discriminative E^3*Outlier* (in %) before and after joint score refinement (JSR) in terms of Area Under ROC curve, PR curve with inliers to be the positive class (PR-I) and PR curve with outliers to be the positive class (PR-O). Each benchmark shows the case where $\rho = 10\%$ and $\rho = 20\%$ due to the space limit.

Dataset	MNIST			Fashion-MNIST			CIFAR10			SVHN			CIFAR100		
	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O
$\rho = 10\%$															
E^3Out.	94.1	99.3	67.5	93.3	**99.0**	75.9	83.5	97.5	43.4	86.0	98.0	36.7	79.2	96.8	33.3
E^3Out.+JSR	**94.9**	**99.4**	**71.0**	**93.5**	**99.0**	**77.2**	**84.7**	**97.7**	**45.7**	**87.1**	**98.2**	**37.7**	**81.3**	**97.2**	**37.0**
$\rho = 20\%$															
E^3Out.	91.3	97.6	72.3	91.2	97.1	78.9	79.3	93.1	52.7	81.0	93.4	47.0	77.0	92.4	46.5
E^3Out.+JSR	**92.9**	**98.1**	**76.3**	**92.1**	**97.4**	**81.9**	**80.3**	**93.5**	**54.5**	**82.0**	**94.2**	**47.9**	**79.1**	**93.1**	**49.9**

demonstrates the effectiveness of the proposed deep OD framework. **(3)** Third, it is interesting to note that two-stage OD approaches can be more effective than previous end-to-end OD approaches. Specifically, the two-stage counterpart of discriminative E^3*Outlier* SSD+IF achieves fairly close performance to discriminative E^3*Outlier* on relatively simple gray-scale image datasets (MNIST/Fashion-MNIST). Meanwhile, CAE based end-to-end OD solutions (DRAE/DSEBM/DAGMM/RSRAE) cannot constantly outperform their two-stage counterparts (CAE+IF/RDAE), and CAE+IF even performs much better than some CAE based end-to-end solutions on MNIST/Fashion-MNIST. Nevertheless, as shown in Fig. 7a-Fig. 7e, the proposed discriminative E^3*Outlier* almost defeats its two-stage baseline SSD+IF in all experiments, and it suffers from evidently worse performance (i.e. over 10% AUROC loss) on difficult datasets like CIFAR10/SVHN/CIFAR100. **(4)** Among existing end-to-end OD methods, we notice that although recent end-to-end DNN based OD methods (RSRAE) are indeed making progress on relatively simple benchmarks like MNIST and Fashion-MNIST, their performance on difficult datasets like CIFAR10 is still as unsatisfactory as previous counterparts. Besides, MOGAAL performs poorly in almost all cases, which suggests that generating proper pseudo outliers are s till very difficult for deep OD by now.

4.2.2 Score Refinement

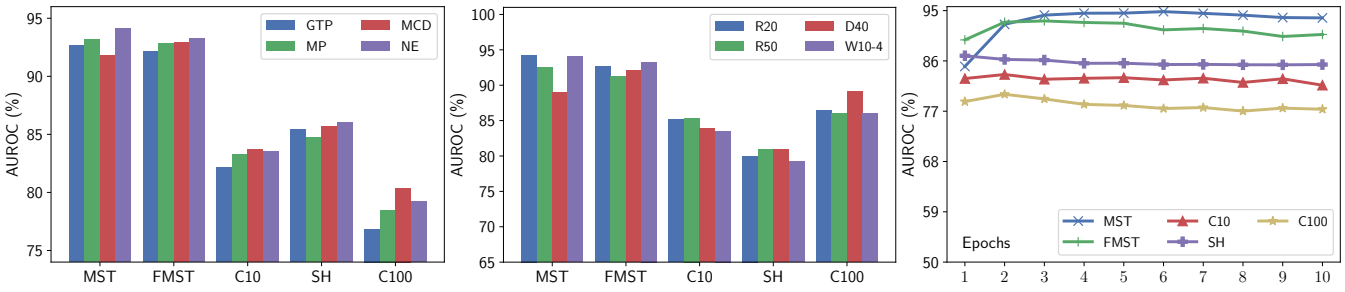
In this section, we validate the effectiveness of score refinement for discriminative E^3*Outlier*. As shown in Table 2, JSR enables consistent performance improvement under different outlier ratios and all evaluation metrics. To show the effect of each score refinement strategy, we further compare the OD performance of five cases in terms of AUROC: Baseline using no score refinement (BAS), using the online re-weighting strategy only (ORW), with the reboot re-weighting strategy only (RRW), using the ensemble strategy only (ENS) and using the joint score refinement (JSR), under $\rho = 10\%$ with default NE score for discriminative E^3*Outlier*. We report the results in Table 3, from which the following facts are drawn: First, when compared with the baseline (BAS), score refinement strategies are able to produce performance gain on all benchmarks by up to 2.1% AUROC gain. The improvement tends to be more tangible on comparatively difficult benchmarks like CIFAR100. Besides, under other outlier ratios, using score refinement also produces stable performance improvement (1% to 2% AUROC) on difficult benchmarks. Second, RRW tends to be slightly better than ORW, while ORW enjoys lower computational cost. Finally, the joint score refinement (JSR) with both reboot re-weighting and ensemble is typically better than a single score refinement strategy, except for the case Fashion-MNIST where JSR performs comparably to other refinement strategies. We also discuss the parameters in score refinement in Sec. 4 of supplementary material.

TABLE 3: comparison of score refinement strategies (in %).

CONFIG.	MST	FMST	C10	SH	C100
BAS	94.1	93.3	83.5	86.0	79.2
BAS+ORW	94.4	93.6	84.1	86.7	80.3
BAS+RRW	94.6	93.6	84.4	86.5	80.5
BAS+ENS	94.3	93.4	84.1	86.7	80.7
BAS+JSR	94.9	93.5	84.7	87.1	81.3

4.2.3 Discussion

In this section, we discuss several key factors in E^3*Outlier*. Similarly, we conduct experiments under $\rho = 10\%$ to show the general trends. We investigate the following factors of discriminative E^3*Outlier*: **(1)** Outlier scores: We compare four different outlier scores for discriminative E^3*Outlier*, i.e. GTP/MP/MCD/NE. As shown by Fig. 8a, uncertainty based scores (MP/MCD/NE) basically prevail over the baseline GTP score, which validates the advantages of exploring network uncertainty as outlierness measure for E^3*Outlier*. Among uncertainty based outlier scores, MCD and NE are prone to outperform the simplest MP. Although MCD achieves the best performance on some benchmarks, it requires multiple forward passes and tends to be less efficient than NE. By contrast, NE consistently outperforms the baseline by a notable margin, and it realizes a good trade-off between performance and efficiency. **(2)** The network architecture of SSD: With other settings fixed, we additionally explore ResNet20/ResNet50 [19] and DenseNet40 [85] as the backbone architecture for SSD (shown in Fig. 8b). Despite of some differences, those frequently-used architectures basically perform satisfactorily. Interestingly, we note that a more complex architecture (ResNet50/DenseNet40) tends to be more effective on relatively complex datasets (CIFAR10, SVHN and CIFAR100), but its performance is inferior on simpler datasets. **(3)** Training epochs (see Fig. 8c): We measure the OD performance when the SSD is trained by different epoch numbers to evaluate its impact on self-supervised learning. In general, we notice that the OD performance is



(a) Different outlier scores. (b) Different network architectures. (c) Performance with different training epochs.

Fig. 8: Different factors' influence on E^3*Outlier*'s performance under $\rho = 10\%$.

inclined to be improved at the initial stage of training (less than $\lceil \frac{250}{K} \rceil$ training epochs) and then reach a plateau. No drastic performance changes are observed as the training epochs continue to increase. **(4)** Pseudo label design. Since the operation set is often constructed by a composite of multiple types of transformations, it is natural to consider a multi-label way to assign pseudo labels. To explore its possibility, we assign each transformed datum with 5 labels based on the performed transformations: Simple rotation label (4 classes in total), translation label ($3 \times 3 = 9$ classes in total), irregular rotation label (8+1=9 classes in total), flip label (2 classes in total) and patch re-arranging label (23+1=24 classes in total). The DNN is equiped with 5 classification heads to predict 5 labels, while the outlier score is computed by averaging the outlier scores yielded by 5 heads. We report the performance of such a multi-label setup in Table 4, and the results suggest that it can yield slightly better performance on most benchmark datasets. Thus, it is possible to explore a more effective design of pseudo labels for E^3*Outlier*. For generative and contrastive E^3*Outlier*, we investigate two major factors: **(1)** Backbone architecture for generative E^3*Outlier*. In fact, one can explore different backbone architecture to implement the generative DNN $\mathcal{G}$ for generative E^3*Outlier*, and we test UNet as an example. As shown in Table 5, the results suggest that UNet is also able to yield fairly satisfactory OD performance, and we notice that UNet performs evidently better than CAE on relatively difficult datasets CIFAR10/SVHN/CIFAR100, while CAE tends to be better on simpler MNIST/Fashion-MNIST. **(2)** Classification loss $\mathcal{L}_{cls}$ for contrastive E^3*Outlier*. It is noted that the loss of classification $\mathcal{L}_{cls}$ when training the DNN model of contrastive E^3*Outlier*, and we also discuss the case where only the contrastive loss $\mathcal{L}_{scl}$ is applied. Interestingly, contrastive E^3*Outlier* without $\mathcal{L}_{cls}$ yields sigificantly worse performance on CIFAR10/CIFAR100 (77.3%/76.6% AUROC under $\rho = 10\%$), but the performance is better on SVHN (91.7% AUROC under $\rho = 10\%$). The reason is that the performance on "0" class of SVHN suffers from a drastic degradation when classification is performed, as "0" is still a "0" aften a rotation of 90, 180 or 270 degrees. Thus, the classification task is completely invalid in this case.

4.3 E^3*Outlier* based Video Abnormal Event Detection

4.3.1 Unsupervised Video Abnormal Event Detection

Inspired by E^3*Outlier*'s success with images, it is natural to explore E^3*Outlier* for other type of visual data, e.g. videos. To this end, unsupervised video abnormal event detection (UVAD) [10] is exactly an application of deep OD to videos. UVAD is an emerging task that aims to detect those unusual events that divert from other frequently-encountered routine in completely unlabeled video sequences. As it does not require labeling and enumerating normal video events to construct a training set, UVAD is more challenging than semi-supervised VAD that has been thoroughly studied [86]. Most existing UVAD solutions approach UVAD by change detection and its variants [10], [87], [88], while the recent work [89] also proposes a different solution that first initializes the detection results based on IF and pre-trained DNNs, and then refines the detection iteratively. However, existing UVAD solutions typically perform unsatisfactorily.

4.3.2 Design of E^3*Outlier* based UVAD Solution

Before we tailor the E^3*Outlier* for UVAD, we notice two important differences between UVAD and previous outlier image removal task: First, despite that discriminative and contrastive E^3*Outlier* are shown to be highly effective in detecting outlier images by appearance information (e.g. structure and texture), normal and abnormal video events are often conducted by the same type of subjects in UVAD (For example, humans in Fig. 9). In other words, appearance differences are less important to UVAD. Second, unlike static images, videos are described by both appearance and motion information. As motion is the key to detecting many abnormal events, optical flow maps of video frames are often computed to describe the motion in videos. Therefore, both raw video frames and optical flow maps are supposed to be exploited for providing self-supervision. Due to those differences, we naturally turn to generative E^3*Outlier* to connect both appearance and motion view. Based on generative E^3*Outlier*, the designed UVAD solution is presented below:

First of all, we follow our previous work [90] to extract and represent video events: Foreground objects in each video frame are first localized by a series of regions of interest (RoIs). Then, 5 rectangular patches are extracted from current and 4 neighboring frames by the location of each RoI. Afterwards, they are normalized into 32×32 and stacked into a $5 \times 32 \times 32$ spatio-temporal cube (STC) $\mathbf{x} = [p_1; \cdots; p_5]$, where p_i is a normalized patch ($i = 1, \cdots, 5$). Note that a STC $\mathbf{x}$ serves as the basic representation of a video event, because it not only describes the foreground object but also contains its motion in a time interval. To apply generative E^3*Outlier*, we then design the operation

TABLE 4: Performance comparison (in %) of discriminative E^3*Outlier* with single-label (SL) and multi-label (ML) learning.

Dataset	MNIST			Fashion-MNIST			CIFAR10			SVHN			CIFAR100		
	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O
E^3Out. (SL)	94.1	99.3	67.5	**93.3**	**99.0**	**75.9**	83.5	97.5	43.4	86.0	98.0	36.7	79.2	96.8	33.3
E^3Out. (ML)	**95.4**	**99.5**	**71.1**	92.7	98.9	72.9	**84.1**	**97.6**	**45.1**	**86.9**	**98.1**	**38.5**	**80.0**	**97.0**	**34.9**

TABLE 5: Performance comparison (in %) of different DNN models for generative E^3*Outlier*.

Dataset	MNIST			Fashion-MNIST			CIFAR10			SVHN			CIFAR100		
	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O
CAE	**86.7**	**96.4**	**60.3**	**89.6**	**98.5**	**61.6**	66.3	93.5	20.0	63.6	93.9	15.0	61.2	92.4	16.7
UNet	82.0	95.0	56.5	86.4	98.0	52.8	**72.2**	**92.0**	**26.1**	**68.5**	**94.7**	**18.6**	**65.5**	**93.5**	**20.5**

(a) A person riding in the crowd. (b) A skater and a riding person. (c) A student throwing his backpack.

Fig. 9: Examples of abnormal events on UCSDped1, UCSDped2 and Avenue datasets (walking pedestrians are normal).

TABLE 6: Performance comparison of state-of-the-art UVAD methods with our E^3*Outlier* based UVAD solution in terms of frame-level AUC ("-" indicates that the performance is not reported).

	UCSDped1	UCSDped2	Avenue
SCD [10]	59.6%	63.0%	78.3%
UM [87]	68.4%	82.2%	80.6%
MC2ST [88]	71.8%	87.5%	84.4%
DOR [89]	71.7%	83.2%	-
E^3Out.	**79.5%**	**92.6%**	**89.2%**

$O(\cdot|y_1)$ and $O(\cdot|y_2)$ as follows: Given an input STC, $O(\cdot|y_1)$ is defined by $O(\mathbf{x}|y_1) = [p_1; p_2; p_4; p_5]$, which means deleting the middle patch in the STC $\mathbf{x}$. Meanwhile, we devise two types of $O(\cdot|y_2)$: (1) $O(\mathbf{x}|y_2) = p_3$, which suggests fetching the middle patch of $\mathbf{x}$. (2) $O(\mathbf{x}|y_2) = OF(p_3)$, which means transforming p_3 into its corresponding optical flow map. In this way, we actually define a self-supervised learning task that aims to infer p_3 and its optical flow map based on $\mathbf{x}$'s remaining patches p_1, p_2, p_4, p_5. We simple use CAE to carry out this generative task. As described in Sec. 3.6.1, we can train the models by the objective in Eq. (14) and score each STC by Eq. (15). The scores yielded by two types of $O(\cdot|y_2)$ operations are normalized and then summed to obtain the final score of each STC. The minimum of all STCs' scores on a frame is viewed as the frame score. More details are provided in Sec. 5 of supplementary material.

4.3.3 Performance Evaluation and Comparison

To evaluate the performance of our UVAD solution, we conduct experiments on three most commonly-used VAD benchmark datasets: UCSDped1 [91], UCSDped2 [91] and Avenue [92]. Following the standard practice in VAD, we compute frame-level AUC [91] as the quantitative performance measure, and compare our method with latest state-of-the-art UVAD approaches: Shuffled change detection (SCD) [10], Unmasking (UM) [87], Multiple Classifier Two Sample Test (MC2ST) [88], and Deep Ordinal Regression (DOR) [89]. The results are displayed in Table 6, and we can discover that the proposed E^3*Outlier* based UVAD solution outperforms existing UVAD solutions by by a 4% to 10% frame-level AUROC, which justifies E^3*Outlier* as a flexible and effective solution to different OD applications. Besides, unlike SCD, UM and MC2ST that require feature extraction based on hand-crafted descriptors, the proposed E^3*Outlier* based solution achieves end-to-end UVAD, while it also leads the other deep UVAD solution DOR by a huge margin.

5 Conclusion

In this paper, we propose a self-supervised deep OD framework named E^3*Outlier*. E^3*Outlier* for the first time leverages discriminative self-supervised learning for deep OD, which facilitates more effective representation learning from raw images. Then we demonstrate inlier priority, a property that lays the foundation for end-to-end OD, by both theory and empirical validations. Afterwards, we illustrate how the network uncertainty of discriminative DNNs can be utilized as a new outlierness measure, and present three specific outlier scores that can outperform the baseline. Then, the joint score refinement that fuses two types of strategies can be used to further boost OD performance. Finally, we demonstrate the applicability of E^3*Outlier* to different learning paradigms and other deep OD applications.

REFERENCES

[1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.

[2] A. Boukerche, L. Zheng, and O. Alfandi, "Outlier detection: Methods, models, and classification," *ACM Computing Surveys (CSUR)*, vol. 53, no. 3, pp. 1–37, 2020.

[3] U. Fiore, A. De Santis, F. Perla, P. Zanetti, and F. Palmieri, "Using generative adversarial networks for improving classification effectiveness in credit card fraud detection," *Information Sciences*, vol. 479, pp. 448–455, 2019.

[4] H. Soleimani and D. J. Miller, "Atd: Anomalous topic discovery in high dimensional discrete data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 9, pp. 2267–2280, 2016.

[5] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *International Conference on Information Processing in Medical Imaging*. Springer, 2017, pp. 146–157.

[6] J. Mao, T. Wang, C. Jin, and A. Zhou, "Feature grouping-based outlier detection upon streaming trajectories," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 12, pp. 2696–2709, 2017.

[7] D. M. J. Tax, "One-class classification," *Applied Sciences*, 2001.

[8] W. Liu, G. Hua, and J. R. Smith, "Unsupervised one-class learning for automatic outlier removal," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3826–3833.

[9] Y. Xia, X. Cao, F. Wen, G. Hua, and J. Sun, "Learning discriminative reconstructions for unsupervised outlier removal," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1511–1519.

[10] A. Del Giorno, J. A. Bagnell, and M. Hebert, "A discriminative framework for anomaly detection in large videos," in *European Conference on Computer Vision*. Springer, 2016, pp. 334–349.

[11] S. Wang, Y. Zeng, Q. Liu, C. Zhu, E. Zhu, and J. Yin, "Detecting abnormality without knowing normality: A two-stage approach for unsupervised video abnormal event detection," in *2018 ACM Multimedia Conference on Multimedia Conference*. ACM, 2018, pp. 636–644.

[12] S. Wang, E. Zhu, X. Hu, X. Liu, Q. Liu, J. Yin, and F. Wang, "Robustness can be cheap: A highly efficient approach to discover outliers under high outlier ratios," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 5313–5320.

[13] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2. Ieee, 1999, pp. 1150–1157.

[14] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 3360–3367.

[15] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[16] C. C. Aggarwal, "An introduction to outlier analysis," in *Outlier analysis*. Springer, 2017, pp. 1–34.

[17] H. Wang, M. J. Bah, and M. Hammad, "Progress in outlier detection techniques: A survey," *Ieee Access*, vol. 7, pp. 107 964–108 000, 2019.

[18] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM Computing Surveys (CSUR)*, vol. 54, no. 2, pp. 1–38, 2021.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[20] S. Wang, Y. Zeng, X. Liu, E. Zhu, J. Yin, C. Xu, and M. Kloft, "Effective end-to-end unsupervised outlier detection via inlier priority of discriminative network," in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2019, pp. 5962–5975.

[21] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," in *ACM Sigmod Record*, vol. 29, no. 2. ACM, 2000, pp. 427–438.

[22] E. M. Knox and R. T. Ng, "Algorithms for mining distancebased outliers in large datasets," in *Proceedings of the international conference on very large data bases*. Citeseer, 1998, pp. 392–403.

[23] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *ACM sigmod record*, vol. 29, no. 2. ACM, 2000, pp. 93–104.

[24] J. Tang, Z. Chen, A. W.-C. Fu, and D. W. Cheung, "Enhancing effectiveness of outlier detections for low density patterns," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2002, pp. 535–548.

[25] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "Loop: local outlier probabilities," in *Proceedings of the 18th ACM conference on Information and knowledge management*, 2009, pp. 1649–1652.

[26] T. R. Bandaragoda, K. M. Ting, D. Albrecht, F. T. Liu, and J. R. Wells, "Efficient anomaly detection by isolation using nearest neighbour ensemble," in *2014 IEEE International Conference on Data Mining Workshop*. IEEE, 2014, pp. 698–705.

[27] G. Pang, K. M. Ting, and D. Albrecht, "Lesinn: Detecting anomalies by identifying least similar nearest neighbours," in *2015 IEEE international conference on data mining workshop (ICDMW)*. IEEE, 2015, pp. 623–630.

[28] X. Yang, L. J. Latecki, and D. Pokrajac, "Outlier detection with globally optimal exemplar-based gmm," in *Proceedings of the 2009 SIAM International Conference on Data Mining*. SIAM, 2009, pp. 145–154.

[29] X.-m. Tang, R.-x. Yuan, and J. Chen, "Outlier detection in energy disaggregation using subspace learning and gaussian mixture model," *International Journal of Control and Automation*, vol. 8, no. 8, pp. 161–170, 2015.

[30] L. J. Latecki, A. Lazarevic, and D. Pokrajac, "Outlier detection with kernel density functions," in *International Workshop on Machine Learning and Data Mining in Pattern Recognition*. Springer, 2007, pp. 61–75.

[31] A. P. Boedihardjo, C.-T. Lu, and F. Chen, "Fast adaptive kernel density estimator for data streams," *Knowledge and Information Systems*, vol. 42, no. 2, pp. 285–317, 2015.

[32] L. Zhang, J. Lin, and R. Karim, "Adaptive kernel density-based anomaly detection for nonlinear systems," *Knowledge-Based Systems*, vol. 139, pp. 50–63, 2018.

[33] X. Qin, L. Cao, E. A. Rundensteiner, and S. Madden, "Scalable kernel density estimation-based local outlier detection over large data streams." in *EDBT*, 2019, pp. 421–432.

[34] M.-F. Jiang, S.-S. Tseng, and C.-M. Su, "Two-phase clustering process for outliers detection," *Pattern recognition letters*, vol. 22, no. 6-7, pp. 691–700, 2001.

[35] Z. He, X. Xu, and S. Deng, "Discovering cluster-based local outliers," *Pattern Recognition Letters*, vol. 24, no. 9-10, pp. 1641–1650, 2003.

[36] J. Yin and J. Wang, "A model-based approach for text clustering with outlier detection," in *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*. IEEE, 2016, pp. 625–636.

[37] M. Chenaghlou, M. Moshtaghi, C. Leckie, and M. Salehi, "Online clustering for evolving data streams with online anomaly detection," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2018, pp. 508–521.

[38] B. Schölkopf, A. Smola, and K.-R. Müller, "Kernel principal component analysis," in *International Conference on Artificial Neural Networks*. Springer, 1997, pp. 583–588.

[39] J. Chen, S. Sathe, C. Aggarwal, and D. Turaga, "Outlier detection with autoencoder ensembles," in *Proceedings of the 2017 SIAM international conference on data mining*. SIAM, 2017, pp. 90–98.

[40] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 2008, pp. 413–422.

[41] S. Hariri, M. C. Kind, and R. J. Brunner, "Extended isolation forest," *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[42] Y. Wang, S. Parthasarathy, and S. Tatikonda, "Locality sensitive outlier detection: A ranking driven approach," in *2011 IEEE 27th International Conference on Data Engineering*. IEEE, 2011, pp. 410–421.

[43] T. Pevnỳ, "Loda: Lightweight on-line detector of anomalies," *Machine Learning*, vol. 102, no. 2, pp. 275–304, 2016.

[44] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," *arXiv preprint arXiv:1901.03407*, 2019.

[45] S. Zhai, Y. Cheng, W. Lu, and Z. Zhang, "Deep structured energy based models for anomaly detection," *international conference on machine learning*, pp. 1100–1109, 2016.

[46] C. Zhou and R. C. Paffenroth, "Anomaly detection with robust deep autoencoders," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017, pp. 665–674.

[47] R. Chalapathy, A. K. Menon, and S. Chawla, "Robust, deep and inductive anomaly detection," in *Joint European Conference on Ma-*

chine Learning and Knowledge Discovery in Databases. Springer, 2017, pp. 36–51.

[48] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," in *International Conference on Learning Representations (ICLR)*, 2018.

[49] G. Pang, L. Cao, L. Chen, and H. Liu, "Learning representations of ultrahigh-dimensional data for random distance-based outlier detection," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 2041–2050.

[50] Y. Liu, Z. Li, C. Zhou, Y. Jiang, J. Sun, M. Wang, and X. He, "Generative adversarial active learning for unsupervised outlier detection," *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[51] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Advances in neural information processing systems*, vol. 3, no. 06, 2014.

[52] C. Lai, D. Zou, and G. Lerman, "Robust subspace recovery layer for unsupervised anomaly detection," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2020.

[53] N. Tajbakhsh, Y. Hu, J. Cao, X. Yan, Y. Xiao, Y. Lu, J. Liang, D. Terzopoulos, and X. Ding, "Surrogate supervision for medical image analysis: Effective deep learning from limited quantities of labeled data," in *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*. IEEE, 2019, pp. 1251–1255.

[54] D. Zhang, J. Han, and Y. Zhang, "Supervision by fusion: Towards unsupervised learning of deep salient object detector," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4048–4056.

[55] X. Ji, J. F. Henriques, and A. Vedaldi, "Invariant information clustering for unsupervised image classification and segmentation," *arXiv: Computer Vision and Pattern Recognition*, 2018.

[56] N. Komodakis and S. Gidaris, "Unsupervised representation learning by predicting image rotations," in *International Conference on Learning Representations (ICLR)*, 2018.

[57] I. Golan and R. El-Yaniv, "Deep anomaly detection using geometric transformations," in *Advances in Neural Information Processing Systems*, 2018, pp. 9758–9769.

[58] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "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)*, 2019, pp. 4171–4186.

[59] R. Santa Cruz, B. Fernando, A. Cherian, and S. Gould, "Visual permutation learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[60] I. Misra, C. L. Zitnick, and M. Hebert, "Shuffle and learn: unsupervised learning using temporal order verification," in *European Conference on Computer Vision*. Springer, 2016, pp. 527–544.

[61] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3733–3742.

[62] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey." *arXiv: Computer Vision and Pattern Recognition*, 2019.

[63] A. Malinin and M. Gales, "Predictive uncertainty estimation via prior networks," in *Advances in Neural Information Processing Systems*, 2018, pp. 7047–7058.

[64] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*, 2016, pp. 1050–1059.

[65] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in neural information processing systems*, 2017, pp. 6402–6413.

[66] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," 2017.

[67] J. Snoek, Y. Ovadia, E. Fertig, B. Lakshminarayanan, S. Nowozin, D. Sculley, J. Dillon, J. Ren, and Z. Nado, "Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift," in *Advances in Neural Information Processing Systems*, 2019, pp. 13 969–13 980.

[68] D. M. Hawkins, *Identification of outliers*. Springer, vol. 11.

[69] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *International Conference on Machine Learning*, 2016, pp. 1558–1566.

[70] A. Dosovitskiy and T. Brox, "Generating images with perceptual similarity metrics based on deep networks," in *Advances in neural information processing systems*, 2016, pp. 658–666.

[71] J. Gui, Z. Sun, Y. Wen, D. Tao, and J. Ye, "A review on generative adversarial networks: Algorithms, theory, and applications," *arXiv preprint arXiv:2001.06937*, 2020.

[72] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *British Machine Vision Conference 2016*. British Machine Vision Association, 2016.

[73] R. Anand, K. G. Mehrotra, C. K. Mohan, and S. Ranka, "An improved algorithm for neural network classification of imbalanced training sets," *IEEE Transactions on Neural Networks*, vol. 4, no. 6, pp. 962–969, 1993.

[74] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1321–1330.

[75] T. G. Dietterich, "Ensemble methods in machine learning," in *International workshop on multiple classifier systems*. Springer, 2000, pp. 1–15.

[76] O. Ronneberger, P. Fischer, and T. Brox, *U-Net: Convolutional Networks for Biomedical Image Segmentation*. Springer, Cham, 2015.

[77] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[78] J. Tack, S. Mo, J. Jeong, and J. Shin, "Csi: Novelty detection via contrastive learning on distributionally shifted instances," *Advances in Neural Information Processing Systems*, vol. 33, pp. 11 839–11 852, 2020.

[79] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[80] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[81] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Citeseer, Tech. Rep., 2009.

[82] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," 2011.

[83] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 233–240.

[84] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *International Conference on Artificial Neural Networks*. Springer, 2011, pp. 52–59.

[85] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[86] B. Ramachandra, M. Jones, and R. R. Vatsavai, "A survey of single-scene video anomaly detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[87] R. Tudor Ionescu, S. Smeureanu, B. Alexe, and M. Popescu, "Unmasking the abnormal events in video," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2895–2903.

[88] Y. Liu, C.-L. Li, and B. Póczos, "Classifier two sample test for video anomaly detections." in *BMVC*, 2018, p. 71.

[89] G. Pang, C. Yan, C. Shen, A. V. D. Hengel, and X. Bai, "Self-trained deep ordinal regression for end-to-end video anomaly detection," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12 170–12 179, 2020.

[90] G. Yu, S. Wang, Z. Cai, E. Zhu, C. Xu, J. Yin, and M. Kloft, "Cloze test helps: Effective video anomaly detection via learning to complete video events," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 583–591.

[91] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos, "Anomaly detection in crowded scenes," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 1975–1981.

[92] C. Lu, J. Shi, and J. Jia, "Abnormal event detection at 150 fps in matlab," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 2720–2727.

Siqi Wang received the Ph.D. degree in computer science and technology from the National University of Defense Technology (NUDT), China. He is currently an assistant research professor in College of Computer, NUDT. His main research include outlier/anomaly detection and unsupervised learning. His works have been published on leading conferences and journals, such as NeurIPS, AAAI, IJCAI, ACM MM, TPAMI, TIP, PR, TCYB and Neurocomputing. He serves as a PC member and reviewer for top-tier conference like NeurIPS and AAAI and several prestigious journals.

Yijie Zeng received the B.Sc. in computational mathematics from University of Science and Technology of China in 2015, and the Ph.D. degree in the School of Electrical and Electronic Engineering from Nanyang Technological University, Singapore in 2020. His research interests include machine learning, computer vision, and pattern recognition.

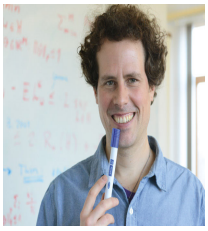
Guang Yu received the bachelor's degree in computer science and technology from Sichuan University, Chengdu, China, in 2018. He is currently working toward the Ph.D. degree at the College of Computer, National University of Defense Technology, Changsha, China. His main research interests include anomaly/outlier detection and self-supervised/unsupervised learning.

Zhen Cheng is currently pursuing the Ph.D. degree with the National University of Defense Technology (NUDT), China. His current research interests include transfer learning, outlier detection, and deep neural networks.

Xinwang Liu received his PhD degree from National University of Defense Technology (NUDT), China. He is now Professor of School of Computer, NUDT. His current research interests include kernel learning and unsupervised feature learning. Dr. Liu has published 60+ peer-reviewed papers, including those in highly regarded journals and conferences such as TPAMI, TKDE, TIP, TNNLS, TMM, TIFS, NeurIPS, ICCV, CVPR, AAAI, IJCAI, etc.

Sihang Zhou received his PhD degree from National University of Defense Technology (NUDT), China. He is now lecturer at College of Intelligence Science and Technology, NUDT. His current research interests include machine learning and medical image analysis. Dr. Zhou has published 20+ peer-reviewed papers, including IEEE T-IP, IEEE T-NNLS, IEEE T-MI, Information Fusion, Medical Image Analysis, AAAI, MICCAI.

En Zhu received his PhD degree from National University of Defense Technology (NUDT), China. He is now Professor at School of Computer Science, NUDT, China. His main research interests are pattern recognition, image processing, machine vision and machine learning. Dr. Zhu has published 60+ peer-reviewed papers, including IEEE T-CSVT, IEEE T-NNLS, PR, AAAI, IJCAI, etc. He was awarded China National Excellence Doctoral Dissertation.



Marius Kloft is a professor of computer science at TU Kaiserslautern and an adjunct faculty member of the University of Southern California. Previously he was a junior professor at HU Berlin and a joint postdoctoral fellow at the Courant Institute of Mathematical Sciences and Memorial Sloan-Kettering Cancer Center, New York. He earned his PhD at TU Berlin and UC Berkeley.

Jianping Yin received his PhD degree from National University of Defense Technology (NUDT), China. He is now the distinguished Professor at Dongguan University of Technology. His research interests include pattern recognition and machine learning. Dr. Yin has published 150+ peer-reviewed papers, including IEEE T-CSVT, IEEE T-NNLS, PR, AAAI, IJCAI, etc. He was awarded China National Excellence Doctoral Dissertation' Supervisor and National Excellence Teacher. He served on the Technical Program Committees of 30+ international conferences and workshops.

Qing Liao received her Ph.D. degree in computer science and engineering in 2016 supervised by Prof. Qian Zhang from the Department of Computer Science and Engineering of the Hong Kong University of Science and Technology. She is currently an assistant professor with School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), China. Her research interests include artificial intelligence and bioinformatics.

E^3*Outlier*: A Self-supervised Framework for Unsupervised Deep Outlier Detection –Supplementary Material–

Siqi Wang, Yijie Zeng, Guang Yu, Zhen Cheng, Xinwang Liu, Sihang Zhou, En Zhu, Marius Kloft, Jianping Yin, Qing Liao

1 Data Operation Design

To obtain sufficient data operations for creating abundant pseudo classes, we design each data operation $O(\cdot|y)$ by combining one or more basic transformations drawn from several transformation categories below: (1) *Rotation*, which includes simple rotation transformations that clock-wisely rotate images by integer times of 90°: $\mathcal{T}_{SR} = \{Rot(\cdot, (y-1)\cdot 90°)\}_{y=1}^{4}$, and irregular rotation transformations by integer times of 30° (transformations already in $\mathcal{T}_{SR}$ are excluded): $\mathcal{T}_{IR} = \{Rot(\cdot, (y-1)\cdot 30°)\}_{y=1}^{12} - \mathcal{T}_{SR}$. (2) *Flip*: $\mathcal{T}_F = \{Flip(\cdot, y)\}_{y=0}^{1}$, where $y = 1/0$ refers to flipping the image or not. (3) *Shifting*, which includes x-axis shifting: $\mathcal{T}_{Sx} = \{S_x(\cdot, (y-2)\cdot D)\}_{y=1}^{3}$ and y-axis shifting: $\mathcal{T}_{Sy} = \{S_y(\cdot, (y-2)\cdot D)\}_{y=1}^{3}$ (D is the step of shifting). (4) *Patch re-arranging*, which partitions the image into M equally-sized patches and re-organizes them into a new image by a permutation selected from $M!$ possible permutations: $\mathcal{T}_{PR} = \{PR(\cdot, perm_y)\}_{y=1}^{M!}$. Next, we design three operation subsets (regular affine operation set $\mathcal{O}_{RA}$, irregular affine operation set $\mathcal{O}_{IA}$ and patch re-arranging operation set $\mathcal{O}_{PR}$) by joining transformations from above categories ("$\times$" refers to Cartesian product):

$$\begin{aligned} \mathcal{O}_{RA} &= \mathcal{T}_{SR} \times \mathcal{T}_F \times \mathcal{T}_{Sx} \times \mathcal{T}_{Sy}, \\ \mathcal{O}_{IA} &= \mathcal{T}_{IR} \times \mathcal{T}_F, \quad \mathcal{O}_{PR} = \mathcal{T}_{PR} \end{aligned} \tag{1}$$

In our experiments, we choose $D = 8$ pixels and $M = 4$. In this way, we construct $72, 16$ and 24 operations for $\mathcal{O}_{RA}$, $\mathcal{O}_{IA}$ and $\mathcal{O}_{PR}$ respectively. The final operation set is yielded by $\mathcal{O} = \mathcal{O}_{RA} \cup \mathcal{O}_{IA} \cup \mathcal{O}_{PR}$. The reason why we construct operation sets as (1) is to avoid joining two transformation types that both produce obvious image artifact (e.g. shifting and irregular rotation), which tends to degrade OD performance. It should be noted that other ways to design data operations are also possible. $\mathcal{O}$ can be extended by adding new data operations, and SSD can adapt by simply modifying the number of nodes in its Softmax layer. To just the necessity of such an operation design, we evaluate the performance of E^3*Outlier* without score refinement when different combinations of operation sets are used to provide self-supervision: As suggested by results in Table 1, using $\mathcal{O}_{RA}$ alone has already been able to achieve superior performance to previous DNN based OD methods, but adding more types of operations sets constantly brings about performance gain. When using $\mathcal{O}_{RA} \cup \mathcal{O}_{IA} \cup \mathcal{O}_{PR}$, we even obtain up to 6.2% AUROC improvement when compared with using $\mathcal{O}_{RA}$ alone on CIFAR100. Such results verify the necessity to combine different types of operation sets for self-supervision, and they also reveal the potential to develop more types of operation sets for further performance improvement.

TABLE 1: Comparison of different operation set designs.

	MST	FMST	C10	SH	C100
$\mathcal{O}_{RA}$	92.6	92.3	80.7	81.9	73.0
$\mathcal{O}_{RA} \cup \mathcal{O}_{IA}$	92.9	93.0	82.7	83.6	77.3
$\mathcal{O}_{RA} \cup \mathcal{O}_{IA} \cup \mathcal{O}_{PR}$	94.1	93.3	83.5	86.0	79.2

2 Theoretical Deviration on Inlier Priority

We consider an SSD with its network weights randomly initialized by i.i.d. uniform distribution on $[-1, 1]$. Suppose that the network of SSD has an $(L+1)$-node penultimate layer and a final K-node softmax layer. We discuss the case of inliers X_{in} first: For cross-entropy loss $\mathcal{L}$, only transformed inliers generated by the c-th operation $X_{in}^{(c)} = \{\mathbf{x}^{(c)} | \mathbf{x} \in X_{in}\}$ are used to update $\mathbf{w}_c$. The gradient vector incurred by $X_{in}^{(c)}$ is denoted by $\nabla_{\mathbf{w}_c}^{(in)} \mathcal{L} = [\nabla_{w_{s,c}} \mathcal{L}]_{s=1}^{(L+1)}$ with its element $\nabla_{w_{s,c}} \mathcal{L}$ given by:

$$\nabla_{w_{s,c}} \mathcal{L} = \sum_{i=1}^{N_{in}} \nabla_{w_{s,c}} \mathcal{L}(\mathbf{x}_i) = \sum_{i=1}^{N_{in}} (P^{(c)}(\mathbf{x}_i) - 1) h^{(s)}(\mathbf{x}_i) \tag{2}$$

where N_{in} is the inlier number (N_{out} is the outlier number), $P^{(c)}(\mathbf{x}_i)$ is the c-th node's output of the Softmax layer and $h^{(s)}(\mathbf{x}_i)$ is the s-th node's output of the penultimate layer for $\mathbf{x}_i \in X_{in}^{(c)}$. Since SSD is randomly initialized, we compute the expectation of inliers' gradient magnitude to update

Manuscript received June 30, 2020.

$\mathbf{w}_c$, i.e. $E(||\nabla^{(in)}_{\mathbf{w}_c}\mathcal{L}||_2^2)$. As $||\nabla^{(in)}_{\mathbf{w}_c}\mathcal{L}||_2^2 = \sum_{s=1}^{L+1}(\nabla_{w_{s,c}}\mathcal{L})^2$, it needs to compute the term below:

$$\begin{aligned} E((\nabla_{w_{s,c}}\mathcal{L})^2) &= E((\sum_{i=1}^{N_{in}}\nabla_{w_{s,c}}\mathcal{L}(\mathbf{x}_i))^2) \\ &= \sum_{i=1}^{N_{in}}\sum_{j=1}^{N_{in}} E(\nabla_{w_{s,c}}\mathcal{L}(\mathbf{x}_i)\nabla_{w_{s,c}}\mathcal{L}(\mathbf{x}_j)). \end{aligned} \tag{3}$$

To compute (3), we first define a function $g_{ij}^{(s,c)}$ as follows:

$$\begin{aligned} g_{ij}^{(s,c)} &= \nabla_{w_{s,c}}\mathcal{L}(\mathbf{x}_i)\nabla_{w_{s,c}}\mathcal{L}(\mathbf{x}_j) \\ &= (P^{(c)}(\mathbf{x}_i)-1)(P^{(c)}(\mathbf{x}_j)-1)h^{(s)}(\mathbf{x}_i)h^{(s)}(\mathbf{x}_j) \end{aligned} \tag{4}$$

where $h^{(s)}(\mathbf{x}_i)$ is the penultimate layer's s-th node's output and $P^{(c)}(\mathbf{x}_i)$ denotes the softmax layer's c-th node's output for $\mathbf{x}_i$. Our goal is to compute $E(g_{ij}^{(s,c)})$ w.r.t the weights between the penultimate layer and the final softmax layer, which is a $(L+1)\times K$ vector $\mathbf{w} = [\mathbf{w}_c]_{c=1}^{K}$, with the weights associated with the c-th class ($1 \le c \le K$) to be a $(L+1)$ column vector $\mathbf{w}_c = [w_{s,c}]_{s=1}^{L+1}$. To simplify computation, we use the second-order Taylor series expansion of $g_{ij}^{(s,c)}$:

$$\begin{aligned} g_{ij}^{(s,c)}(\mathbf{w}) \approx & g_{ij}^{(s,c)}(\boldsymbol{\mu}) + \nabla_{\mathbf{w}}g_{ij}^{(s,c)}(\boldsymbol{\mu})\cdot(\mathbf{w}-\boldsymbol{\mu}) \\ & + \frac{1}{2}(\mathbf{w}-\boldsymbol{\mu})^T\cdot\nabla^2_{\mathbf{w}}g_{ij}^{(s,c)}(\boldsymbol{\mu})\cdot(\mathbf{w}-\boldsymbol{\mu}) \end{aligned} \tag{5}$$

where $\boldsymbol{\mu}$ is the expectation of $\mathbf{w}$. Since each weight in $\mathbf{w}$ is drawn from i.i.d uniform distribution on $[-1,1]$, we have $\mu_{s,c} = E(w_{s,c}) = 0$, $E(w_{s,c}^2) = \frac{1}{3}$ and $E(w_{s,c}w_{t,c}) = 0$ ($s \neq t$). Therefore, the expectation of $g_{ij}^{(s,c)}$ w.r.t. $\mathbf{w}$ is approximated as

$$\begin{aligned} E(g_{ij}^{(s,c)}(\mathbf{w})) \approx & g_{ij}^{(s,c)}(\mathbf{0}) + \frac{1}{2}\sum_{t=1}^{L+1}\sum_{l=1}^{K}\nabla^2_{w_{t,l}}g_{ij}^{(s,c)}(\mathbf{0})E(w_{s,c}^2) \\ = & g_{ij}^{(s,c)}(\mathbf{0}) + \frac{1}{6}\sum_{t=1}^{L+1}\sum_{l=1}^{K}\nabla^2_{w_{t,l}}g_{ij}^{(s,c)}(\mathbf{0}) \end{aligned} \tag{6}$$

Thus, computing $E(g_{ij}^{(s,c)}(\mathbf{w}))$ requires the computation of $\nabla^2_{w_{t,l}}g_{ij}^{(s,c)}(\mathbf{0})$. Recall the softmax probability is computed by:

$$P^{(c)}(\mathbf{x}_i) = \frac{e^{\mathbf{h}^\top(\mathbf{x}_i)\cdot\mathbf{w}_c}}{\sum_{l=1}^{K}e^{\mathbf{h}^\top(\mathbf{x}_i)\cdot\mathbf{w}_l}} \tag{7}$$

where $\mathbf{h}(\mathbf{x}_i) = [h^{(s)}(\mathbf{x}_i)]_{s=1}^{L+1}$ is penultimate layer's output for $\mathbf{x}_i$. Since $\mathbf{h}(\mathbf{x}_i)$ is independent of $\mathbf{w}$, we have:

$$\nabla_{w_{t,l}}P^{(c)}(\mathbf{x}_i) = -P^{(c)}(\mathbf{x}_i)(\delta_{c,l}-P^{(l)}(\mathbf{x}_i))\cdot h^{(t)}(\mathbf{x}_i) \tag{8}$$

where $\delta_{c,l} = 1$ if $c = l$ and $\delta_{c,l} = 0$ otherwise. Using (4), (7) and (8), we can calculate $\nabla^2_{w_{t,l}}g_{ij}^{(s,c)}$ by:

$$\begin{aligned} &\nabla^2_{w_{t,l}}g_{ij}^{(s,c)} = h^{(s)}(\mathbf{x}_i)h^{(s)}(\mathbf{x}_j)\times \\ &\Big[-(h^{(t)}(\mathbf{x}_i))^2P^{(c)}(\mathbf{x}_i)(\delta_{c,l}-P^{(l)}(\mathbf{x}_i))^2(1-P^{(c)}(\mathbf{x}_j))+ \\ &(h^{(t)}(\mathbf{x}_i))^2P^{(c)}(\mathbf{x}_i)P^{(l)}(\mathbf{x}_i)(1-P^{(l)}(\mathbf{x}_i))(1-P^{(c)}(\mathbf{x}_j))+ \\ &2h^{(t)}(\mathbf{x}_i)h^{(t)}(\mathbf{x}_j)P^{(c)}(\mathbf{x}_i)P^{(c)}(\mathbf{x}_j)(\delta_{c,l}-P^{(l)}(\mathbf{x}_i))(\delta_{c,l}-P^{(l)}(\mathbf{x}_j)) \\ &-(h^{(t)}(\mathbf{x}_j))^2P^{(c)}(\mathbf{x}_j)(\delta_{c,l}-P^{(l)}(\mathbf{x}_j))^2(1-P^{(c)}(\mathbf{x}_i))+ \\ &(h^{(t)}(\mathbf{x}_j))^2P^{(c)}(\mathbf{x}_j)P^{(l)}(\mathbf{x}_j)(1-P^{(l)}(\mathbf{x}_j))(1-P^{(c)}(\mathbf{x}_i))\Big]times \end{aligned} \tag{9}$$

Therefore, in the summation term of (6), we have $(L+1)$ terms that satisfy $c = l$, and in this case $\nabla^2_{w_{t,l}}g_{ij}^{(s,c)}|_{\mathbf{w}=\mathbf{0}}$ is:

$$\begin{aligned} &h^{(s)}(\mathbf{x}_i)h^{(s)}(\mathbf{x}_j)\times \\ &\Big[(h^{(t)}(\mathbf{x}_i))^2\frac{(K-1)^2(2-K)}{K^4}+ \\ &(h^{(t)}(\mathbf{x}_j))^2\frac{(K-1)^2(2-K)}{K^4}+2h^{(t)}(\mathbf{x}_i)h^{(t)}(\mathbf{x}_j)\frac{(K-1)^2}{K^4}\Big] \end{aligned} \tag{10}$$

For the rest $(L+1)(K-1)$ terms in the summation term that satisfy $c \neq l$, $\nabla^2_{w_{t,l}}g_{ij}^{(s,c)}|_{\mathbf{w}=\mathbf{0}}$ is:

$$\begin{aligned} &h^{(s)}(\mathbf{x}_i)h^{(s)}(\mathbf{x}_j)\times \\ &\Big[(h^{(t)}(\mathbf{x}_i))^2\frac{(K-1)(K-2)}{K^4}+ \\ &(h^{(t)}(\mathbf{x}_j))^2\frac{(K-1)(K-2)}{K^4}+2h^{(t)}(\mathbf{x}_i)h^{(t)}(\mathbf{x}_j)\frac{1}{K^4}\Big] \end{aligned} \tag{11}$$

By substituting (10) and (11) into (6), we can obtain the result of (5) in the original manuscript:

$$\begin{aligned} &E(\nabla_{w_{s,c}}\mathcal{L}(\mathbf{x}_i)\nabla_{w_{s,c}}\mathcal{L}(\mathbf{x}_j)) \\ &= E(g_{ij}^{(s,c)}(\mathbf{w})) \\ &\approx h^{(s)}(\mathbf{x}_i)h^{(s)}(\mathbf{x}_j)\Big[\frac{(K-1)^2}{K^2}+\frac{(K-1)}{3K^3}\sum_{t=1}^{L+1}h^{(t)}(\mathbf{x}_i)h^{(t)}(\mathbf{x}_j)\Big] \end{aligned} \tag{12}$$

It remains to calculate the expectation of $h^{(t)}(\mathbf{x}_i)h^{(t)}(\mathbf{x}_j)$ in (12). To make its calculation tractable, we consider a simplified case of a network with a single hidden-layer and sigmoid activation. In this case, by [1, Lemma 3.b], the expectation of $h^{(s)}(\mathbf{x}_i)h^{(s)}(\mathbf{x}_j)$ w.r.t. the randomly initialized weights between the input and hidden layer satisfies $E(h^{(s)}(\mathbf{x}_i)h^{(s)}(\mathbf{x}_j)) \approx \frac{1}{4}$ and $E(h^{(s)}(\mathbf{x}_i)^2h^{(s)}(\mathbf{x}_j)^2) \approx \frac{1}{16}$. Thus, by definition of $||\nabla^{(in)}_{\mathbf{w}_c}\mathcal{L}||_2^2$ and (3), we yield:

$$\begin{aligned} &E(||\nabla^{(in)}_{\mathbf{w}_c}\mathcal{L}||_2^2) \\ &\approx N_{in}^2\Big[(L+1)(\frac{(K-1)^2}{4K^2}+\frac{(K-1)(L+1)}{48K^3})\Big] \\ &\triangleq N_{in}^2\cdot Q \end{aligned} \tag{13}$$

Since L, K, Q are constant, (13) suggests that the magnitude of inliers' gradient $E(||\nabla^{(in)}_{\mathbf{w}_c}\mathcal{L}||_2^2)$ is proportional to N_{in}^2. Similarly, outliers' gradient magnitude $E(||\nabla^{(out)}_{\mathbf{w}_c}\mathcal{L}||_2^2) \approx N_{out}^2\cdot Q$.

TABLE 2: Influence of rebooting times for re-weighting.

# OF REB.	MST	FMST	C10	SH	C100
0	94.1	93.3	83.2	85.9	79.2
1	94.7	93.6	84.3	86.6	80.6
2	94.9	93.6	84.0	86.3	80.3
3	95.0	93.6	84.3	86.4	80.3
4	95.1	93.6	84.2	86.7	80.9
5	95.0	93.6	83.5	86.1	80.5

3 Details of Network Uncertainty Demonstration Experiment

To serve this purpose, we generate 2D data $\mathbf{x}_i = [x_i, y_i]$ by a Gaussian Process (GP) with Radial Basis Function (RBF) kernel, and its covariance matrix is computed by $K_{ij} = exp[-\frac{(x_i - x_j)^2}{2\sigma^2}]$. Meanwhile, we add a term ϵ_i^2 to the i-th diagonal element of covariance matrix, where $\epsilon_i = 0.3x_i^2$. Abscissa vales of sampled data (x_i) are uniformly drawn from the $(-2.25, 2.25)$ interval, while ordinate values (y_i) are sampled from the GP. The MC-Dropout method is implemented by a public respiratory[1]. Codes of the demonstration experiment are also available in our open respiratory[2].

4 Details of Score Refinement

TABLE 3: Influence of model number for ensemble.

# OF ENS.	MST	FMST	C10	SH	C100
1	94.0	93.1	83.5	85.8	78.8
3	94.3	93.2	83.7	86.4	80.5
5	94.4	93.3	84.0	86.6	80.9
7	94.4	93.3	84.0	86.6	81.1

In this section, we show the full algorithm procedure of online re-weighting, reboot re-weighting and joint score refinement in Algorithm 1, 2 and 3 respectively. We also discuss two key parameters for score refinement: **(1)** The times of rebooting for re-weighting strategy. Intuitively, the rebooting can be performed multiple times, as better outlier scores will be obtained for SSD's initialization after each rebooting. Thus, we evaluate the OD performance when rebooting is performed by 0 to 5 times. As shown by Table 2, we obtain some interesting observations: Using rebooting always produces improvement when compared with the case without rebooting, but rebooting for multiple times does not necessarily lead to better OD performance than rebooting once. This also explains why we only perform rebooting once in the final JSR. **(2)** The model number for ensemble. We also explore how model number influences the ensemble strategy, and we evaluate the performance when 1 to 7 SSD models are used for ensemble in one experiment. As can be seen from Table 3, it is observed on all benchmarks that adding more models can steadily improve performance, but the performance tends to level off as the number of models reach a certain point.

1. https://github.com/JavierAntoran/Bayesian-Neural-Networks
2. https://github.com/demonzyj56/E3Outlier

Algorithm 1 Online Re-weighting

Input: X', Y, total training epochs T.
Output: SSD model M.
Randomly initialize SSD M, set weight $w_i = \frac{1}{N}$, $i = 1, \cdots, N$.
for $t = 1, \cdots T$ **do**
  Training M by mini-batch optimization of (12) in the manuscript with X', Y, w_i, $i = 1, \cdots, N$.
  Obtain scores $S(x_1), \cdots, S(x_N)$ by a forward pass of X' into M and any outlier score.
  Normalize $S(x_i)$ into $\bar{S}(x_i)$, $i = 1, \cdots, N$, such that $\sum_{i=1}^{N} \bar{S}(x_i) = 1$ and $\bar{S}(x_i) \geq 0$.
  Update $w_i = \bar{S}(x_i)$, $i = 1, \cdots, N$.
end for

Algorithm 2 Reboot Re-weighting

Input: X', Y, total training epochs T.
Output: SSD model M'.
Randomly initialize SSD M and M'.
for $t = 1, \cdots T$ **do**
  Training M by mini-batch optimization of (1) in the manuscript with X', Y.
end for
Obtain scores $S(x_1), \cdots, S(x_N)$ by a forward pass of X' into M and any outlier score.
Normalize $S(x_i)$ into $\bar{S}(x_i)$, $i = 1, \cdots, N$, such that $\sum_{i=1}^{N} \bar{S}(x_i) = 1$ and $\bar{S}(x_i) \geq 0$.
Set weight $w_i = \bar{S}(x_i)$, $i = 1, \cdots, N$.
for $t = 1, \cdots T$ **do**
  Training M' by mini-batch optimization of (12) in the manuscript with X', Y, w_i, $i = 1, \cdots, N$.
end for

5 Implementation Details

For discriminative *E*^{3}*Outlier*, we use the Wide ResNet (WRN) with the widen factor $k = 4$ as the backbone DNN architecture. As illustrated in Sec. 1, $K = 112$ operations are used for self-supervised learning. Since the self-supervised learning paradigm will augment original data by K times, we train WRN for $\lceil \frac{250}{K} \rceil$ epochs. The batch size is 128. A learning rate 0.001 and a weight decay 0.0005 are adopted. The SGD optimizer with momentum 0.9 is used for MNIST and Fashion-MNIST, while the Adam optimizer with $\boldsymbol{\beta} = (0.9, 0.999)$ is used on CIFAR10, CIFAR100 and SVHN for better convergence. For the ensemble strategy, we set $e = 5$ with different random seeds. We use NE score S_{ne} by default, as it achieves the best trade off between performance and computational cost. As to generative *E*^{3}*Outlier*, we choose the $O(\cdot|y_1)$ from the operation set defined below: Given an operation set with flip $\mathcal{T}'_F = \{Flip(\cdot)\}$, an operation set with RGB-to-gray operation $\mathcal{T}_G = Gray(\cdot)$ and an operation set with simple rotations $\mathcal{T}_{SR} = \{Rot(\cdot, (y-1) \cdot 90°)\}_{y=1}^{4}$, the operation set $O(\cdot|y_1)$ is chosen from the composited set $\mathcal{T}'_F \times \mathcal{T}_G \times \mathcal{T}_{SR}$, which contains 4 operations in total. $O(\cdot|y_2)$ is chosen to be an identity map. CAE and UNet are used to implement the generative DNN $\mathcal{G}$. The structure of CAE is the same as other CAE based deep OD methods, while the adopted U-

Algorithm 3 Joint Score Refinement

Input: X', Y, total training epochs T.
Output: An ensemble of SSD models $\{M_1, \cdots M_e\}$.
Randomly initialize SSD M.
for $t = 1, \cdots T$ **do**
    Training M by mini-batch optimization of (1) in the manuscript with X', Y.
end for
Obtain scores $S(x_1), \cdots, S(x_N)$ by a forward pass of X' into M and any outlier score.
Normalize $S(x_i)$ into $\bar{S}(x_i)$, $i = 1, \cdots, N$, such that $\sum_{i=1}^{N} \bar{S}(x_i) = 1$ and $\bar{S}(x_i) \geq 0$.
Set weight $w_i = \bar{S}(x_i)$, $i = 1, \cdots, N$.
for $j = 1, \cdots e$ **do**
    Initialize SSD model M_j with random seed s_j.
    for $t = 1, \cdots T$ **do**
        Training M_j by mini-batch optimization of (12) in the manuscript with X', Y, w_i, $i = 1, \cdots, N$.
    end for
end for

Net has four blocks for the encoder and four blocks for the decoder. Each block has a max-pooling or an upsampling operation, following two convolutional layers with kernel size 3. We use upsampling instead of deconvolution for efficiency. The ability to recover image details for upsampling is limited, so we add skip-connection operations to pass input details from top layers to bottom layers. A SGD optimizer with learning rate 0.1 is used to train the model of generative *E*^{3}*Outlier*. When it comes to contrastive *E*^{3}*Outlier*, we use ResNet18 as the backbone architecture $f(\cdot)$ as feature extractor, and a three-layer fully-connected network with 128 hidden nodes and 128 output nodes are used as the projection head $g(\cdot)$. To perform data augmentation and construct positive data pair, we exploit a composite of random color jitting, random image crop and random RGB-to-gray operation to construct the augmentation set $\mathcal{A}$. The simple rotation set $\mathcal{T}_{SR} = \{Rot(\cdot, (y-1) \cdot 90^\circ)\}_{y=1}^{4}$ is used as operation set $\mathcal{O}_c$ to expand the mini-batch. The τ of contrastive loss is set to 0.5. The DNN model of contrastive *E*^{3}*Outlier* is trained by 50, 100 and 200 epochs on SVHN, CIFAR10 and CIFAR100 dataset respectively, by a SGD optimizer with 10 epoch warm-up and cosine learning rate scheduler. The learning rate is set to be 0.03 for SVHN and 0.1 for CIFAR10/CIFAR100. As to the *E*^{3}*Outlier* based UVAD solution, the CAE is trained for 10 epochs with the default Adam optimizer in PyTorch. As to model architecture, we adopt a basic CAE architecture that consists of an encoder and a decoder. The encoder consists of three blocks, and each block contains a convolution layer (with kernel size 3, stride 2 and padding 1), a batch normalization (BN) layer and a ReLU activation layer. Similarly, the decoder consists of three blocks. For the first two blocks, they are both made up of a deconvolution layer (with kernel size 3, stride 2, padding 1, output padding 1), a BN layer and a ReLU layer, while the last block contains a single deconvolution layer. The transformation from patch to optical flow $OF(\cdot)$ is performed by a pretrained FlowNetv2 model [2].

As to competing methods, we adopt the deep CAE architecture from [3] with a 4-layer encoder and 4-layer decoder, which is estimated to have a close depth to the used WRN: $conv(k=3, s=2) - bn - Relu - conv(k=3, s=2) - bn - relu - conv(k=3, s=2) - bn - relu - reshape - fc(4096, 256) - tanh - fc(256, 4096) - bn - relu - reshape - deconv(k=3, s=2) - bn - relu - deconv(k=3, s=2) - bn - relu - deconv(k=3, s=2) - tanh$, while k and s refer to kernel size and stride. We do not use a more complex CAE architecture (e.g. CAE using skip connection [4] or more layers) since they usually lower outliers' reconstruction error as well and actually do not contribute to the OD performance. For each individual method, the parameters are set as follows: **(1)** CAE [5]. CAE is trained by Mean Square Error Loss (MSE) and its reconstruction loss is directly used to perform UOD. The CAE is trained by default Adam optimizer in PyTorch[3] for 250 epochs with learning rate 0.001 and weight decay 0.0005. The batch size is 128. **(2)** CAE-IF. CAE-IF is a decoupled/hybrid method that feeds the learned representations of CAE into isolation forest (IF) [6]. The training of CAE is the illustrated above, and the IF is realized by Scikit-learn framework[4]. The contamination parameter of IF is set by $p = \rho$ to yield better OD performance for comparison with *E*^{3}*Outlier*, and other parameters are set to default values in scikit-learn. **(3)** Discriminative reconstruction based autoencoder (DRAE) [7]. We set DRAE's encouraging term weight $\lambda = 0.1$ as recommended in [7], while other training setting is the same as CAE. **(4)** Deep Structured Energy based Models (DSEBM). For DSEBM, we use the implementation from [3], which trains the CAE used in DSEBM by 200 epochs and use the energy based score to perform OD. **(5)** Robust Deep Autoencoder (RDAE) [8]. We set $\lambda = 0.00065$ for RDAE's regularization, which performs best in the empirical evaluation of [8]. To yield the best performance, we use 20 outer epochs and 1 inner epochs for the alternating optimization. **(6)** Deep Autoencoding Gaussian Mixture Model (DAGMM) [9]. As suggested by [9], we adopt $\lambda_1 = 0.1$, $\lambda_2 = 0.005$ for the energy regularization term and the singularity penalty term respectively. An Adam optimizer with the recommended learning rate 0.0001 is used to optimize the CAE and density estimation network for 200 epochs. The batch size is 1024 as set in [9]. **(7)** Multiple-Objective Generative Adversarial Active Learning (MOGAAL). We strictly follow the original implementation from [10], with minor modifications to make MOGAAL applicable to image data: A deep convolutional GAN (DCGAN) is used to generate pseudo outliers, and 128-d random vectors and the batch-size 64 are used for image generation. The pseudo outlier generator and discriminator are jointly trained for 25 epochs, while the discriminator is then separately trained for 75 epochs to perform OD. **(8)** Robust Subspace Recovery based AE (RSRAE). We follow the parameter setup in [11]: The latent subspace dimension is set to be $d = 10$, and an Adam optimizer with learning rate $lr = 0.00025$ and weight decay $\lambda = 0.0005$ is used to optimize RSRAE. All threshold values are set as [11]. **(9)** For two-stage methods, the outputs of penultimate layer of pre-trained ResNet50 model are extracted as features, and they are used to train a LoF or

3. https://pytorch.org/
4. https://scikit-learn.org/

IF model. The contamination parameter of LoF or IF is set by $p = \rho$ to yield better OD performance for comparison with $E^3Outlier$, and other parameters are set to default values in scikit-learn. In general, the hyperparameters of the compared methods are set to recommended values (if provided) or the values that produce the best performance.

All experiments are run on a PC with dual NVIDIA Titan Xp GPUs, 64 GiB RAM and Intel 7820X CPU, under a programming environment with Python 3.6, PyTorch 0.4.1 and Keras 2.2.0. All implementation details can be found in our publicly available codes[5].

6 Clarification of Terms

In this section, we differentiate three terms that are often confused in the literature: Outlier detection (OD), out-of-distribution detection (OOD), (semi-supervised) anomaly detection (AD):

- OD is a long-standing problem [12] that handles completely *unlabeled* data, and it aims to detect those minority data that divert significantly from the majority data using some outlierness measures (e.g. proximity or density). Meanwhile, OD follows a *transductive* learning setup, i.e. OD directly computes outlier scores of all given unlabeled data, and it does not require a separated labeled training set to establish an inductive model. For example, as shown in Fig. 1a, without any labeled training data, two data clusters are likely to be viewed as inliers, while the rest of data that are distributed distantly are viewed as outliers. Thus, OD is a *fully-unsupervised* task.
- OOD an is emerging topic [13] that aims to determine whether an incoming datum is from the same data distribution of an trained model's training data set. It often follows an *inductive* learning setup, as it usually involves a labeled binary/multi-class training set to train an inductive model in a supervised manner. As the example shows in Fig. 1b, OOD leverages the labeled two-class data in the circle to train a binary classifier. It is supposed to classify newly-incoming data into two classes and exclude out-of-distribution data outside the training distribution. OOD differs from OD in two facets: 1) OOD often requires a separated labeled training set to know which data should be viewed as in-distribution data, while OD directly sorts out outliers from given unlabeled data by some outlierness measure. 2) The labeled binary/multi-class training set still provides abundant supervision information for OOD, and the OOD model (usually discriminative DNNs) can be easily trained in a *supervised* manner. It significantly facilitates OOD to learn more reasonable representations than deep OD.
- AD, which may also be referred as *novelty detection* or *one-class classification*, is another classic topic that aims to detect anomalies that are different from the labeled normal data. In fact, AD (rather than OD) is a highly similar *inductive* task to OOD, because it also requires a labeled training set to build a normality model, which is then used to discriminate anomalies or novelties in inference. However, AD's main difference from OOD is that its training data are usually labeled by one rough label ("normal" or "observed"). Due to the absence of subclass labels within training data, AD does not require classifying subclasses like OOD does during inference, and it is often viewed as a *semi-supervised* problem that aims to establish a valid description of appointed normal data domain (a.k.a data description [14]). Therefore, the labeling of normality domain plays an important role in AD: As shown in Fig. 1c and Fig. 1d, AD is expected to output completely different anomalies when the labeling of normality is different. In other words, the detected anomalies/novelties are often influenced by the definition of normality rather than the data distribution itself. This is different from OD that manifests outliers by some intrinsic data characteristics within the unlabeled dataset.

5. https://github.com/demonzyj56/E3Outlier

7 Details of OD Performance

We present the full results of OD performance comparison in Table 4-8, under outlier ratio 5%, 10%, 15%, 20% and 25% respectively. Each table contains Area Under ROC curve (ROC), PR curve with inliers to be the positive class (PR-I) and PR curve with outliers to be the positive class (PR-O) as evaluation metrics. Note that only performance of NE based outlier score is shown for $E^3Outlier$ due to the limit of space. In each table, the best OD performance on each benchmark is shown in bold.

References

[1] R. Anand, K. G. Mehrotra, C. K. Mohan, and S. Ranka, "An improved algorithm for neural network classification of imbalanced training sets," *IEEE Transactions on Neural Networks*, vol. 4, no. 6, pp. 962–969, 1993.

[2] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2462–2470.

[3] I. Golan and R. El-Yaniv, "Deep anomaly detection using geometric transformations," in *Advances in Neural Information Processing Systems*, 2018, pp. 9758–9769.

[4] X. Mao, C. Shen, and Y.-B. Yang, "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections," in *Advances in neural information processing systems*, 2016, pp. 2802–2810.

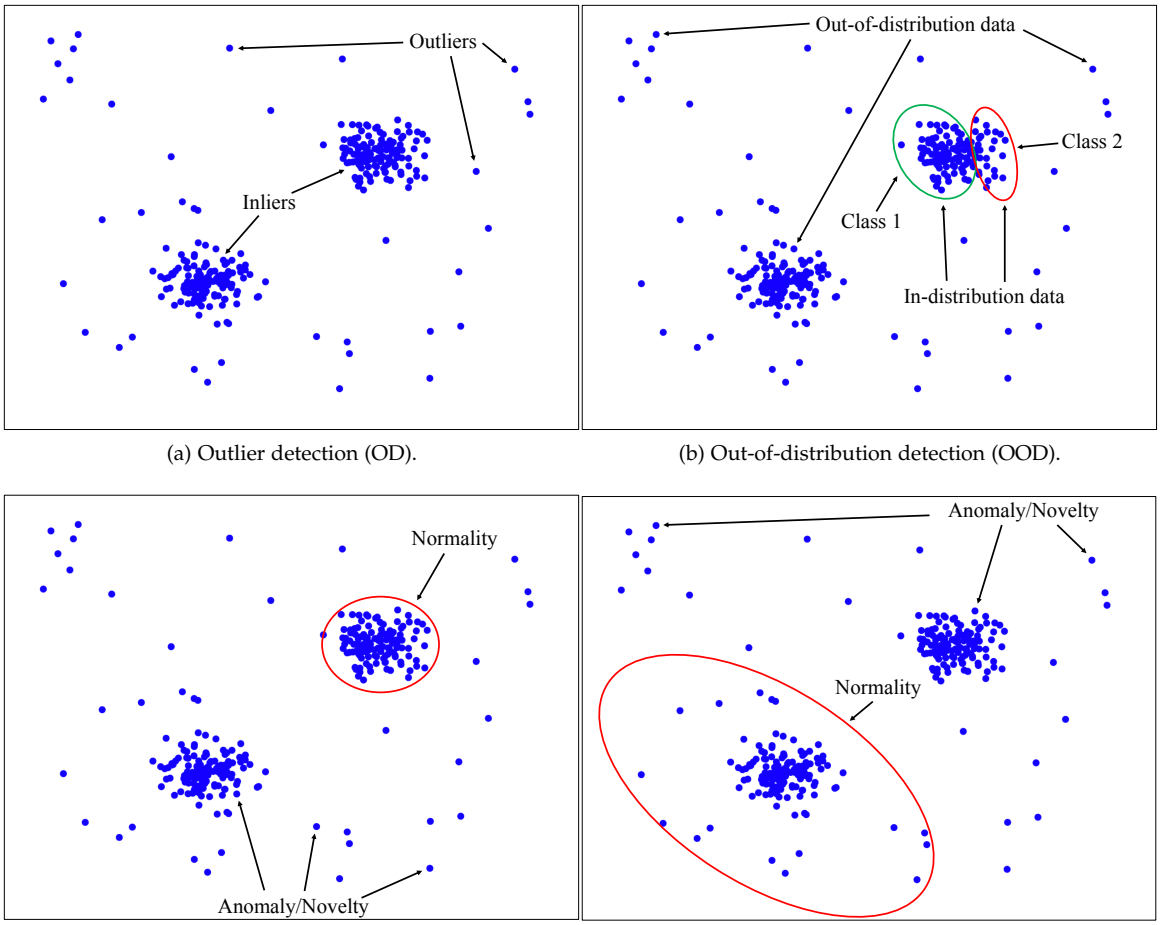
[5] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *International Conference on Artificial Neural Networks*. Springer, 2011, pp. 52–59.

[6] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 2008, pp. 413–422.

[7] Y. Xia, X. Cao, F. Wen, G. Hua, and J. Sun, "Learning discriminative reconstructions for unsupervised outlier removal," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1511–1519.

[8] C. Zhou and R. C. Paffenroth, "Anomaly detection with robust deep autoencoders," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017, pp. 665–674.

[9] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," in *International Conference on Learning Representations (ICLR)*, 2018.



(a) Outlier detection (OD).

(b) Out-of-distribution detection (OOD).

(c) Anomaly detection (AD) with upper right cluster labeled as normal (red circle).

(d) Anomaly detection (AD) with all lower left data labeled as normal (red circle).

Fig. 1: The comparison of outlier detection/out-of-distribution detection/anomaly detection formulation in this paper.

[10] Y. Liu, Z. Li, C. Zhou, Y. Jiang, J. Sun, M. Wang, and X. He, "Generative adversarial active learning for unsupervised outlier detection," *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[11] C. Lai, D. Zou, and G. Lerman, "Robust subspace recovery layer for unsupervised anomaly detection," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2020.

[12] A. Boukerche, L. Zheng, and O. Alfandi, "Outlier detection: Methods, models, and classification," *ACM Computing Surveys (CSUR)*, vol. 53, no. 3, pp. 1–37, 2020.

[13] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," 2017.

[14] D. M. Tax and R. P. Duin, "Support vector data description," *Machine learning*, vol. 54, no. 1, pp. 45–66, 2004.

TABLE 4: Performance comparison of E^3*Outlier* with baseline and state-of-the-art DNN based OD methods on benchmarks in terms of Area Under ROC curve, PR curve with inliers to be the positive class (PR-I) and PR curve with outliers to be the positive class (PR-O), under outlier ratio $\rho = 5\%$. The best performer is shown in bold font.

Dataset	MNIST			Fashion-MNIST			CIFAR10			SVHN			CIFAR100		
	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O
$\rho = 5\%$															
CAE	69.3	96.0	25.0	75.1	97.8	23.9	56.2	95.5	7.6	51.5	95.2	5.3	55.6	95.5	8.2
CAE+IF	87.3	99.1	39.2	84.8	98.8	33.1	54.2	95.0	7.3	55.4	95.8	6.1	54.3	95.3	7.4
DRAE	67.6	96.4	24.4	71.5	97.4	21.9	57.0	95.5	8.1	50.6	95.1	5.2	57.2	95.6	9.0
DSEBM	64.1	96.2	17.4	53.2	94.1	12.5	61.6	96.3	8.1	49.7	95.0	5.0	60.1	96.3	9.3
RDAE	74.0	96.8	29.4	78.9	98.3	26.3	55.9	95.2	7.7	52.2	95.3	5.4	56.1	95.5	8.6
DAGMM	60.7	96.1	18.0	64.1	95.5	25.3	55.1	95.5	8.7	49.7	95.0	15.8	54.4	95.5	8.2
MOGAAL	30.2	89.6	7.0	22.3	80.6	9.6	55.3	95.4	6.9	49.3	94.9	4.9	53.5	95.3	6.7
RSRAE	85.8	98.8	41.2	81.9	98.4	33.1	56.0	95.6	7.2	52.0	95.2	5.5	57.9	95.9	7.7
Res50+LoF	80.2	98.3	26.8	64.7	96.5	13.7	67.0	97.2	11.0	62.5	96.8	7.6	71.8	97.7	15.2
Res50+IF	84.4	98.9	33.6	85.3	98.9	37.6	65.1	97.0	9.6	57.8	96.1	6.6	68.0	97.3	12.1
SSD+IF	**95.5**	**99.7**	**65.1**	92.4	99.4	66.5	66.7	97.1	12.0	76.2	98.2	13.9	57.1	96.0	7.2
E^3Out. (G)	88.9	98.5	54.6	90.2	99.3	54.7	67.1	96.8	11.7	64.0	97.0	8.0	61.8	96.3	9.3
E^3Out. (D)	95.2	**99.7**	59.8	**94.1**	**99.6**	**70.4**	85.7	99.0	34.6	88.9	99.3	27.2	80.3	98.5	22.7
E^3Out. (C)	-	-	-	-	-	-	**91.6**	**99.5**	**45.8**	**93.6**	**99.6**	**44.9**	**84.1**	**99.0**	**24.0**

TABLE 5: Performance comparison of E^3*Outlier* with baseline and state-of-the-art DNN based OD methods on benchmarks in terms of Area Under ROC curve, PR curve with inliers to be the positive class (PR-I) and PR curve with outliers to be the positive class (PR-O), under outlier ratio $\rho = 10\%$. The best performer is shown in bold font.

Dataset	MNIST			Fashion-MNIST			CIFAR10			SVHN			CIFAR100		
	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O
$\rho = 10\%$															
CAE	68.0	92.0	32.9	70.3	94.3	29.3	55.8	91.0	14.4	51.2	90.3	10.6	55.2	91.0	14.5
CAE+IF	85.5	97.8	49.0	82.3	97.2	40.3	54.1	90.2	13.7	55.0	91.4	11.9	55.0	90.7	13.8
DRAE	66.9	93.0	30.5	67.1	93.9	25.5	56.0	90.7	14.7	51.0	90.3	10.5	55.6	90.9	15.0
DSEBM	60.5	91.6	23.0	53.2	88.9	19.7	60.2	92.3	14.7	50.0	90.0	10.1	59.2	92.2	16.2
RDAE	71.8	93.1	35.8	75.3	95.8	31.7	55.4	90.7	14.9	52.1	90.6	10.8	55.6	90.9	15.0
DAGMM	64.0	92.9	26.6	64.0	92.7	30.3	56.1	91.3	15.6	50.0	90.0	19.3	54.9	91.1	14.2
MOGAAL	30.9	78.8	15.2	22.8	74.8	14.8	56.2	91.1	13.6	49.0	89.7	9.8	53.2	90.4	12.6
RSRAE	84.8	97.4	45.4	78.3	96.2	37.0	56.6	91.4	14.0	51.5	90.3	10.6	57.1	91.6	14.1
Res50+LoF	71.2	97.5	26.6	57.8	96.2	16.9	59.9	91.4	17.4	61.3	90.3	14.0	69.1	94.6	22.2
Res50+IF	83.4	97.5	43.3	82.7	97.3	43.8	64.8	93.8	17.9	57.4	92.0	12.8	67.5	94.3	21.0
SSD+IF	93.8	99.2	**68.7**	90.6	98.5	68.6	64.0	93.5	18.3	73.4	95.9	22.0	55.6	91.5	13.0
E^3Out. (G)	86.7	96.4	60.3	89.6	98.5	61.6	66.3	93.5	20.0	63.6	93.9	15.0	61.2	92.4	16.7
E^3Out. (D)	**94.1**	**99.3**	67.5	**93.3**	**99.0**	**75.9**	83.5	97.5	43.4	86.0	98.0	36.7	79.2	96.8	33.3
E^3Out. (C)	-	-	-	-	-	-	**89.0**	**98.5**	**53.2**	**90.1**	**98.5**	**51.3**	**84.1**	**97.8**	**38.0**

TABLE 6: Performance comparison of E^3*Outlier* with baseline and state-of-the-art DNN based OD methods on benchmarks in terms of Area Under ROC curve, PR curve with inliers to be the positive class (PR-I) and PR curve with outliers to be the positive class (PR-O), under outlier ratio $\rho = 15\%$. The best performer is shown in bold font.

Dataset	MNIST			Fashion-MNIST			CIFAR10			SVHN			CIFAR100		
	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O
$\rho = 15\%$															
CAE	63.9	86.7	34.8	68.5	90.7	34.6	56.1	86.7	20.6	51.2	85.4	15.8	54.9	86.4	20.4
CAE+IF	84.5	96.2	54.9	79.9	94.8	45.5	54.0	85.4	19.7	54.3	86.7	17.2	53.7	85.7	19.5
DRAE	66.5	89.5	35.0	68.2	91.1	33.3	56.3	86.4	21.1	51.0	85.5	15.6	56.1	86.5	21.6
DSEBM	63.6	88.5	32.6	55.0	84.9	26.4	60.0	88.4	21.1	50.1	85.0	15.2	58.7	88.0	22.6
RDAE	67.0	88.1	37.2	72.1	92.4	36.6	55.1	86.0	20.1	52.2	85.9	16.1	55.0	86.2	21.0
DAGMM	62.0	88.5	31.5	58.0	83.7	42.8	55.4	86.5	20.5	50.0	84.1	24.2	54.2	86.3	19.7
MOGAAL	35.1	73.2	21.2	29.6	73.4	19.6	54.6	86.2	18.9	49.2	84.6	14.8	53.3	85.7	18.8
RSRAE	80.4	93.9	49.2	78.1	94.1	43.5	55.7	86.5	20.3	51.3	85.4	15.8	56.9	87.3	19.9
Res50+LoF	66.2	90.0	28.4	55.0	85.6	20.7	61.5	89.4	22.9	60.3	89.2	19.8	67.1	91.4	27.8
Res50+IF	82.4	96.0	50.4	82.3	95.7	50.7	63.9	90.3	24.1	57.1	87.7	18.6	65.9	90.8	26.9
SSD+IF	92.1	98.4	70.1	89.3	97.3	70.5	62.6	89.6	24.2	71.2	93.0	28.4	55.5	87.1	18.7
E^3Out. (G)	84.5	93.1	66.7	84.6	96.3	62.2	64.7	89.5	26.3	63.4	90.5	21.9	60.3	88.2	23.0
E^3Out. (D)	**92.9**	**98.6**	**70.6**	**92.3**	**98.2**	**78.2**	81.3	95.6	48.6	83.3	96.2	42.5	78.1	94.7	40.7
E^3Out. (C)	-	-	-	-	-	-	**86.4**	**97.0**	**57.5**	**87.3**	**96.9**	**55.1**	**83.7**	**96.6**	**46.8**

TABLE 7: Performance comparison of *E*^{3}*Outlier* with baseline and state-of-the-art DNN based OD methods on benchmarks in terms of Area Under ROC curve, PR curve with inliers to be the positive class (PR-I) and PR curve with outliers to be the positive class (PR-O), under outlier ratio $\rho = 20\%$. The best performer is shown in bold font.

Dataset	MNIST			Fashion-MNIST			CIFAR10			SVHN			CIFAR100		
	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O
$\rho = 20\%$															
CAE	64.0	82.7	40.7	64.4	85.3	36.8	54.7	81.6	25.5	50.7	80.2	20.7	54.4	81.7	25.6
CAE+IF	81.5	93.6	57.2	77.8	92.2	49.0	53.8	80.7	25.3	54.0	82.0	22.4	53.5	80.9	25.1
DRAE	67.3	86.6	42.5	65.7	86.9	36.6	55.6	81.7	26.8	50.6	80.4	20.5	55.5	81.8	27.0
DSEBM	56.3	81.2	32.3	53.1	79.6	31.7	61.4	85.2	27.8	50.2	80.3	20.2	57.9	83.7	27.8
RDAE	67.0	89.2	43.2	70.9	89.2	41.4	54.2	81.0	25.7	51.8	80.9	21.1	54.9	81.5	26.5
DAGMM	65.9	86.7	41.3	66.0	86.7	43.5	54.7	81.8	26.3	50.0	79.9	29.6	53.8	81.5	24.7
MOGAAL	37.8	70.6	28.0	34.0	66.6	28.3	55.7	82.0	25.0	49.6	79.8	19.8	53.1	80.9	24.4
RSRAE	78.9	91.3	53.0	74.5	90.4	46.3	55.6	82.1	25.8	51.1	80.3	21.0	56.3	82.7	25.2
Res50+LoF	62.4	84.9	31.0	53.4	80.3	24.9	63.6	84.9	27.9	59.3	85.0	25.2	65.3	87.5	32.6
Res50+IF	79.8	93.6	52.1	80.7	93.5	55.0	63.4	86.6	30.4	56.8	83.3	24.2	64.7	87.1	32.4
SSD+IF	90.5	97.3	71.0	87.6	95.6	71.4	60.2	85.0	28.3	69.2	89.5	33.7	54.3	82.1	23.4
E^{3}Out. (G)	83.2	90.4	67.9	85.3	95.2	66.4	64.5	85.7	33.0	62.8	86.8	27.9	59.6	83.8	28.6
E^{3}Out. (D)	**91.3**	**97.6**	**72.3**	**91.2**	**97.1**	**78.9**	79.3	93.1	52.7	81.0	93.4	47.0	77.0	92.4	46.5
E^{3}Out. (C)	-	-	-	-	-	-	**83.6**	**94.8**	**59.0**	**84.8**	**94.9**	**57.6**	**82.9**	**95.1**	**53.0**

TABLE 8: Performance comparison of *E*^{3}*Outlier* with baseline and state-of-the-art DNN based OD methods on benchmarks in terms of Area Under ROC curve, PR curve with inliers to be the positive class (PR-I) and PR curve with outliers to be the positive class (PR-O), under outlier ratio $\rho = 25\%$. The best performer is shown in bold font.

Dataset	MNIST			Fashion-MNIST			CIFAR10			SVHN			CIFAR100		
	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O	ROC	PR-I	PR-O
$\rho = 25\%$															
CAE	60.9	77.3	42.7	62.8	80.9	41.2	54.2	76.9	30.5	50.7	75.2	25.7	54.6	77.3	31.1
CAE+IF	79.5	90.6	59.7	76.5	89.2	53.0	53.6	76.0	30.6	53.6	77.1	27.5	53.3	76.1	30.1
DRAE	64.2	81.2	44.9	65.3	83.0	41.4	55.4	77.1	32.0	50.5	75.3	25.5	55.1	77.0	32.0
DSEBM	62.9	80.8	43.3	53.8	75.0	38.0	60.0	80.4	32.8	50.2	75.1	25.3	56.7	78.6	32.6
RDAE	65.4	79.7	47.0	69.6	85.7	45.1	54.0	76.4	31.0	51.6	75.9	26.2	54.5	76.7	31.6
DAGMM	60.5	80.1	40.9	58.3	78.6	43.3	54.1	76.6	31.1	50.2	75.0	36.0	52.8	76.2	29.4
MOGAAL	38.9	65.5	35.7	35.8	62.4	34.2	54.4	76.7	29.5	49.4	74.5	24.7	53.0	76.1	29.6
RSRAE	76.6	87.5	56.4	72.6	86.5	50.8	55.7	77.6	31.6	51.1	75.4	26.0	55.5	77.9	30.3
Res50+LoF	59.8	79.5	34.1	52.5	75.1	29.2	58.6	80.0	32.7	58.4	80.5	30.2	63.8	83.2	37.1
Res50+IF	79.1	91.3	57.3	79.6	91.2	57.5	62.0	82.2	34.6	56.0	78.5	29.3	64.2	83.2	37.8
SSD+IF	88.5	95.6	72.0	85.9	93.5	72.1	59.0	80.2	32.9	67.0	85.3	38.0	53.7	77.2	28.3
E^{3}Out. (G)	80.2	85.8	69.1	79.2	90.5	64.1	63.6	81.4	38.3	62.3	83.0	33.5	59.4	79.6	34.2
E^{3}Out. (D)	**89.8**	**96.2**	**73.7**	**89.6**	**95.6**	**78.7**	77.4	90.0	55.7	78.8	91.0	51.0	76.0	89.7	51.3
E^{3}Out. (C)	-	-	-	-	-	-	**80.8**	**91.5**	**61.3**	**80.6**	**91.5**	**58.1**	**82.8**	**93.6**	**58.8**

RE: "E^3*Outlier*: A Self-supervised Framework for Unsupervised Deep Outlier Detection"
Manuscript Type: Regular
Authors: Siqi Wang, Yijie Zeng, Guang Yu, Zhen Cheng, Xinwang Liu, Sihang Zhou, En Zhu, Marius Kloft, Jianping Yin, Qing Liao

Dear Editors and Reviewers,

This manuscript is a significantly extended version of our conference paper published in Annual Conference on Neural Information Processing Systems (NeurIPS), 2019. The detailed information of the conference paper is shown in [1], and its published version is submitted with this manuscript as a supplementary material. The main improvements of the manuscript are summarized as follows:

1) Compared with the conference version that only explores the discriminative learning paradigm for deep outlier detection (OD), we further design two deep OD solutions that leverage generative learning paradigm (see *generative* E^3*Outlier* in Sec. 3.6.1) and contrastive learning paradigm (see *contrastive* E^3*Outlier* in Sec. 3.6.2) to provide self-supervision. Generative E^3*Outlier* can not only can not only use the same CAE architecture to achieve evidently superior OD performance to existing CAE based deep OD solutions, but also enables more flexible application of E^3*Outlier* to other scenarios like data with multiple modalities/views. Contrastive E^3*Outlier* is able to produce evident performance gain (up to 4% to 6% AUROC) on relatively difficult benchmark datasets, i.e. CIFAR10/SVHN/CIFAR100. In this way, we extend the proposed E^3*Outlier* from a specific single deep OD solution to a stronger and more general self-supervised deep OD framework.
2) In addition to the outlier image removal task in conference verision, we design a new E^3*Outlier* based solution to the unsupervised video abnormal event detection (UVAD) task (see the new Sec. 4.3), which is another important application of deep OD. We conduct experiments on three commonly-used video benchmark datasets, and our solution significantly outperforms state-of-the-art UVAD solutions by about 4% to 10% AUROC, which demonstrates the flexibility and effectiveness of the proposed E^3*Outlier*.
3) A separated new section (Sec. 3.4) is added to discuss the usage of network uncertainty as a new outlierness measure for outlier detection. This section first analyzes the drawbacks of the baseline outlier score and the motivation for improvement (Sec. 3.4.1), then it illustrates the underlying connections among outlier detection, self-supervised learning and network uncertainty (Sec. 3.4.2), and devises several network uncertainty based outlier scores (Sec. 3.4.3). Compared with the conference version, this manuscript not only elucidates the insights why network uncertainty based outlier scores are better than baseline outlier score, but also points out the direction to design the new outlier score like MC-Dropout (MCD) score.
4) We propose joint score refinement (Sec. 3.5) based on the new re-weighting and ensemble strategy, which can produce consistent performance improvement for discriminative E^3*Outlier*. The new re-weighting strategy (Sec. 3.5.2) can further magnify the inlier priority during the self-supervised learning. The ensemble strategy (Sec. 3.5.3) is expected to enhance the OD performance by improving the network uncertainty estimation. The way to combine the re-weighting and ensemble for joint score refinement is presented in Sec. 3.5.4.
5) Recent deep outlier detection methods (MOGAAL and RSRAE) are included for comparison with the proposed method. They are published after the submission of NeurIPS paper, so they are absent in the conference version. Besides, two-stage baseline solutions that combines pretrained DNN feature extractor and classic OD models (ResNet50+LoF and ResNet50+IF) are also included for a more comprehensive comparison.
6) The detailed theoretical analysis is provided in Sec. 3.3.2 of the manuscript to demonstrate the effects of inlier priority during the self-supervised learning, which is absent in the conference version.
7) More comprehensive introduction and literature review are provided in Sec. 1 and Sec. 2.
8) More experiments and discussion are conducted and presented in Sec. 4.

REFERENCES

[1] Siqi Wang, Yijie Zeng, Xinwang Liu, En Zhu, Jianping Yin, Chuanfu Xu, and Marius Kloft. Effective end-to-end unsupervised outlier detection via inlier priority of discriminative network. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 5960–5973, 2019.

Effective End-to-end Unsupervised Outlier Detection via Inlier Priority of Discriminative Network

Siqi Wang[1]*, **Yijie Zeng**[2]*, **Xinwang Liu**[1], **En Zhu**[1], **Jianping Yin**[3], **Chuanfu Xu**[1], **Marius Kloft**[4]
[1]National University of Defense Technology, [2]Nanyang Technological University
[3]Dongguan University of Technology, [4]Technische Universität Kaiserslautern
wangsiqi10c@nudt.edu.cn, yzeng004@e.ntu.edu.sg, {xinwangliu, enzhu}@nudt.edu.cn
jpyin@dgut.edu.cn, xuchuanfu@nudt.edu.cn, kloft@cs.uni-kl.de

Abstract

Despite the wide success of deep neural networks (DNN), little progress has been made on end-to-end unsupervised outlier detection (UOD) from high dimensional data like raw images. In this paper, we propose a framework named E^3*Outlier*, which can perform UOD in a both ***e**ffective* and ***e**nd-to-**e**nd* manner: First, instead of the commonly-used autoencoders in previous end-to-end UOD methods, E^3*Outlier* for the first time leverages a discriminative DNN for better representation learning, by using *surrogate supervision* to create multiple pseudo classes from original unlabelled data. Next, unlike classic UOD that utilizes data characteristics like density or proximity, we exploit a novel property named *inlier priority* to enable end-to-end UOD by discriminative DNN. We demonstrate theoretically and empirically that the intrinsic class imbalance of inliers/outliers will make the network prioritize minimizing inliers' loss when inliers/outliers are indiscriminately fed into the network for training, which enables us to differentiate outliers directly from DNN's outputs. Finally, based on inlier priority, we propose the negative entropy based score as a simple and effective outlierness measure. Extensive evaluations show that E^3*Outlier* significantly advances UOD performance by up to 30% AUROC against state-of-the-art counterparts, especially on relatively difficult benchmarks.

1 Introduction

An outlier is defined as "an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism" [1]. In some context of the literature, outliers are also referred as anomalies, deviants, novelties or exceptions [2]. Outlier detection (OD) has broad applications such as financial fraud detection [3], intrusion detection [4], fault detection [5], etc. Various solutions have been proposed to tackle OD (see [6] for a comprehensive review). Based on the availability of labels, those solutions can be accordingly divided into three categories below [7]: **1)** *Supervised* OD (SOD) deals with the case where a training set is provided with both labelled inliers/outliers, but it suffers from expensive data labelling and the rarity of outliers in practice [6]. **2)** *Semi-supervised* OD (SSOD) only requires pure single-class training data that are labelled as "inlier" or "normal", and no outlier is involved during training. **3)** *Unsupervised* OD (UOD) handles completely unlabelled data mixed with outliers, and no data label is provided for training at all.

In this paper we will limit our discussion to **UOD**, as most data are unlabelled in practice and UOD is the most widely applicable [7]. In particular, two ***clarifications of concepts*** must be made: First, in some literature like [8, 9], "unsupervised outlier/anomaly detection" actually refers to SSOD rather than UOD by our definition. Second, a recent topic is *out-of-distribution sample detection*, which

*Authors contribute equally.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

detects samples that are not from the distribution of training samples [10, 11, 12]. It is similar to SSOD, but it requires well-labelled multi-class data for training rather than single-class data in SSOD. Both cases above are different from UOD that does not use any label information in this paper.

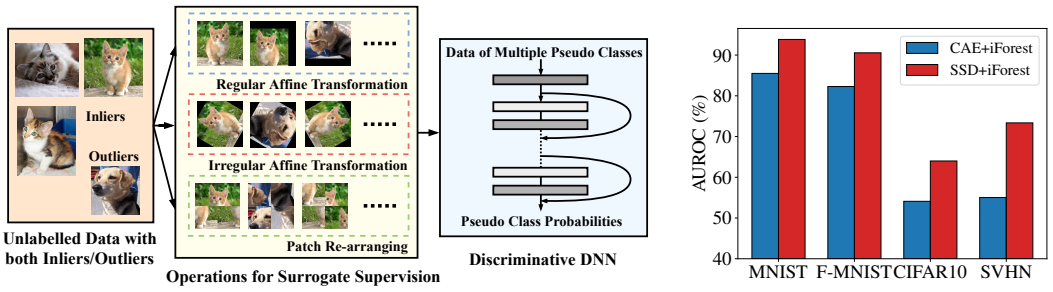
Recently, surging image/video data have inspired important UOD applications in computer vision, e.g. refining web image query results [13] and video abnormal event detection [14]. Unfortunately, despite the remarkable success of end-to-end deep neural networks (DNN) in computer vision [15], an *effective* and *end-to-end* UOD strategy is still under exploration: State-of-the-art methods [16, 17, 18] unexceptionally rely on deep autoencoders (AE) or convolutional autoencoders (CAE) to realize easily achievable DNN based UOD, but they all suffer from AE/CAE's ineffective representation learning (detailed in Sec. 3.1). Motivated by this gap, we aim to address UOD in a both effective and end-to-end fashion, with the application to detect outlier images from contaminated datasets.

Contributions. This paper proposes an **e**ffective and **e**nd-to-**e**nd UOD framework named E^3*Outlier*. Specifically, our contributions can be summarized below: **1)** To liberate DNN based UOD from AE/CAE's ineffective representation learning, E^3*Outlier* for the first time enables us to adopt powerful discriminative DNN architectures like ResNet [19] for representation learning in UOD. This is realized by *surrogate supervision*, which creates multiple pseudo classes by imposing various simple operations on original unlabelled data. **2)** E^3*Outlier* discovers outliers based on a novel property of discriminative network named *inlier priority*, which evidently differs from previous methods that utilize certain data characteristics (e.g. density, proximity, distance) to perform UOD. Through both theory and experiments, we demonstrate that inlier priority will encourage the network to prioritize the reduction of inliers' loss during network training. On the foundation of inlier priority, E^3*Outlier* is able to achieve end-to-end UOD by directly inspecting the DNN's outputs, which reflect each datum's priority level. In this way, it avoids the possible suboptimal performance yielded by feeding the DNN's learned representations into a decoupled UOD method [20]. **3)** Based on inlier priority, we explore several strategies and propose a simple and effective negative entropy based score to measure outlierness. Extensive experiments report a remarkable improvement by E^3*Outlier* against state-of-the-art methods, particularly on relatively difficult benchmarks for unsupervised tasks.

2 Related Work

Classic Outlier Detection. For classic SOD, labelled data are utilized to build discriminative models by well-studied supervised binary/multi-class classification techniques, such as support vector machine (SVM) [21], random forest [22] and recent XGBoost [23]. In contrast, SSOD that requires only labelled inliers is much more prevalent, and it is also called *one-class classification* [24] or *novelty detection* [25]. Classic SSOD usually involves training a model on pure inliers and detecting those data that evidently deviate from this model as outliers, and representative SSOD methods include SVM based methods [26, 27], replicator network/autoencoders [28, 29], principle component analysis (PCA)/kernel based PCA [30, 31]. Compared with SOD and SSOD, UOD handles the most challenging case where no labelled data is available. Classic UOD methods discover outliers by examining the basic characteristics of data, such as statistical properties [32], cluster membership [33, 34], density [35, 36, 37], proximity [38, 39], etc. Besides, ensemble methods like isolation forest [40] and its variants [41, 42] are popular in UOD. However, most state-of-the-art UOD methods like [40, 37, 13] still require manual feature extraction from high dimensional data like raw images.

DNN based Outlier Detection. DNN's recent success naturally inspires DNN based OD [20]. For SOD, discriminative DNN can be directly applied, while the main issue is the class imbalance of inliers/outliers [20], which is explored by [43, 44, 45, 46]. For SSOD, the case is more difficult as only labelled inliers are provided. DNN solutions for SSOD fall into three types: Mainstream DNN based SSOD methods handle high dimensional data by label-free generative models, i.e. AE/CAE [47, 48, 49, 50] and generative adversarial network (GAN) [51, 52, 53]. The second type extends classic SSOD methods into their deep counterparts, such as deep support vector data description [54] and deep one-class SVM [55]. The last type turns SSOD into SOD by certain means like introducing reference datasets [56], intra-class splitting [57], geometric transformations [58] or synthetic outlier generation [59]. As to UOD, the absence of both inlier and outlier label poses great challenges to combining UOD with DNN, which results in much less progress than SOD and SSOD. In addition to the naive solution that feeds DNN's learned representations into a separated UOD method [20], to our best knowledge only the following works have explored DNN based UOD: Zhou et al. [17] propose a decoupled solution that combines a deep AE with Robust PCA, which decomposes the inputs into a



(a) (b)

Figure 1: Surrogate supervision workflow (left) and the comparison of learned representations (right).

low-rank part from inliers and a sparse part from outliers; For end-to-end UOD, Xia et al. [16] use deep AE directly and propose a variant that estimates inliers by seeking a threshold that maximizes the inter-class variance of AE's reconstruction loss. A loss function is designed to encourage the separation of estimated inliers/outliers; Zong et al. [18] jointly optimize a deep AE and an estimation network to perform simultaneous representation learning and density estimation for end-to-end UOD.

Surrogate Supervision. Recent studies propose surrogate supervision to improve DNN pre-training for downstream high-level tasks like image classification and object detection. It imposes certain operations on unlabelled data to create corresponding pseudo classes and provide supervision signal, such as rotation [60], image patch permutation [61], clustering [62], etc. Surrogate supervision is also called self-supervision (see [63] for a comprehensive survey), but we use surrogate supervision to better distinguish it from AE/CAE, which are also viewed as "self-supervised" in some context. To our best knowledge, our work is the first to connect surrogate supervision with end-to-end UOD.

3 The proposed *E*^{3}*Outlier* Framework

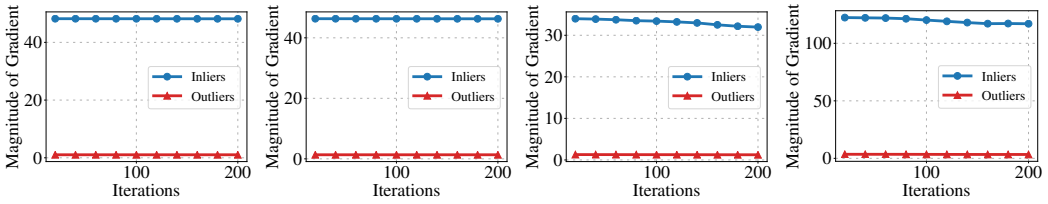
Problem Formulation of UOD. Considering a data space $\mathcal{X}$ (in this context the space of images), an unlabelled data collection $X \subseteq \mathcal{X}$ consists of an inlier set X_{in} and an outlier set X_{out}, which originate from fundamentally different underlying distributions [1]. Our goal is to obtain an end-to-end UOD method $S(\cdot)$ that in the ideal case outputs $S(\mathbf{x}) = 1$ for inlier $\mathbf{x} \in X_{in}$ and $S(\mathbf{x}) = 0$ for outlier $\mathbf{x} \in X_{out}$. In practice, a smaller $S(\mathbf{x})$ indicates a higher likelihood of $\mathbf{x}$ to be an outlier.

3.1 Surrogate Supervision Based Effective Representation Learning for UOD

Why NOT AE/CAE? We note that existing DNN based UOD methods rely on AE/CAE [16, 17, 18]. However, it is hard for them to handle relatively complex datasets like CIFAR10 and SVHN: As our UOD experiments[2] show in Fig. 1(b), even a sophisticated deep CAE with isolation forest [40] only performs slightly better than random guessing (50% AUROC). Similar results are reported in other AE/CAE based unsupervised tasks like deep clustering [64, 65]. This is because AE/CAE typically adopt mean square error (MSE) as loss function, which forces AE/CAE to focus on reducing low-level pixel-wise error that is not sensitive to human perception, rather than learning high-level semantic features [66, 67]. Therefore, AE/CAE based representation learning is often ineffective.

Surrogate Supervision. Discriminative DNNs like ResNet [19] and Wide ResNet (WRN) [68] have proved to be highly effective in learning high-level semantic features, but they have not been explored in UOD due to the lack of supervision. To remedy the absence of data labels and substitute AE/CAE, we propose a *surrogate supervision based discriminative network* (SSD) for more effective representation learning in UOD. Specifically, we first define an operation set with K operations $\mathcal{O} = \{O(\cdot|y)\}_{y=1}^{K}$, where y represents the pseudo label associated with the operation $O(\cdot|y)$. Applying an operation $O(\cdot|y)$ to $\mathbf{x}$ can generate a new datum $\mathbf{x}^{(y)} = O(\mathbf{x}|y)$, and all data generated by the operation $O(\cdot|y)$ belong to the pseudo class with pseudo label y. Next, given a datum $\mathbf{x}^{(y')}$, a discriminative DNN with a K-node softmax layer is trained to classify the type of applied

[2] All UOD experiments in Sec. 3 follow the setup detailed in Sec. 4.1 and the outlier ratio is fixed to 10%.



(a) MNIST ("3") (b) F-MNIST ("bag") (c) CIFAR10 ("horse") (d) SVHN ("3")

Figure 2: Inliers and outliers' gradient magnitude on example cases of benchmark datasets during SSD training. The class used as inliers is in brackets.

operation, i.e. the DNN is supposed to classify $\mathbf{x}^{(y')}$ into the y'-th pseudo class. With $P^{(y)}(\cdot)$ and $\boldsymbol{\theta}$ denoting the probability output by the y-th node of softmax layer and DNN's learnable parameters respectively, DNN's output probability vector for K operations is $P(\mathbf{x}^{(y')}|\boldsymbol{\theta}) = [P^{(y)}(\mathbf{x}^{(y')}|\boldsymbol{\theta})]_{y=1}^{K}$. To train such a DNN with an unlabelled data collection $X = \{\mathbf{x}_i\}_{i=1}^{N}$, the objective function is:

$$\min_{\boldsymbol{\theta}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{SS}(\mathbf{x}_i|\boldsymbol{\theta}) \tag{1}$$

where $\mathcal{L}_{SS}(\mathbf{x}_i|\boldsymbol{\theta})$ is the loss incurred by $\mathbf{x}_i$ under surrogate supervision. When the commonly-used cross entropy loss is used to classify pseudo classes of surrogate supervision, it can be written as:

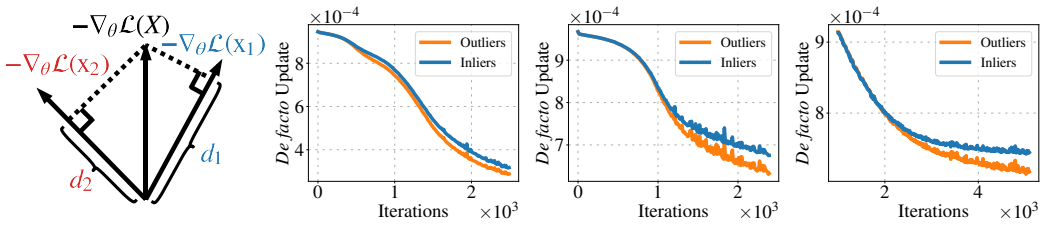
$$\mathcal{L}_{SS}(\mathbf{x}_i|\boldsymbol{\theta}) = -\frac{1}{K} \sum_{y=1}^{K} \log(P^{(y)}(\mathbf{x}_i^{(y)}|\boldsymbol{\theta})) = -\frac{1}{K} \sum_{y=1}^{K} \log(P^{(y)}(O(\mathbf{x}_i|y)|\boldsymbol{\theta})). \tag{2}$$

As to the operation set $\mathcal{O}$, each operation $O(\cdot|y) \in \mathcal{O}$ is defined as a combination of one or more basic transformations from the following transformation sets: **1)** *Rotation*: This set's transformations clock-wisely rotate images by a certain degree. **2)** *Flip*: This set's transformations refer to flipping the image or not. **3)** *Shifting*: This set's transformations shift the image by some pixels along x-axis or y-axis. **4)** *Patch re-arranging*: This set's transformations partition the image into several equally-sized patches and re-organize them into a new image by a certain permutation. Based on them, we construct three operation subsets, i.e. regular affine transformation set $\mathcal{O}_{RA}$, irregular affine transformation set $\mathcal{O}_{IA}$ and patch re-arranging set $\mathcal{O}_{PR}$ (detailed in Sec.1 in supplementary material). The final operation set is $\mathcal{O} = \mathcal{O}_{RA} \cup \mathcal{O}_{IA} \cup \mathcal{O}_{PR}$, and Fig. 1(a) shows SSD's entire workflow. To verify SSD's effectiveness, we extract the outputs of its penultimate layer as the learned representations, while the outputs of deep CAE's intermediate hidden layer (with the same dimension as SSD) are used for comparison. We feed them into isolation forest [40], which is generally acknowledged to be a good UOD method [69], to perform UOD under the same parameterization. As shown in Fig. 1(b), SSD's learned representations are able to outperform CAE by a large magrin (8%-10% AUROC).

3.2 Inlier Priority: The Foundation of End-to-end UOD

Motivation. The above simple solution feeds SSD's learned representations into a decoupled UOD method, which may yield suboptimal performance because SSD and the UOD method are trained separately [18, 20]. Our goal is to achieve end-to-end UOD without using a decoupled UOD method. Recall that outliers are essentially rare patterns in a data collection [7], which implies an intrinsic *class imbalance* between inliers/outliers. Class imbalance is unfavorable in machine learning as it leads to the bias towards majority class during training [70, 71]. However, we argue that class imbalance can be favorably exploited in UOD as it gives rise to "*inlier priority*": ***Despite that inliers/outliers are indiscriminately fed into SSD for training, SSD will prioritize the minimization of inliers' loss***. This intuition naturally inspires an end-to-end UOD solution by measuring how well the SSD's output of a datum matches its target pseudo label, which directly indicates its priority level in training and the likelihood to be an inlier. We demonstrate the inlier priority in terms of two aspects below:

Priority by Gradient Magnitude. *Our first point is that inliers will produce gradient with stronger magnitude to update the SSD network than outliers*. To demonstrate this point, we consider an SSD



(a) *De facto* update (b) MNIST ("3") (c) F-MNIST ("bag") (d) CIFAR10 ("horse")

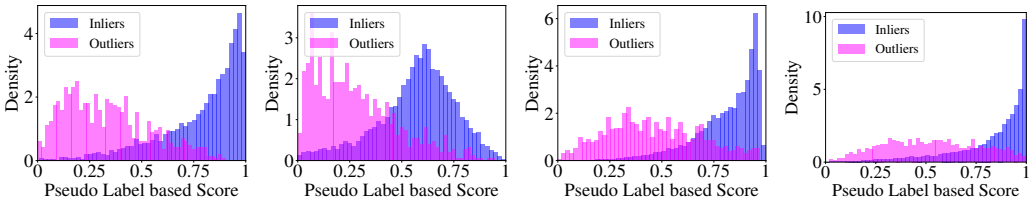
Figure 3: An illustration of *de facto* update and some example cases of the average *de facto* update for inliers/outliers during the network training. The class used as inliers is in brackets.

with its network weights randomly initialized by i.i.d. uniform distribution on $[-1, 1]$. Without loss of generality, we consider the gradients w.r.t. the weights associated with the c-th class ($1 \leq c \leq K$) between the penultimate layer and softmax layer, $\mathbf{w}_c = [w_{s,c}]_{s=1}^{(L+1)}$ ($w_{L+1,c}$ is bias), because these weights are directly responsible for making predictions. For the commonly-used cross-entropy loss $\mathcal{L}$, only data transformed by the c-th operation $X^{(c)} = \{O(\mathbf{x}|c)|\mathbf{x} \in X\}$ are used to update $\mathbf{w}_c$. The gradient vector incurred by $\mathcal{L}$ is denoted by $\nabla_{\mathbf{w}_c}\mathcal{L} = [\nabla_{w_{s,c}}\mathcal{L}]_{s=1}^{(L+1)}$, which will be used to update $\mathbf{w}_c$ in back-propagation based optimizer like Stochastic Gradient Descent (SGD) [72]. Given unlabelled data with N_{in} inliers and N_{out} outliers, it is easy to know that $X^{(c)}$ also contains N_{in} transformed inliers and N_{out} transformed outliers. Here we are interested in the magnitude of transformed inliers and outliers' aggregated gradient to update $\mathbf{w}_c$, i.e. $||\nabla_{\mathbf{w}_c}^{(in)}\mathcal{L}||$ and $||\nabla_{\mathbf{w}_c}^{(out)}\mathcal{L}||$, which directly reflect inliers/outliers' strength to affect the training of SSD. Since SSD is randomly initialized, we need to compute the expectation of gradient magnitude. As shown in Sec. 2 of supplementary material, for a simplified SSD network with a single hidden-layer and sigmoid activation, we can quantitatively derive the following approximation on inliers and outliers' gradient magnitude:

$$\frac{E(||\nabla_{\mathbf{w}_c}^{(in)}\mathcal{L}||^2)}{E(||\nabla_{\mathbf{w}_c}^{(out)}\mathcal{L}||^2)} \approx \frac{N_{in}^2}{N_{out}^2} \tag{3}$$

where $E(\cdot)$ denotes the probability expectation. As the class imbalance between inliers and outliers leads to $N_{in} \gg N_{out}$, we naturally yield $E(||\nabla_{\mathbf{w}_c}^{(in)}\mathcal{L}||) \gg E(||\nabla_{\mathbf{w}_c}^{(out)}\mathcal{L}||)$. Therefore, it serves as a theoretical indication that *the gradient magnitude induced by inliers will be significantly larger than outliers for an untrained SSD network*. Since it is particularly difficult to directly analyze more complex network architectures such as Wide ResNet [68], we empirically examine inliers and outliers' gradient magnitude during training by experiments (see Fig. 2), and the observations on different benchmarks are consistent with the above analysis on the simplified case: The magnitude of inliers' aggregated gradient has constantly been larger than outliers during the process of SSD training.

Priority by Network Updating Direction. *Our second point is that the network updating direction of SSD will bias towards the direction that prioritizes reducing inliers' loss during the SSD training*. Since training is dynamic and a theoretical analysis is intractable, we demonstrate this point using an empirical verification by computing inliers/outliers' average "*de facto* update": As illustrated by Fig. 3(a), consider a datum $\mathbf{x}_i$ from a batch of data X, and its negative gradient $-\nabla_{\boldsymbol{\theta}}\mathcal{L}(\mathbf{x}_i)$ is the fastest network updating direction to reduce $\mathbf{x}_i$'s loss. However, the network weights $\boldsymbol{\theta}$ are actually updated by the negative gradient of the entire batch X, $-\nabla_{\boldsymbol{\theta}}\mathcal{L}(X) = -\frac{1}{N}\sum_i \nabla_{\boldsymbol{\theta}}\mathcal{L}(\mathbf{x}_i)$. It is actually different from the best updating direction for each individual datum. Thus, the *de facto* update d_i for $\mathbf{x}_i$ refers to the actual gradient magnitude that $\mathbf{x}_i$ obtains along its best direction for loss reduction from the network update direction $-\nabla_{\boldsymbol{\theta}}\mathcal{L}(X)$, which can be computed by projecting $-\nabla_{\boldsymbol{\theta}}\mathcal{L}(X)$ onto the direction of $-\nabla_{\boldsymbol{\theta}}\mathcal{L}(\mathbf{x}_i)$: $d_i = -\nabla_{\boldsymbol{\theta}}\mathcal{L}(X) \cdot \frac{-\nabla_{\boldsymbol{\theta}}\mathcal{L}(\mathbf{x}_i)}{||-\nabla_{\boldsymbol{\theta}}\mathcal{L}(\mathbf{x}_i)||}$. In this way, d_i reflects how much effort the network will devote to reduce $\mathbf{x}_i$'s loss, and it is a direct indicator of data's priority during network training. We calculate the average *de facto* update of inliers/outliers w.r.t the weights between SSD's penultimate and softmax layer and visualize some examples in Fig. 3(b)-3(d): Although the average *de facto* update of inliers/outliers is very close at the beginning, the average *de*



(a) MNIST ("3") (b) F-MNIST ("bag") (c) CIFAR10 ("horse") (d) SVHN ("3")

Figure 4: Normalized histograms of inliers/outliers' $S_{pl}(\mathbf{x})$. The class used as inliers is in brackets.

facto update of inliers becomes evidently higher than outliers as the training continues, which implies that *SSD will devote more efforts to reducing inliers' loss by its network updating direction.*

Remarks on Inlier Priority. 1) Based on the discussion above, inliers will gain priority in terms of both the gradient magnitude and the updating direction of SSD's network weights. Such priority leads to a lower loss for inliers after training, which enables us to discern outliers by SSD's outputs and serves as a foundation of end-to-end UOD. **2)** Intuitively, inlier priority will also happen when using AE/CAE based end-to-end UOD methods. However, the effect of inlier priority is severely diminished in this case for two reasons: First, AE/CAE typically uses the raw image pixels as learning targets, but the intra-class difference of inlier images can be very large, which means AE/CAE usually does not have a unified learning target like SSD. Second, AE/CAE is ineffective in learning high-level representations (as we discussed in Sec. 3.1), which makes it difficult to capture common high-level semantics of inlier images. Both factors above disable inliers from being a joint force to dominate the training of AE and produce a strong inlier priority effect like SSD, which is also demonstrated by AE/CAE's poor UOD performance in empirical evaluation (see experimental results in Sec. 4.2).

3.3 Scoring Strategies for UOD

Based on inlier priority, we need a strategy $S(\cdot)$ to score a datum $\mathbf{x}$. Given $\mathbf{x}^{(y)} = O(\mathbf{x}|y)$ and the probability vector $P(\mathbf{x}^{(y)}|\boldsymbol{\theta})$ from SSD's softmax layer, we explore three strategies below:

Pseudo Label based Score (PL): Inlier priority suggests that SSD will prioritize reducing inliers' loss during training. For the datum $\mathbf{x}^{(y)}$, we note that the calculation of its cross entropy loss only depends on the probability $P^{(y)}(\mathbf{x}^{(y)}|\boldsymbol{\theta})$ that corresponds to its pseudo label y in $P(\mathbf{x}^{(y)}|\boldsymbol{\theta})$. Thus, we propose a direct scoring strategy $S_{pl}(\mathbf{x})$ by averaging $P^{(y)}(\mathbf{x}^{(y)}|\boldsymbol{\theta})$ for all K operations:

$$S_{pl}(\mathbf{x}) = \frac{1}{K}\sum_{y=1}^{K} P^{(y)}(\mathbf{x}^{(y)}|\boldsymbol{\theta}). \tag{4}$$

Maximum Probability based Score (MP): PL seems to be an ideal score. However, we note that operations for surrogate supervision do not always create sufficiently separable classes, e.g. image with a digit "8" is still an "8" when applying a flip operation. Hence, misclassifications will happen and the probability $P^{(y)}(\mathbf{x}^{(y)}|\boldsymbol{\theta})$ that corresponds to pseudo label y may not be the only or the best indicator to reflect how well the loss of a datum is reduced. Therefore, instead of $P^{(y)}(\mathbf{x}^{(y)}|\boldsymbol{\theta})$, we alternatively adopt the maximum probability of $P(\mathbf{x}^{(y)}|\boldsymbol{\theta})$ to calculate the score $S_{mp}(\mathbf{x})$ as follows:

$$S_{mp}(\mathbf{x}) = \frac{1}{K}\sum_{y=1}^{K} \max_{t} P^{(t)}(\mathbf{x}^{(y)}|\boldsymbol{\theta}). \tag{5}$$

Negative Entropy based Score (NE). Both strategies above rely on a single probability retrieved from $P(\mathbf{x}^{(y)}|\boldsymbol{\theta})$, while the information of the rest $(K-1)$ classes' probability is ignored. If we consider the entire probability distribution $P(\mathbf{x}^{(y)}|\boldsymbol{\theta})$, the training actually encourages SSD to output a probability distribution closer to the label's one-hot distribution. With inlier priority, we can expect SSD to output a sharper probability distribution $P(\mathbf{x}^{(y)}|\boldsymbol{\theta})$ for inliers and a more uniform $P(\mathbf{x}^{(y)}|\boldsymbol{\theta})$

for outliers. Thus, we propose to use information entropy $H(\cdot)$ [73] as a simple and effective measure to the sharpness of a distribution, which gives the negative entropy based score $S_{ne}(\mathbf{x})$:

$$S_{ne}(\mathbf{x}) = -\frac{1}{K}\sum_{y=1}^{K} H(P(\mathbf{x}^{(y)}|\boldsymbol{\theta})) = \frac{1}{K}\sum_{y=1}^{K}\sum_{t=1}^{K} P^{(t)}(\mathbf{x}^{(y)}|\boldsymbol{\theta})\log(P^{(t)}(\mathbf{x}^{(y)}|\boldsymbol{\theta})). \tag{6}$$

A comparison of PL/MP/NE is given in Sec. 4.2. In Fig. 4(a)-4(d), we calculate the most intuitive $S_{pl}(\mathbf{x})$ of inliers/outliers on benchmarks and visualize the normalized histograms of $S_{pl}(\mathbf{x})$, which are favorably separable for UOD. Besides, such results also verify the effectiveness of inlier priority.

4 Experiments

4.1 Experiment Setup

UOD Performance Evaluation on Image Benchmarks. We follow the standard procedure from previous image UOD literature [13, 16, 17] to construct an image set with outliers: Given a standard image benchmark, all images from a class with one common semantic concept (e.g. "horse", "bag") are retrieved as inliers, while outliers are randomly sampled from the rest of classes by an outlier ratio ρ. We vary ρ from 5% to 25% by a step of 5%. The assigned inlier/outlier labels are strictly unknown to UOD methods and only used for evaluation. Each class of a benchmark is used as inliers in turn and the performance on all classes is averaged as the overall UOD performance. The experiments are repeated for 5 times to report the average results. Five public benchmarks: MNIST [74], Fashion-MNIST (F-MNIST) [75], CIFAR10 [76], SVHN [77], CIFAR100 [76] are used for experiments[3]. Raw pixels are directly used as inputs with their intensity normalized into $[-1, 1]$. As for evaluation, we adopt the commonly-used Area under the Receiver Operating Characteristic curve (AUROC) and Area under the Precision-Recall curve (AUPR) as threshold-independent metrics [78].

Implementation Details and Compared Methods. For *E*^{3}*Outlier*, we use an $n = 10$ layer wide ResNet (WRN) with a widen factor $k = 4$ as the backbone DNN architecture. $K = 111$ operations are used for surrogate supervision, and NE is used as the scoring strategy. Since surrogate supervision augments original data by K times, we train WRN for $\lceil \frac{250}{K} \rceil$ epochs. The batch size is 128. A learning rate 0.001 and a weight decay 0.0005 are adopted. The SGD optimizer with momentum 0.9 is used for MNIST and F-MNIST, while the Adam optimizer with $\boldsymbol{\beta} = (0.9, 0.999)$ is used for CIFAR10, CIFAR100 and SVHN for better convergence. We compare *E*^{3}*Outlier* with the baselines and existing state-of-the-art DNN based UOD methods (reviewed in Sec. 2) below: **1)** CAE [79]. It directly uses CAE's reconstruction loss to perform UOD. **2)** CAE-IF. It feeds CAE's learned representations into isolation forest (IF) [40] as explained in Sec. 3.1. **3)** Discriminative reconstruction based autoencoder (DRAE) [16]. **4)** Robust deep autoencoder (RDAE) [17]. **5)** Deep autoencoding gaussian mixture model (DAGMM) [18]. **6)** SSD-IF. It shares *E*^{3}*Outlier*'s SSD part but feeds SSD's learned representations into IF to perform UOD. For all AE based UOD methods above, we adopt the same CAE architecture from [58] with a 4-layer encoder and 4-layer decoder. We do not use more complex CAE (e.g. CAE using skip connection [80] or more layers) since they usually lower outliers' reconstruction error as well and do not contribute to CAE's UOD performance. The hyperparameters of the compared methods are set to recommended values (if provided) or the values that produce the best performance. More implementation details are given in Sec. 1 of the supplementary material. Our codes and results can be verified at `https://github.com/demonzyj56/E3Outlier`.

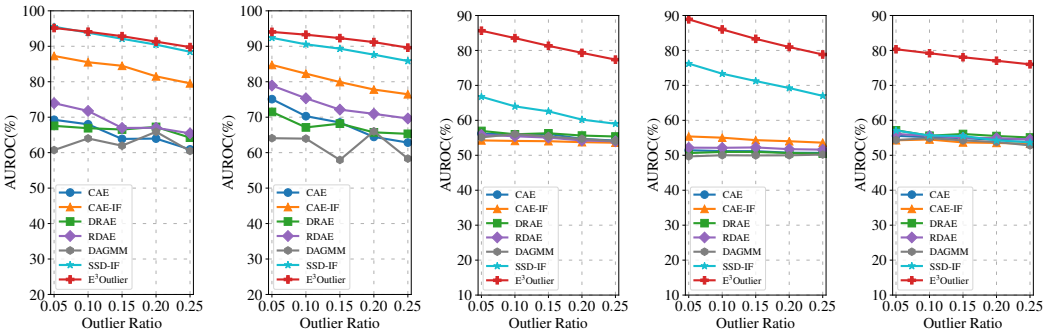
4.2 UOD Performance Comparison and Discussion

UOD Performance Comparison. We report the numerical results on each benchmark under $\rho = 10\%$ and 20% in Table 1, and UOD performance by AUROC under ρ from 5% to 25% is shown in Fig. 5(a)-Fig. 5(e) (full results are given in Sec. 4 of supplementary material). AUPR-in and AUPR-out in Table 1 denote the AUPR calculated when inliers and outliers are used as positive class respectively. We draw the following observations from those results: Above all, *E*^{3}*Outlier* overwhelmingly outperforms existing DNN based UOD methods by a large margin. As Table 1 shows, *E*^{3}*Outlier* usually improves AUROC/AUPR by 5% to 30% when compared with state-of-the-art UOD methods. In particular, *E*^{3}*Outlier* produces a significant performance leap

[3] As all images are viewed as unlabelled in UOD, we do not split train/test set. CIFAR100 uses 20 superclasses.

Table 1: AUROC/AUPR-in/AUPR-out (%) for UOD methods. The best performance is in bold.

Dataset	ρ	CAE	CAE-IF	DRAE	RDAE	DAGMM	SSD-IF	*E*^{3}*Outlier*
MNIST	10%	68.0/92.0/32.9	85.5/97.8/49.0	66.9/93.0/30.5	71.8/93.1/35.8	64.0/92.9/26.6	93.8/99.2/**68.7**	**94.1/99.3**/67.5
	20%	64.0/82.7/40.7	81.5/93.6/57.2	67.2/86.6/42.5	67.0/84.2/43.2	65.9/86.4/41.3	90.5/97.3/71.0	**91.3/97.6/72.3**
F-MNIST	10%	70.3/94.3/29.3	82.3/97.2/40.3	67.1/93.9/25.5	75.3/95.8/31.7	64.0/92.7/30.3	90.6/98.5/68.6	**93.3/99.0/75.9**
	20%	64.4/85.3/36.8	77.8/92.2/49.0	65.7/86.9/36.6	70.9/89.2/41.4	66.0/86.7/43.5	87.6/95.6/71.4	**91.2/97.1/78.9**
CIFAR10	10%	55.9/91.0/14.4	54.1/90.2/13.7	56.0/90.7/14.7	55.4/90.7/14.0	56.1/91.3/15.6	64.0/93.5/18.3	**83.5/97.5/43.4**
	20%	54.7/81.6/25.5	53.8/80.7/25.3	55.6/81.7/26.8	54.2/81.0/25.7	54.7/81.8/26.3	60.2/85.0/28.3	**79.3/93.1/52.7**
SVHN	10%	51.2/90.3/10.6	55.0/91.4/11.9	51.0/90.3/10.5	52.1/90.6/10.8	50.0/90.0/19.3	73.4/95.9/22.0	**86.0/98.0/36.7**
	20%	50.7/80.2/20.7	54.0/82.0/22.4	50.6/80.4/20.5	51.8/80.9/21.1	50.0/79.9/29.6	69.2/89.5/33.7	**81.0/93.4/47.0**
CIFAR100	10%	55.2/91.0/14.5	54.5/90.7/13.8	55.6/90.9/15.0	55.8/90.9/15.0	54.9/91.1/14.2	55.6/91.5/13.0	**79.2/96.8/33.3**
	20%	54.4/81.7/25.6	53.5/80.9/25.1	55.5/81.8/27.0	54.9/81.5/26.5	53.8/81.5/24.7	54.3/82.1/23.4	**77.0/92.4/46.5**

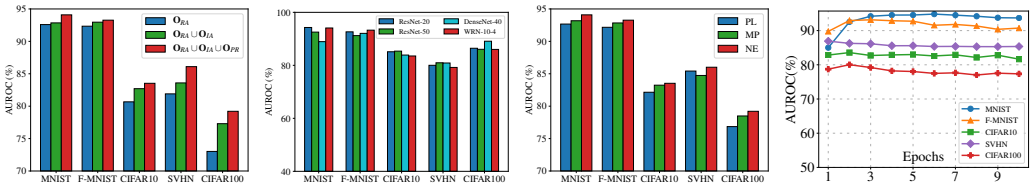


(a) MNIST (b) F-MNIST (c) CIFAR10 (d) SVHN (e) CIFAR100

Figure 5: UOD performance (AUROC) comparison with varying ρ from 5% to 25%.

($\geq$ 20% AUROC gain) on CIFAR10, SVHN and CIFAR100, which have constantly been difficult benchmarks for UOD. Next, end-to-end *E*^{3}*Outlier* almost consistently outperforms its decoupled counterpart SSD-IF. Although SSD-IF performs closely to *E*^{3}*Outlier* in simple cases, *E*^{3}*Outlier* evidently prevails over SSD-IF on CIFAR10/SVHN/CIFAR100 by 11% to 24% AUROC gain. By contrast, the decoupled CAE-IF/RDAE get better UOD performance than their end-to-end counterparts CAE/DRAE/DAGMM on MNIST/F-MNIST, and all of them yield inferior performance on CIFAR10/SVHN/CIFAR100. Hence, observations above have justified *E*^{3}*Outlier* as a highly effective and end-to-end UOD solution. In addition, we would like to make two remarks: **1)** We must point out that the data augmentation effect (surrogate supervision will augment the training data by K times) is not the reason why *E*^{3}*Outlier* outperforms existing methods by a large margin. Experiments show that when we train CAE with the same training data with *E*^{3}*Outlier*, the performance typically becomes worse than original CAE (e.g. 55.5%/63.9%/54.2%/50.0%/53.8% AUROC on MNIST/F-MNIST/CIFAR10/SVHN/CIFAR100 when $\rho = 10\%$). By contrast, *E*^{3}*Outlier* can effectively exploit the high-level discriminative label information from data of pseudo classes, which is fundamentally different from generative models like AE/CAE. **2)** To fairly compare the quality of learned representation for CAE and SSD, CAE's hidden layer by default shares SSD's penultimate layer dimension, which is fixed to 256 by Wide-ResNet architecture. A different latent dimension may influence CAE's performance, but it cannot enable CAE to perform comparably to *E*^{3}*Outlier*, especially on difficult datasets like CIFAR10. We also test other values for CAE's latent dimensions, and experimental results show that even for a carefully selected latent dimension (e.g. 64) that performs best on most benchmarks, it brings minimal gain to CAE's performance on difficult datasets CIFAR10/CIFAR100 (e.g. 56.3%/56.1% AUROC when $\rho = 10\%$), and on simpler datasets (MNIST/F-MNIST/SVHN) CAE's performance (71.9%/75.6%/53.4%, $\rho = 10\%$) is still far behind *E*^{3}*Outlier* (94.1%/93.3%/86.0%) despite some limited improvement. More importantly, a prior choice of the optimal latent dimension or CAE architecture for UOD is difficult in itself.

Discussion. We discuss five factors that are related to our *E*^{3}*Outlier* framework's performance by experiments. Since the trends under different values of ρ are fairly similar, we visualize the results when using $\rho = 10\%$: **1)** Operation set for surrogate supervision (see Fig. 6(a)): We test the UOD



(a) Operation set (b) Network architecture (c) Scoring strategy (d) Training epochs

Figure 6: Different factors' influence on *E*^{3}*Outlier*'s performance under $\rho = 10\%$.

performance with different combinations of operation subsets to be $\mathcal{O}$. The results suggest that $\mathcal{O}_{RA}$ alone already works satisfactorily, but a union of $\mathcal{O}_{RA}$, $\mathcal{O}_{IA}$ and $\mathcal{O}_{PR}$ produces the best performance, which reflects the extendibility of operation sets. **2)** Network architecture (see Fig. 6(b)): In addition to WRN, we explore ResNet-20/ResNet-50 [19] and DenseNet-40 [81] for SSD with other settings fixed. The results show that those architectures basically achieve satisfactory UOD performance with minor differences, which verifies the applicability of different network architectures. In particular, we note that a more complex architecture (ResNet-50/DenseNet-40) improves the UOD performance on relatively complex datasets (CIFAR10, SVHN and CIFAR100), but its performance is inferior on simple datasets. **3)** Scoring strategy (see Fig. 6(c)): Among three scoring strategies (PL/MP/NE) proposed in Sec. 3.3, NE constantly yields the best performance by up to 2.3% AUC gain compared with PL/MP, while MP also outperforms the naive PL. Thus, we use the NE by default for *E*^{3}*Outlier*. **4)** Training epochs (see Fig. 6(d)): We measure the UOD performance when the SSD is trained by 1 to 10 epochs respectively. In general, the UOD performance is improved at the initial stage of training (less than 3 training epochs) and then stabilizes as the training epochs continue to increase. **5)** Outlier ratio: First, we note that sometimes the ratio of outliers can be very small (e.g. $\leq 1\%$), so we also test *E*^{3}*Outlier*'s performance in such case. The experiments show that *E*^{3}*Outlier* still achieves satisfactory performance: For example, when $\rho = 0.5\%$, *E*^{3}*Outlier* achieves 96.0%/93.6%/87.4%/91.0%/80.7% AUROC for MNIST/F-MNIST/CIFAR10/SVHN/CIFAR100 respectively, which is even better than the case with a higher outlier ratio. We also notice that the performance of *E*^{3}*Outlier* tends to drop as the outlier ratio ρ increases. This is reasonable in the setting of UOD because the "outlierness" of outliers will decrease as their number increases, i.e. they are less likely to be viewed as "outliers" under the unsupervised setting as they gradually play a more important role in constituting the original unlabelled data.

5 Conclusion

In this paper, we propose a framework named *E*^{3}*Outlier* to achieve effective and end-to-end UOD from raw image data. *E*^{3}*Outlier* exploits surrogate supervision rather than traditional AE/CAE for representation learning in UOD, while a new property named inlier priority is demonstrated theoretically and empirically as the foundation of end-to-end UOD. By inlier priority and the negative entropy based score, *E*^{3}*Outlier* achieves significant UOD performance leap when compared with state-of-the-art DNN based UOD methods. For future research, it is interesting to explore a quantitative measure of each operation's effectiveness for surrogate supervision and develop effective late fusion strategies of different operations for scoring. As an open framework, different network architectures, surrogate supervision operations and scoring strategies can also be explored for *E*^{3}*Outlier*.

Acknowledgement

This work is supported by National Key R&D Program of China 2018YFB1003203 and National Natural Science Foundation of China (NSFC) under Grant No. 61773392, 61672528. This work is also supported by the German Research Foundation (DFG) award KL 2698/2-1 and by the German Federal Ministry of Education and Research (BMBF) awards 031L0023A, 01IS18051A, and 031B0770E. Xinwang Liu, En Zhu and Jianping Yin are corresponding authors of this paper.

References

[1] Douglas M Hawkins. *Identification of outliers*, volume 11. Springer.

[2] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15, 2009.

[3] Mohiuddin Ahmed, Abdun Naser Mahmood, and Md Rafiqul Islam. A survey of anomaly detection techniques in financial domain. *Future Generation Computer Systems*, 55:278–288, 2016.

[4] Anna L Buczak and Erhan Guven. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials*, 18(2):1153–1176.

[5] Alessandra De Paola, Salvatore Gaglio, Giuseppe Lo Re, Fabrizio Milazzo, and Marco Ortolani. Adaptive distributed outlier detection for wsns. *IEEE transactions on cybernetics*, 45(5):902–913, 2015.

[6] Charu C Aggarwal. *Outlier Analysis*. Springer, 2016.

[7] Varun Chandola and Vipin Kumar. Outlier detection : A survey. *Acm Computing Surveys*, 41(3), 2007.

[8] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International Conference on Information Processing in Medical Imaging*, pages 146–157. Springer, 2017.

[9] B Ravi Kiran, Dilip Mathew Thomas, and Ranjith Parakkal. An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos. *Journal of Imaging*, 4(2):36, 2018.

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

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

[12] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Neural Information Processing Systems*, pages 7167–7177, 2018.

[13] Wei Liu, Gang Hua, and John R Smith. Unsupervised one-class learning for automatic outlier removal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3826–3833, 2014.

[14] Siqi Wang, Yijie Zeng, Qiang Liu, Chengzhang Zhu, En Zhu, and Jianping Yin. Detecting abnormality without knowing normality: A two-stage approach for unsupervised video abnormal event detection. In *2018 ACM Multimedia Conference on Multimedia Conference*, pages 636–644. ACM, 2018.

[15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[16] Yan Xia, Xudong Cao, Fang Wen, Gang Hua, and Jian Sun. Learning discriminative reconstructions for unsupervised outlier removal. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1511–1519, 2015.

[17] Chong Zhou and Randy C Paffenroth. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 665–674. ACM, 2017.

[18] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International Conference on Learning Representations (ICLR)*, 2018.

[19] 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*, pages 770–778, 2016.

[20] Raghavendra Chalapathy and Sanjay Chawla. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*, 2019.

[21] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[22] Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (8):832–844, 1998.

[23] Yue Zhao and Maciej K Hryniewicki. Xgbod: improving supervised outlier detection with unsupervised representation learning. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2018.

[24] David Martinus Johannes Tax. One-class classification: Concept learning in the absence of counter-examples. 2002.

[25] Markos Markou and Sameer Singh. Novelty detection: a review—part 1: statistical approaches. *Signal processing*, 83(12):2481–2497, 2003.

[26] Bernhard Scholkopf, Ralf Herbrich, and Alexander J Smola. A generalized representer theorem. *european conference on computational learning theory*, pages 416–426, 2001.

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

[28] Graham Williams, Rohan Baxter, Hongxing He, Simon Hawkins, and Lifang Gu. A comparative study of rnn for outlier detection in data mining. In *2002 IEEE International Conference on Data Mining, 2002. Proceedings.*, pages 709–712. IEEE, 2002.

[29] Nathalie Japkowicz, Catherine Myers, and Mark Gluck. A novelty detection approach to classification. In *International Joint Conference on Artificial Intelligence*, pages 518–523, 1995.

[30] Mei-ling Shyu, Shu-ching Chen, Kanoksri Sarinnapakorn, and Liwu Chang. A novel anomaly detection scheme based on principal component classifier. In *in Proceedings of the IEEE Foundations and New Directions of Data Mining Workshop, in conjunction with the Third IEEE International Conference on Data Mining (ICDM'03*. IEEE, 2003.

[31] Heiko Hoffmann. Kernel pca for novelty detection. *Pattern recognition*, 40(3):863–874, 2007.

[32] Frank E Grubbs. Procedures for detecting outlying observations in samples. *Technometrics*, 11(1):1–21, 1969.

[33] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[34] Zengyou He, Xiaofei Xu, and Shengchun Deng. Discovering cluster-based local outliers. *Pattern Recognition Letters*, 24(9-10):1641–1650, 2003.

[35] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *ACM sigmod record*, volume 29, pages 93–104. ACM, 2000.

[36] Emanuel Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962.

[37] JooSeuk Kim and Clayton D Scott. Robust kernel density estimation. *Journal of Machine Learning Research*, 13(Sep):2529–2565, 2012.

[38] Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. In *ACM Sigmod Record*, volume 29, pages 427–438. ACM, 2000.

[39] Fabrizio Angiulli and Clara Pizzuti. Fast outlier detection in high dimensional spaces. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 15–27. Springer, 2002.

[40] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422. IEEE, 2008.

[41] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. On detecting clustered anomalies using sciforest. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 274–290. Springer, 2010.

[42] Sunil Aryal, Kai Ming Ting, Jonathan R Wells, and Takashi Washio. Improving iforest with relative mass. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 510–521. Springer, 2014.

[43] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5375–5384, 2016.

[44] Salman H Khan, Munawar Hayat, Mohammed Bennamoun, Ferdous A Sohel, and Roberto Togneri. Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE transactions on neural networks and learning systems*, 29(8):3573–3587, 2018.

[45] Qi Dong, Shaogang Gong, and Xiatian Zhu. Imbalanced deep learning by minority class incremental rectification. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[46] Chen Huang, Chen Change Loy, and Xiaoou Tang. Discriminative sparse neighbor approximation for imbalanced learning. *IEEE transactions on neural networks and learning systems*, 29(5):1503–1513, 2018.

[47] Shuangfei Zhai, Yu Cheng, Weining Lu, and Zhongfei Zhang. Deep structured energy based models for anomaly detection. *international conference on machine learning*, pages 1100–1109, 2016.

[48] Dan Xu, Yan Yan, Elisa Ricci, and Nicu Sebe. Detecting anomalous events in videos by learning deep representations of appearance and motion. *Computer Vision and Image Understanding*, 156:117–127, 2017.

[49] Mahmudul Hasan, Jonghyun Choi, Jan Neumann, Amit K Roy-Chowdhury, and Larry S Davis. Learning temporal regularity in video sequences. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 733–742, 2016.

[50] Yiru Zhao, Bing Deng, Chen Shen, Yao Liu, Hongtao Lu, and Xian-Sheng Hua. Spatio-temporal autoencoder for video anomaly detection. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1933–1941. ACM, 2017.

[51] Lucas Deecke, Robert A Vandermeulen, Lukas Ruff, Stephan Mandt, and Marius Kloft. Anomaly detection with generative adversarial networks. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 3–17, 2018.

[52] Chu Wang, Yan-Ming Zhang, and Cheng-Lin Liu. Anomaly detection via minimum likelihood generative adversarial networks. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 1121–1126. IEEE, 2018.

[53] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Georg Langs, and Ursula Schmidt-Erfurth. f-anogan: Fast unsupervised anomaly detection with generative adversarial networks. *Medical image analysis*, 54:30–44, 2019.

[54] Lukas Ruff, Nico Görnitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Robert Vandermeulen, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *International Conference on Machine Learning*, pages 4390–4399, 2018.

[55] Raghavendra Chalapathy, Aditya Krishna Menon, and Sanjay Chawla. Anomaly detection using one-class neural networks. *arXiv preprint arXiv:1802.06360*, 2018.

[56] Pramuditha Perera and Vishal M Patel. Learning deep features for one-class classification. *arXiv preprint arXiv:1801.05365*, 2018.

[57] Patrick Schlachter, Yiwen Liao, and Bin Yang. Deep one-class classification using data splitting. *arXiv preprint arXiv:1902.01194*, 2019.

[58] Izhak Golan and Ran El-Yaniv. Deep anomaly detection using geometric transformations. In *Advances in Neural Information Processing Systems*, pages 9758–9769, 2018.

[59] Yezheng Liu, Zhe Li, Chong Zhou, Yuanchun Jiang, Jianshan Sun, Meng Wang, and Xiangnan He. Generative adversarial active learning for unsupervised outlier detection. *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[60] Nikos Komodakis and Spyros Gidaris. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations (ICLR)*, 2018.

[61] Rodrigo Santa Cruz, Basura Fernando, Anoop Cherian, and Stephen Gould. Visual permutation learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[62] Xu Ji, Joao F Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. *arXiv: Computer Vision and Pattern Recognition*, 2018.

[63] Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *arXiv: Computer Vision and Pattern Recognition*, 2019.

[64] Xi Peng, Jiashi Feng, Jiwen Lu, Wei-Yun Yau, and Zhang Yi. Cascade subspace clustering. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[65] Jianlong Chang, Lingfeng Wang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. Deep adaptive image clustering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5879–5887, 2017.

[66] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In *International Conference on Machine Learning*, pages 1558–1566, 2016.

[67] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. In *Advances in neural information processing systems*, pages 658–666, 2016.

[68] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference 2016*. British Machine Vision Association, 2016.

[69] Andrew F Emmott, Shubhomoy Das, Thomas Dietterich, Alan Fern, and Weng-Keen Wong. Systematic construction of anomaly detection benchmarks from real data. In *Proceedings of the ACM SIGKDD workshop on outlier detection and description*, pages 16–21. ACM, 2013.

[70] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge & Data Engineering*, (9):1263–1284, 2008.

[71] Justin M. Johnson and Taghi M. Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):27, 2019.

[72] Léon Bottou. Online learning and stochastic approximations. *On-line learning in neural networks*, 17(9):142, 1998.

[73] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[74] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[75] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[76] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

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

[78] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240, 2006.

[79] Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In *International Conference on Artificial Neural Networks*, pages 52–59. Springer, 2011.

[80] Xiaojiao Mao, Chunhua Shen, and Yu-Bin Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In *Advances in neural information processing systems*, pages 2802–2810, 2016.

[81] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.