UTS UNIVERSITY OF TECHNOLOGY SYDNEY

Statistical Methods for Out-of-distribution Detection

by Zhilin Zhao

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

under the supervision of Longbing Cao

University of Technology Sydney
Faculty of Engineering and Information Technology

April, 2023

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, Zhilin Zhao, declare that this thesis is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the Faculty of Engineering and Information Technology at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Production Note:
Signature removed prior to publication.

April, 2023

ABSTRACT

For a network trained on in-distribution (ID) samples, test samples could be out-of-distribution (OOD) that are drawn from distributions different from that of ID samples. Accordingly, OOD detection aims to identify OOD samples in test phases. The main challenge lies in that a network could provide high-confidence predictions for OOD samples, which indicates that the network cannot distinguish ID and OOD samples. The main causes of the high-confidence issue include limited ID and unavailable OOD samples in training processes. One strategy to enhance the detection performance of a network is to make the outputs more sensitive to OOD samples, i.e., the network tends to provide high- and low-confidence predictions for ID and OOD samples, respectively.

Improving the OOD sensitivity for a network requires to address a series of important problems and challenges: (1) Penalizing OOD samples with high-confidence predictions can improve the OOD sensitivity. Accordingly, how to generate specific OOD samples for a network? (2) If partial OOD samples are observed, how to involve them in the retraining process to balance the ID generalization and OOD detection? (3) If OOD samples are unavailable, how to fine-tune a network with augmented ID samples to improve the OOD sensitivity? (4) If modifying the network is not allowed, how to learn an auxiliary network to capture the OOD-sensitive information for the network?

This thesis systematically studies how to effectively solve the aforementioned issues with experimental and theoretical support. Due to the significant difference between ID and OOD samples, it is essential to consider the data characteristics and data correlations that statistical methods can model. Accordingly, this thesis attempts to incorporate statistical methods into deep neural networks to improve the OOD sensitivity. Specifically, this thesis proposes four novel methods to address these issues. The main ideas include inferring an implicit generator based on the Shannon entropy to generate high-confidence OOD samples, constructing adaptive supervision information for OOD samples to minimize the disruption for learning to classify ID samples, exploring the data space around ID samples to construct the vicinity distributions for OOD samples, and utilizing an auxiliary network to explore the discarded OOD-sensitive information in ID samples according to information bottleneck theory.

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to my supervisor Prof. Longbing Cao for his patient and valuable guidance. He taught me how to find interesting research questions, how to provide valuable insights, how to develop novel algorithms, how to write technical papers and how to become an independent researcher all from scratch. He gave me great freedom to explore advanced research directions, showed extreme patience in revising papers, and shared everything he knew with me. Without his instructions, inspiration, encouragement, and guidance, I would not finish research works and submit to leading journals. Therefore, I feel lucky to be led by Prof. Longbing Cao to do research in the doctoral stage.

I am also greatly indebted my co-supervisor Prof. Philip.S Yu, for his precious encouragement. I would like to thank A/Prof. Chang-dong Wang for helping me take the first step toward the academic circle. I would like to thank the panel members of my candidature assessment, A/Prof. Wei Liu, and Dr. Marian-Andrei RIZOIU, for their constructive comments.

Furthermore, I thank all my colleagues in the Data Science Lab for their help in my daily life and for their critical suggestions for my research works: Guansong Pang, Chengzhang Zhu, Qi Zhang, Wei Wang, Qing Liu, Shoujin Wang, Liang Hu, Trong Dinh Thac Do, Tianfu Zhang, Yawen Zheng, Qinfen Wang, Siyuan Ren, Oliver Qi, Hu Cao, James Yang, and Zhangkai Wu. I would also like to thank my friends from other institutions. I thank Yuanyu Wan at Nanjing University for learning machine learning theory with me. I thank Kunyu Lin at Sun Yat-sen University for writing code and bugs with me.

Last but not least, I wholeheartedly thank my father and mother for their unconditional support throughout my Ph.D. study. Without their understanding and support, I cannot make any achievements and also cannot see the vastness of the world.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

In this chapter, we briefly introduce the background of out-of-distribution detection, related challenges and questions, thesis contributions, and finally show the framework of the entire thesis.

1.1 Background

Deep neural networks (DNNs) have demonstrated a significant generalization ability when the independent and identically distributed (i.i.d.) assumption is satisfied [1, 2, 3]. This assumption indicates that training and test samples are drawn from the same distribution, i.e., *in-distribution* (ID). However, in real-world scenarios, test samples may come from different distributions differing from that of ID samples, i.e., *out-of-distribution* (OOD). An undesirable situation is that networks tend to make high-confidence prediction [4] on test samples with semantic shift [5], i.e., OOD samples, as shown in Fig. 1.1. This over-confidence phenomenon on OOD samples makes DNNs fail to know whether test samples are OOD, which can limit its adaption and cause serious OOD issues [6, 7]. Therefore, improving the ability to distinguish ID and OOD samples, i.e., the OOD sensitivity of a network, is a significant concern for DNN robustness and AI Safety [8]. To evaluate the OOD sensitivity of networks, the considered task *OOD detection* aims to identify the test samples that are semantically different from the training ID samples and should be rejected to belong to the known classes.

The cause of the over-confidence phenomenon lies in the significant difference between ID and OOD samples [10]. Specifically, in the training phase, only limited ID samples are observed, and OOD samples are unavailable. Accordingly, DNNs only learn to assign high-confidence predictions to the observed ID samples, which indicates that the predictions for OOD samples are uncertain. It indicates that some OOD samples could receive unexpected high-confidence predictions in the test phase, which causes that networks cannot distinguish between ID and OOD samples. Due to the significant difference between ID and OOD samples, it is essential to consider the data characteristics and data correlations to improve the OOD sensitivity of DNNs. One interesting and challenging direction is to incorporate *statistical methods* into the pow-

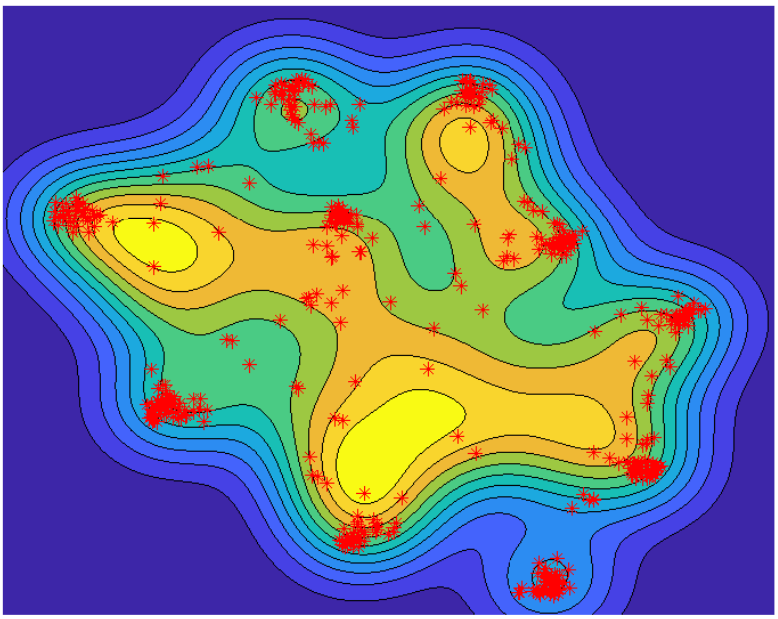


Fig. 1.1 Heat map of prediction confidence.
The embedding results are constructed by t-SNE [9]. Red points correspond to training ID samples. Yellow regions correspond to high confidence for predictions, while blue regions correspond to low confidence. The trained network assigns high-confidence predictions on samples located in the regions outside the training ID samples, i.e., OOD samples. It shows the network does not discriminate between ID and OOD samples. The figure is best viewed in color.

erful DNNs. The main insight is that statistical methods, extracting information from data, to consider the complex data characteristics and data correlations that are sensitive to OOD samples in the training process of DNNs.

1.2 Research Questions

For a pretrained network learned from a training ID dataset, to improve its OOD sensitivity, we retrain or fine-tune the pretrained network by incorporating statistical methods. When statistical methods meet OOD detection, we consider a series of research questions in this thesis. Penalizing OOD samples with confidence predictions can improve OOD sensitivity because OOD samples are expected to have low confidence predictions. However, OOD samples are usually unavailable in the training phases. Accordingly, the first research question is how to generate specific OOD samples with high-confidence predictions for a given network. Suppose OOD samples can be obtained from real-world datasets or generated by other generative models in training phases. In that case, the second research question that should be considered is how to balance

ID classification and OOD detection. This is because ID classification performance is the major concern of the pretrained network. If the balance issue can be addressed, we can improve OOD sensitivity while maintaining ID classification capacity by using generated OOD samples when real-world OOD samples are unavailable. However, generating OOD samples by training generative models is expensive. Accordingly, the third research question is how to improve the OOD sensitivity by exploring augmented ID samples rather than OOD samples. If improving OOD sensitivity can be achieved by exploring the information from ID samples and applying the information to retrain or finetune the network, the fourth research question is how to construct an auxiliary network for a given pretrained network to explore the OOD-sensitive information without modifying this network. The detailed research questions and the corresponding outcomes are presented below.

- *How to generate network-specific OOD samples for fine-tuning a pretrained network?* Limited training ID samples without OOD samples cause *distributional vulnerability*, i.e., the trained network makes uncertain predictions of OOD samples. One possible speculation is that the distributional vulnerability is ID sample-based network-specific. This is because altering training data results in different network parameters [11] and different networks generate various distributions of data representations [12]. To reveal and patch the distributional vulnerability of a network, one idea is to fine-tune the network with OOD samples drawn from a specific OOD generator, which makes it sensitive to the distributional vulnerability. However, the relevant methods [13, 14] without considering data and network characteristics cannot generate specific OOD samples with semantic shift and high-confidence predictions. They define OOD samples according to prior knowledge without targeting the distributional vulnerability, misaddressing the distributional vulnerability of the given network. Such methods cannot explore most high-confidence OOD samples specific to the network.

- *How to balance the ID classification and OOD detection when OOD samples are involved in the retraining process?* One straightforward idea to address the over-confidence issue is to introduce extra samples from other datasets as OOD to restrict the network behavior. Since the introduced OOD samples are label-free, their supervision information is usually determined manually based on prior knowledge, which is then applied in the training process. However, this approach sacrifices the classification accuracy because the artificial supervision information (i.e., manual labeling per prior knowledge) incorporated on OOD samples is independent of ID samples. Specifically, the OOD supervision information exclusive to ID samples could interfere with the ID classification process. For example, the artificial supervision information could be defined as an extra class [15] for

all OOD samples drawn from different distributions. Forcing all OOD samples to share the same class leads to an extra hyperplane that regards ID samples located between different OOD samples in the data space as OOD, which further decreases the classification accuracy. In addition, the artificial supervision information could be a class probability vector drawn from a flat distribution [16]. The class probability vector is associated with an OOD sample by an extra regularizer without changing the number of classes. This method could improperly change the class distribution due to the artificially-incorporated class probability vectors. Also, the regularizer strictly independent of the ID samples affects the classification capacity since it makes networks less attentive to learning from ID samples.

- *How to improve OOD sensitivity by retraining with augmented ID samples?* According to the learning rule of empirical risk minimization [17, 18], a pretrained network is learned by minimizing the average error over the independent and identically distributed ID samples. To address the over-confidence issue in the pretrained network, the empirical risk minimization principle should be extended to consider OOD samples in the training process, i.e., going beyond the conventional IID assumption for training and test samplings. Specifically, one straightforward idea is to fine-tune the pretrained network with generated OOD samples if no extra real-world OOD samples are available [16]. OOD samples are relative to an ID dataset, e.g., an OOD sample for an ID dataset could be ID for other ID datasets, and an ID sample could be OOD for other ID datasets. Therefore, it would be appropriate to tailor OOD samples for a given ID dataset per its data characteristics. Accordingly, we explore the related but different samples around training ID samples in the data space, i.e., finding the augmented ID samples [19] that can be treated as OOD samples.

- *How to design an OOD-sensitive auxiliary network for a pretrained network with retraining and fine-tuning?* We argue some fundamental causes of the over-confidence issue in a pretrained network include: (1) the learned *label-discriminative representations* from the pretrained network focus on capturing the ID input-label mapping, while (2) the network overlooks or weakens the learning of *distribution-discriminative representations* that can distinguish ID and OOD. This argument can be explained by the information bottleneck principle [20] of learning a trade-off between input compression and its label prediction. As a result, pretrained networks only or mainly learn label-discriminative representations from the training inputs strongly related to the labeling but discard complementary distribution-discriminative representations that may be weakly related to the labeling but strongly coupled with the label-discriminative representations. Both label- and distribution-discriminative representations contain labeling information for an ID

sample. However, an OOD sample with even weak labeling-sensitive information may hold distribution-discriminative representations corresponding to other labels or even none of any labels and still receive a high-confidence prediction from a pretrained network. Therefore, the label-discriminative representations are sufficient for the classification task but insufficient for OOD detection, and the labeling consistency between label- and distribution-discriminative representations could distinguish ID and OOD samples. Differing from the retraining and fine-tuning methods that modify network parameters, we explore an auxiliary network which captures the distribution-discriminative representations for a pretrained network to improve OOD sensitivity.

1.3 Thesis Contributions

This thesis systematically studies the above four research questions about improving OOD sensitivity and makes the following contributions.

- *Fine-tuning Discriminators by Implicit Generators (FIG) (resubmitted to TPAMI)*

    1. An implicit generator is proportional to the negative entropy of the output probabilities from a pretrained network without extra training costs.
    2. A sampler based on the Langevin dynamics efficiently draws high-confidence OOD samples from the implicit generator.
    3. A regularizer based on the design principle of the implicit generator encourages high entropy of the generated OOD samples.

- *Supervision Adaptation (SA) (submitted to TPAMI)*

    1. To improve the network OOD sensitivity and minimize the OOD interference in classifying ID samples, we reveal a form of adaptive supervision information for OOD samples by measuring the relationship between ID samples and their labels in the mixed data space containing ID and OOD samples in the lower bound of the mutual information.
    2. To further improve the classification accuracy, we estimate the adaptive supervision information by measuring the data relations between OOD and ID samples with the same class by resolving several binary regression problems.
    3. To balance the network generalization ability on ID samples and the detection capacity on OOD samples, we combine the lower bound on the mutual information in the mixed data space and the estimated supervision information of OOD samples and then simplify the combined result.

- *Learning from Cross-class Vicinity Distribution (LCVD) (submitted to TNNLS)*

    1. According to the mutual information maximization, we derive a generic expected risk for optimizing networks on ID and OOD samples.
    2. Given an ID dataset, we construct the cross-class vicinity distribution to generate its corresponding augmented ID samples that can be treated as OOD samples.
    3. We improve the discriminability of a pretrained network by fine-tuning it with the generated OOD samples according to the generic empirical risk of the generic expected risk.

- *Dual Representation Learning (DRL) (submitted to TNNLS)*

    1. Taking the information bottleneck principle, we reveal that learning a label-discriminative representation by a pretrained network alone may not sufficiently capture all labeling information of an ID sample. There may exist a complementary distribution-discriminative representation capturing the remaining labeling information.
    2. We infer the information bottleneck principle to learn the complementary distribution-discriminative representations. Accordingly, we train an auxiliary network owning the same backbone as the pretrained network to integrate multiple intermediate representations different from a label-discriminative representation into the corresponding complementary distribution-discriminative representation.
    3. By exploring the different informativeness properties of ID and OOD samples, the label and distribution-discriminative representations are combined to form OOD scores for distinguishing ID and OOD samples.

1.4 Thesis Outline

This thesis systematically studies the underlying problems behind OOD detection and provides the solutions with provable guarantee. This thesis is organized as follows:

- Chapter 2 reviews the related work.
- Chapter 3 introduces FIG generating network-specific OOD samples.
- Chapter 4 introduces SA balancing the ID classification and OOD detection when OOD samples are involved in the retraining process.

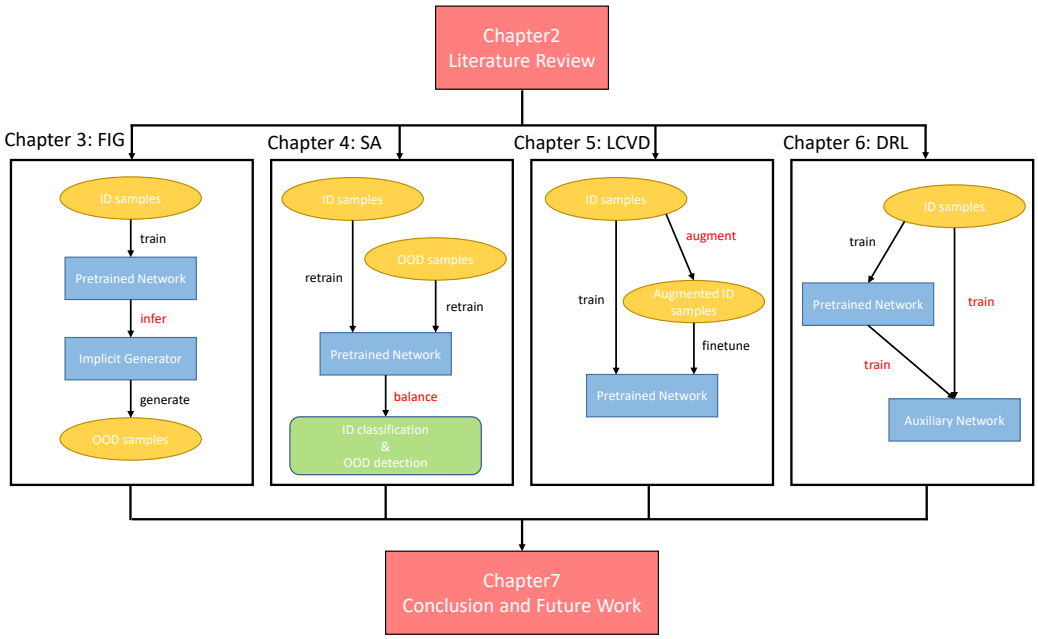


Fig. 1.2 Thesis Structure.

- Chapter 5 introduces LCVD exploring the augmented ID samples to improve OOD sensitivity.
- Chapter 6 introduces DRL training an auxiliary network to learn distribution-discriminative representations that can distinguish ID and OOD samples.
- Chapter 7 concludes this thesis and presents the challenging future work.

The organization of this thesis is presented in Fig.1.2.

CHAPTER 2

Literature Review

OOD detection [6] aims to detect whether a test sample for a network is drawn from an ID or OOD. The OOD detection performance is applied to evaluate the OOD sensitivity of networks. This problem is related to outlier detection [21], which detects whether some training samples (rather than test samples) deviate from the majority. Another related setting is the open-set recognition [22], which trains a network to assign test samples not belonging to any classes in the training set to an extra unknown class, while OOD detection does not involve extra unknown classes. In this chapter, we first introduce different kinds of OOD detection methods, including post-hoc detection methods, confidence enhancement methods, and out-of-distribution exposure methods. Then, we introduce the statistical methods for network analysis. Finally, we introduce the evaluation metrics.

2.1 Post-hoc Detection Methods

Post-hoc detection methods aim to design an OOD detector to distinguish ID and OOD samples according to the outputs from a trained network/discriminator without modifying the training procedure and objective. This property is essential for applying OOD detection methods in real-world environments because retraining or fune-tining a pre-trained network could be expensive. The baseline method [6] designs a threshold-based detector to distinguish the two kinds of samples according to maximum probabilities represented by softmax outputs [23], and the basic assumption is that a trained network tends to provide high confidence prediction for ID samples or vice versa. However, this assumption does not hold in general cases, and OOD samples could have high softmax scores due to the unavailable OOD samples in the training process. Specifically, OOD samples are unavailable and not be considered in the training process. Therefore, the predictions for OOD samples are uncertain, which indicates that some of them could have high confidence. To improve this baseline, an Out-of-DIstribution detector for Neural networks (ODIN) [24] adds negative adversarial perturbations to inputs to make ID and OOD samples more distinguishable and applies temperature scaling to the softmax function to make trained networks more sensitive to OOD samples. An

OOD sample could be assigned with high confidence prediction because it is mapped to the feature representations of ID samples, causing feature collapse [25]. Therefore, to improve the above softmax-based detectors, another set of detectors model the output distributions of various network layers. For example, MahaLanoBis (MLB) [26] combines the Mahalanobis distance calculation with input preprocessing to measure the OOD score according to the feature representations from different network layers. Based on ODIN and MLB, Deep Residual Flow (DRF) [27] leverages an expressive density model by normalizing flows to calculate the residual flows of each layer and each class for a test sample. Gram Matrix (GM) [28] calculates the OOD score by identifying feature correlations between activity patterns from all layers and the predicted class. An energy-based detector [29] applies the negative energy function in terms of the denominator of the softmax activation for OOD detection, and the log of the confidence in the Baseline method is a particular case of the negative energy function. Although the post-hoc detection methods can be efficiently applied to pretrained networks, the OOD sensitivity of pretrained networks cannot be improved, which causes the OOD detection performance heavily depends on learned knowledge in the pretrained networks. This thesis explores any other line of research to improve OOD detection performance, i.e., improving network OOD sensitivity.

2.2 Confidence Enhancement Methods

Confidence enhancement methods aim to improve the discriminability of a pretrained network by modifying the training procedure or objective. Based on ODIN, DeConf-C (DCC) [30] trains an OOD scoring function according to the divisor structure of class probability confidence and searches for the adversarial perturbation magnitude with only ID samples. Bevandic et al. [31] propose a two-head model to predict a uniform distribution of OOD samples. Blum et al. [32] propose a novel approach to separating ID and OOD samples by training a logistic regressor to aggregate the negative log-likelihoods of embeddings from all layers. Deep Gambler (DG) [33] sets a threshold for networks in the training phase, and the network abstains from making a prediction when the confidence is lower than the predetermined threshold. Based on the experimental observation that the OOD detection performance declines as the number of ID classes increases, MOS [34] groups training ID samples according to label concepts. However, the taxonomy of the label space is usually unavailable, and applying K-Means clustering on feature representations and random grouping to divide the ID dataset cannot ensure that the samples in a group have similar concepts. The confidence enhancement methods improve OOD sensitivity by involving OOD prior knowledge in training processes and objective functions. However, the two strategies of modifying networks can significantly affect the main task performance of networks. Therefore, this thesis ex-

plores learning an auxiliary network to capture the OOD information from training ID samples for a pretrained network without modification.

2.3 Out-of-distribution Exposure Methods

Out-of-distribution exposure methods aim to involve OOD samples in the retraining or fine-tuning process to improve the OOD sensitivity of a pretrained network. The OOD samples can be collected from other real-world datasets or generated by generative models. By applying the real-world samples drawn from the distributions that are different from the ID as OOD samples, outlier exposure [35] randomly selects an OOD sample for each ID sample and enlarges the gap between the log probabilities of the pair of ID-OOD samples by a margin ranking loss. The prior network [16] penalizes OOD samples by mapping their predicted distribution to a dense Dirichlet distribution in the Kullback-Leibler divergence. Considering the data characteristics of OOD samples, MIXUP [13] trains a network with samples obtained by linearly combining two randomly selected ID samples where the weights are drawn from a beta distribution. When the weights are approximately equal to a half, the generated samples can be considered OOD because the target vector combining two one-hot vectors with two almost equal weights has low confidence. In addition, considering the network characteristics, the adversarial samples [36] generated by back-propagating the gradient of the cross-entropy w.r.t. the input to a trained network are applied to retain the network, whose basic idea is to extend an input by pushing it to the decision boundary. Instead of manipulating data samples, the joint confidence loss (JCL) [37] extends the above idea to the distribution perspective with a model-specific GAN-based generator producing samples on the low-density boundary of ID samples and encouraging the target vectors of the generated samples to satisfy a uniform distribution. Instead of generating OOD samples, Self-Supervised Learning (SSL) [38] augments an ID sample by rotating it $0°, 90°, 180°, 270°$, respectively, and learns the rotation angles and the labels of augmented ID samples simultaneously. Applying the same augmentation method, the Contrasting Shifted Instances (CSI) [39] treats the original and augmented ID samples as positive and negative samples in a contrastive loss, respectively. The OOD exposure methods improve OOD detection by sacrificing ID classification performance. Accordingly, this thesis explores an interesting and challenging issue: can we balance ID generalization and OOD detection when partial OOD samples can be involved in training processes. Although several efforts consider generating OOD samples when real-world OOD samples are unavailable, the generated OOD samples are obtained according to prior knowledge of OOD samples, which indicates that the generated OOD samples cannot reveal the main cause of the high-confidence issue for a pretrained network. Therefore, this thesis considers developing specific OOD samples for a given

pretrained network or a training ID dataset.

2.4 Statistical Methods for Network Analysis

We introduce some statistical methods for network analysis that are applied in the research work in this thesis. In information theory, the Shannon entropy [40] of a random variable measures the average level of uncertainty. Extending to more than one variable, Mutual information (MI) [41] is a quantity of measuring the relationship between random variables. Based on mutual information, the information bottleneck principle [20] finds a tradeoff between the compression of input and the prediction of its label for network learning. Accordingly, the process of extracting label-related information from inputs to learn the corresponding representations can be interpreted from the information-theoretic view [42]. Differing from the previous methods focusing on measuring variable correlations, vicinity distribution enlarges the support of the training distribution to explore the samples around the training samples in the data space. The exploration of statistical methods for improving OOD sensitivity is rare. The research works in this thesis can enrich the literature community on OOD detection.

2.5 Evaluation Metrics

Area under the receiver operating characteristic curve (AUROC) [6, 24, 43] is used to evaluate the detection performance of OOD samples. AUROC is a threshold-independent metric that can be understood as the probability that an ID sample has higher prediction confidence than an OOD sample. Precisely, AUROC is calculated as the area under the ROC curve, which presents the trade-off between true positive rate (TPR) and false positive rate (FPR) across different decision thresholds. The larger AUROC is, the better the OOD detection performance is. Thus, the random method owing a $1/2$ AUROC score is the worst. Apart from AUROC, the true negative rate at 95% (FPR95) [24] and Detection are also applied to evaluate the OOD detection performance. FPR95 indicates the probability that an OOD sample is declared to be an ID sample when the true positive rate is 95%. Detection indicates the misclassification probability when the true positive rate is 95%.

Accuracy (ACC) is used to measure the classification of ID samples. Harmonic mean of AUROC and ACC is used to verify the comprehensive performance about classifying ID samples and detecting OOD samples. ECE [44] uses the difference in expectation between confidence and accuracy to measure calibration. A network is considered calibrated if its predictive confidence aligns with its misclassification rate.

CHAPTER 3

Revealing Distributional Vulnerability of Discriminators by Implicit Generators

3.1 Motivations

For a pretrained discriminator which contains a pretrained network and a linear classifier, *Fine-tuning discriminators by implicit generators* (FIG) aims to improve its OOD sensitivity by fine-tuning with discriminator-specific OOD samples. Limited training ID samples and unavailable OOD samples cause the distributional vulnerability, this vulnerability leads to high-confidence predictions for OOD samples. One possible speculation is that the distributional vulnerability is specific for a discriminator. In light of the above speculation, the following three research questions must be answered to explore distributional vulnerability and improve the OOD sensitivity of a given discriminator: (1) *How to design a specific OOD generator for a discriminator?* Training an extra generator for OOD samples is usually expensive [45, 46, 47]. Also, the generator must be related to the given discriminator to generate specific OOD samples, making it harder to design the generator. (2) *How to efficiently sample high-confidence OOD data from generators?* OOD samples with low confidence and which misaddress distributional vulnerability could mislead the fine-tuning process, and the inefficiency of generating samples results in a significant bottleneck in the fine-tuning. (3) *How to apply the generated OOD samples to patch the vulnerability?* The discriminator should be regulated as OOD sensitive to prevent the corresponding generator from generating high-confidence OOD samples. This requires the regulating method to be contrastive to the design principle of the corresponding generator.

We propose an approach of FIG to address the above challenges. For a pretrained discriminator learned from an ID dataset, we create its *implicit generator* with the same parameters as the discriminator without extra training, which is used to generate OOD samples[1]. The underlying insight is that FIG learns an OOD-sensitive discriminator by making it difficult to draw OOD samples from the corresponding implicit generator.

[1]The same ID dataset is used in both learning the pretrained discriminator and fine-tuning it with the OOD samples generated by its corresponding implicit generator.

Specifically, the implicit generator is proportional to the negative entropy of the output probabilities from the pretrained discriminator. The principle behind this construction method is that an OOD sample with high confidence prediction has a low entropy of class probabilities according to the Shannon entropy [48]. The constructed implicit generator is energy-based [49], and the samplers based on the Langevin dynamics [50] can be applied to draw samples from energy-based models effectively. After generating OOD samples, according to the construction principle of the implicit generator, we penalize them by flattening the class probabilities to make the discriminator sensitive to OOD samples.

3.2 Fine-tuning Discriminators by Implicit Generators (FIG)

We assume that ID samples $(\mathbf{x}_I, y_I)$ are i.i.d. drawn from an unknown distribution $p(\mathbf{x}, y)$, where $\mathbf{x} \in \mathbb{R}^D$ is a D-dimensional input and $y \in \mathbb{R}$ is a label, $\mathcal{D}_I$ is the ID training dataset containing N ID samples. As a typical machine learning setting, a C-class classification problem uses a parametric neural network $f_\theta : \mathbb{R}^D \to \mathbb{R}^C$ to map each input $\mathbf{x}$ to a C-dimensional output vector $(f_\theta(\mathbf{x}, 1), \ldots, f_\theta(\mathbf{x}, C))$, and a softmax output is applied to parameterize a categorical distribution for each output vector. Specifically, for class y, we estimate the probability $p(y|\mathbf{x})$ by:

$$q_\theta(y|\mathbf{x}) = \frac{\exp f_\theta(\mathbf{x}, y)}{\sum_{y' \in [C]} \exp f_\theta(\mathbf{x}, y')}, \tag{3.1}$$

and $q_\theta(y|\mathbf{x})$ is a pretrained discriminator learned from the ID training dataset X_I with the parameter θ. In general, classification tasks learn parameter θ by maximizing the objective function $\mathbb{E}_{p(\mathbf{x},y)} \log q_\theta(y|\mathbf{x})$. However, in practice, only limited ID samples following $p(\mathbf{x}, y)$ are used to estimate the probability $q_\theta(y|\mathbf{x})$, which causes the vulnerability of the pretrained discriminator $q_\theta(y|\mathbf{x})$. Specifically, OOD samples are unavailable in the training phase. It causes that the predictions for unknown OOD samples are uncertain, and some OOD samples could have high-confidence predictions. Therefore, a critical step is to reveal where the vulnerability is before patching it.

3.2.1 Implicit Generator

A pretrained discriminator $q_\theta(y|\mathbf{x})$ may provide high maximum softmax probabilities for some OOD samples due to distributional vulnerability. According to the definition of the Shannon entropy [48], we know that the entropy values of high-confidence OOD samples are low. Accordingly, we define the entropy of a sample $\mathbf{x}$ as

$$H_{\theta,\mathbf{x}}(C) = -\sum_{y \in [C]} q_\theta(y|\mathbf{x}) \log q_\theta(y|\mathbf{x}). \tag{3.2}$$

The range of $H_{\theta,\mathbf{x}}(C)$ is $(0, \log C]$. Inspired by the joint energy-based model (JEM) [51], which infers a density model for inputs by re-interpreting the logits obtained from networks, we construct an implicit generator $q_\theta(\mathbf{x})$ for the discriminator $q_\theta(y|\mathbf{x})$ by assuming that the generator is proportional to the negative entropy, i.e.,

$$q_\theta(\mathbf{x}) \propto -H_{\theta,\mathbf{x}}(C) + c \triangleq G(\mathbf{x}), \tag{3.3}$$

where a constant $c \geq \log C$ ($C \geq 0$) is added to ensure that the probability $q_\theta(\mathbf{x})$ is proportional to a non-negative value. Based on the negative entropy, the samples drawn from G should have high-confidence predictions without necessarily having the same discriminator outputs as ID samples. Therefore, FIG tends to generate OOD samples with distributional shift from training ID samples and high-confidence predictions. In JEM, the density model is inferred by re-interpreting the logits and marginalizing the label without constraints on the logit outputs. Therefore, the samples drawn from JEM are unnecessary for high-confidence predictions, and JEM tends to generate samples similar to ID samples to ensure that they have the same logit outputs. In summary, the negative entropy enables an implicit generator to generate OOD samples compared to JEM which generates ID samples.

However, sampling from G is intractable because we cannot construct an analytic expression of the probability distribution $q_\theta(\mathbf{x})$ based on $G(\mathbf{x})$. Recall that the entropy value of a high-confidence OOD sample is expected to be low, its $G(\mathbf{x})$ thus should be large. Accordingly, we specify a tractable probability distribution by exploring the upper bound of $G(\mathbf{x})$.

Assuming $h(\mathbf{x}) = \sum_{y' \in [C]} \exp f_\theta(\mathbf{x}, y')$ and substituting Eq. (4.15) and Eq. (3.2) into Eq. (3.3), we have

$$\begin{aligned} G(\mathbf{x}) &= \sum_{y \in [C]} \frac{\exp f_\theta(\mathbf{x}, y)}{h(\mathbf{x})} \log \frac{\exp f_\theta(\mathbf{x}, y)}{h(\mathbf{x})} + c \\ &= \frac{\sum_{y \in [C]} f_\theta(\mathbf{x}, y) \exp f_\theta(\mathbf{x}, y)}{h(\mathbf{x})} + \log \frac{\exp c}{h(\mathbf{x})}. \end{aligned} \tag{3.4}$$

To form a tractable bound, we set an upper bound on the second term of the last equality in Eq. (3.4) using inequality: $\log(x) \leq \frac{x}{a} + \log(a) - 1$ for all $x, a \geq 0$, which is derived from the basic logarithm inequality $\log(1+y) \leq y$, for $y > -1$ by assuming $y = \frac{x}{a} - 1$, and obtain the following inequality,

$$\log \frac{\exp c}{h(\mathbf{x})} \leq \frac{\exp c}{h(\mathbf{x}) a(\mathbf{x})} + \log a(\mathbf{x}) - 1 = \frac{\exp(c-1)}{h(\mathbf{x})}. \tag{3.5}$$

We obtain the above equality by setting $a(\mathbf{x})$ as Euler's number e because the inequality

holds for any choice of $a(\mathbf{x}) \geq 0$. Substituting Eq. (3.5) into Eq. (3.4), we have

$$\begin{aligned} G(\mathbf{x}) &\leq \frac{\sum_{y\in[C]} f_\theta(\mathbf{x}, y) \exp f_\theta(\mathbf{x}, y) + \exp(c-1)}{h(\mathbf{x})} \\ &= \left[\exp\left(\underbrace{\log \frac{\sum_{y\in[C]} \exp f_\theta(\mathbf{x}, y)}{\sum_{y\in[C]} f_\theta(\mathbf{x}, y) \exp f_\theta(\mathbf{x}, y) + \exp(c-1)}}_{\triangleq A(\mathbf{x})} \right) \right]^{-1}. \end{aligned} \tag{3.6}$$

To further obtain a tractable bound of $G(\mathbf{x})$, we need a lower bound on $A(\mathbf{x})$. According to the Jensen's inequality and inequality [41] $\frac{x}{x+1} \leq \log(1+x) \leq x$ for all $x \geq -1$, respectively:

$$\log \sum_{y\in[C]} \exp f_\theta(\mathbf{x}, y) \geq \sum_{y\in[C]} f_\theta(\mathbf{x}, y), \tag{3.7}$$

and

$$\begin{aligned} &\log \left(\sum_{y\in[C]} f_\theta(\mathbf{x}, y) \exp f_\theta(\mathbf{x}, y) + \exp(c-1) \right) \\ \leq &\sum_{y\in[C]} f_\theta(\mathbf{x}, y) \exp f_\theta(\mathbf{x}, y) - \exp(c-1) + 1. \end{aligned} \tag{3.8}$$

Substituting Eq. (3.7) and Eq. (3.8) into $A(\mathbf{x})$, we have

$$A(\mathbf{x}) \geq \underbrace{\sum_{y\in[C]} f_\theta(\mathbf{x}, y) \left(1 - \exp f_\theta(\mathbf{x}, y)\right)}_{\triangleq E_\theta(\mathbf{x})} - \left(1 - \exp(c-1)\right). \tag{3.9}$$

Therefore, we obtain the upper bound of G by substituting Eq. (3.9) into Eq. (3.10):

$$\begin{aligned} G(\mathbf{x}) \leq &\exp\left(-E_\theta(\mathbf{x}) + (1 - \exp(c-1))\right) \\ = &\exp\left(-E_\theta(\mathbf{x})\right) \cdot \exp\left(\exp\left(c-1\right) - 1\right) \\ = &\frac{\exp\left(-E_\theta(\mathbf{x})\right)}{\int \exp\left(-E_\theta(\mathbf{x}')\right) \mathrm{d}\mathbf{x}'} \cdot c', \end{aligned} \tag{3.10}$$

where $\int \exp\left(-E_\theta(\mathbf{x}')\right) \mathrm{d}\mathbf{x}'$ is a normalizing constant and

$$c' = \int \exp\left(-E_\theta(\mathbf{x}')\right) \mathrm{d}\mathbf{x}' \cdot \exp\left(\exp\left(c-1\right) - 1\right), \tag{3.11}$$

is a constant which is greater than or equal to zero and is independent of $\mathbf{x}$. Recall that $q_\theta(\mathbf{x}) \propto G(\mathbf{x})$, instead of directly solving $G(\mathbf{x})$ which is intractable, according to the upper bound Eq. (3.10), we take a tractable $q_\theta(\mathbf{x})$ by dropping the constant c', resulting

in:

$$q_\theta(\mathbf{x}) \propto \frac{\exp\left(-E_\theta(\mathbf{x})\right)}{\int \exp\left(-E_\theta(\mathbf{x}')\right) \mathrm{d}\mathbf{x}'}. \tag{3.12}$$

Therefore, we obtain the generator $q_\theta(\mathbf{x})$ from the given discriminator $q_\theta(y|\mathbf{x})$ without retraining, and $q_\theta(\mathbf{x})$ has the same parameter θ as $q_\theta(y|\mathbf{x})$. Thus, $q_\theta(\mathbf{x})$ is the implicit generator of the pretrained discriminator $q_\theta(y|\mathbf{x})$.

3.2.2 Langevin Dynamic Sampler

We cannot easily draw samples from $q_\theta(\mathbf{x})$ because we do not have an analytic expression for $q_\theta(\mathbf{x})$, which needs to integrate $\int \exp\left(-E_\theta(\mathbf{x}')\right) \mathrm{d}\mathbf{x}'$ with respect to $\mathbf{x}'$. However, $q_\theta(\mathbf{x})$ is an energy-based generative model [49] where $E_\theta(\mathbf{x})$ is the energy function. Relying on Markov chain Monte Carlo (MCMC) [52] methods, random walk or Gibbs sampling [53] can be applied, but both of these have long mixing time. To solve this challenge, Langevin dynamics [50], which uses the gradient of the energy function, can draw high-dimensional samples efficiently for energy-based models. Following the sampling method for energy-based models [54], we apply the Langevin dynamic sampler for the implicit generator $q_\theta(\mathbf{x})$ and have

$$\begin{aligned} \widetilde{\mathbf{x}}_t &= \widetilde{\mathbf{x}}_{t-1} - \frac{\epsilon_t}{2} \nabla_{\mathbf{x}} E_\theta(\widetilde{\mathbf{x}}_{t-1}) + \mathbf{z}_t, \\ \mathbf{z}_t &\sim \mathcal{N}(0, \epsilon_t \cdot \mathbf{I}), \\ \widetilde{\mathbf{x}}_0 &\sim p_0(\mathbf{x}), \end{aligned} \tag{3.13}$$

where $p_0(\mathbf{x})$ is an uniform distribution $\mathcal{U}(-1, 1)$, ϵ is a decayed step-size, and $\mathbf{I}$ is an identity matrix. The theoretical results provided by Welling and Teh [50] guarantee that $\widetilde{\mathbf{x}}_T$ is a sample generated from the distribution defined by the energy function as the number of iterations T becomes infinite and the step-size ϵ_t is close to zero, that is

$$\widetilde{\mathbf{x}}_T \approx \widetilde{\mathbf{x}} \sim q_\theta(\mathbf{x}) (\epsilon_t \to 0 \,\text{and}\, T \to \infty). \tag{3.14}$$

According to Eq. (3.13), the optimization of Langevin dynamics can be treated as finding a local optimal solution $\widetilde{\mathbf{x}}_T$ from a posterior distribution that minimizes the energy function $E_\theta(\mathbf{x})$. In this aspect, Langevin dynamics is similar to stochastic gradient descent [17]. However, one clear difference between them lies in that Langevin dynamics injects noise into the parameter updates, which ensures that the trajectory of the parameters will converge to the whole posterior distribution rather than just the point with the highest posterior probability. Beyond that, Langevin dynamics is significantly different from the projected gradient descent method [55] applied in adversarial learning, where the former finds a local optimal point, but the latter finds a saddle point for a min-max problem.

Note that, $E_\theta(\mathbf{x})$ could be infinite because the large output value $f_\theta(\mathbf{x}, y)$ in $E_\theta(\mathbf{x})$ can lead to the infinite exponential value $\exp f_\theta(\mathbf{x}, y)$. Hence, instead of using $E_\theta(\mathbf{x})$ to construct the implicit generator $q_\theta(\mathbf{x})$, we apply the modified version

$$\widehat{E}_\theta(\mathbf{x}) = \sum_{y \in [C]} \frac{f_\theta(\mathbf{x}, y)}{c} \left(1 - \exp \frac{f_\theta(\mathbf{x}, y)}{c}\right), \tag{3.15}$$

where c is a constant to narrow $f_\theta(\mathbf{x}, y)$. The step-size ϵ_t is updated by

$$\epsilon_{t+L} = \epsilon_t \cdot \gamma, \tag{3.16}$$

where γ is the decay rate, and L is the decay period. Following the process of generating adversarial samples [36], only the direction information is adopted to update the generated sample. This trick can improve sampling efficiency and avoid exploding gradients. We also clip the updated samples to the range $[-1, 1]$ to ensure consistency with the normalized input samples, i.e.,

$$\widetilde{\mathbf{x}}_t = \text{clip}\left(\widetilde{\mathbf{x}}_{t-1} - \frac{\epsilon_t}{2}\text{sign}(\nabla_{\mathbf{x}} \widehat{E}_\theta(\widetilde{\mathbf{x}}_{t-1})) + \mathbf{z}_t, -1, 1\right). \tag{3.17}$$

In practice, it is impossible and unnecessary to generate OOD samples by following the theoretical results proposed by Welling and Teh [50] to run Eq. (3.17) an unlimited number of times, as shown in Eq. (3.14). The prediction confidence for OOD samples is expected to be low, and only high-confidence OOD samples should be penalized to patch the distributional vulnerability. We thus only need to explore the high-confidence OOD samples to reveal the distributional vulnerability and ignore the low-confidence OOD samples. Therefore, we stop the iteration Eq. (3.17) until the confidence score of a generated OOD sample converges. The Langevin dynamic sampler (LDS) for generating an OOD sample is summarized in Algorithm 1.

Note that we find the confidence score of a generated OOD sample can converge for a small maximum iteration $T \in [10, 100]$, which means that the step size ϵ_t does not need to change to pursue high OOD detection performance. However, more iterations are required to generate visually meaningful images for visualization where the step-size should be adjusted to guarantee convergence. We further discuss the effect of the number of iterations T on the visualization experiments.

Accordingly, we can reveal the distributional vulnerability of a given discriminator by sampling discriminator-specific OOD samples in terms of Eq. (3.17). We expect that all OOD samples have low prediction confidence, while the existence of vulnerability makes it impossible. Note that ID samples also have high confidence predictions and low entropy. Based on the assumption for implicit generators which are proportional to negative entropy, the generated samples per Eq. (3.17) can also be ID. We assume

Algorithm 1 Langevin Dynamic Sampler (LDS)

Input: discriminator $q_\theta(y|\mathbf{x})$
Initialize $\widetilde{\mathbf{x}}_0 \sim \mathcal{U}(-1, 1)$, ϵ_0, γ, L
while not converged **do**
  $\mathbf{z}_t \sim \mathcal{N}(0, \epsilon_t \cdot \mathbf{I})$
  $\widetilde{\mathbf{x}}_t = \text{clip}(\widetilde{\mathbf{x}}_{t-1} - \frac{\epsilon_t}{2}\text{sign}(\nabla_{\mathbf{x}}\widehat{E}_\theta(\widetilde{\mathbf{x}}_{t-1})) + \mathbf{z}_t, -1, 1)$
  $\epsilon_{t+L} = \epsilon_t \cdot \gamma$
end while
Output: $\widetilde{\mathbf{x}}_t$

that most drawn samples are OOD because ID samples have limited classes while the generated samples are diverse. Different from the generative adversarial network [56] that learns from training ID samples to generate real-world objects, the implicit generator inferred from a pretrained discriminator aims to reveal its distribution vulnerability. Therefore, the OOD samples drawn from the implicit generator are required to have high confidence and differ from ID samples, which, however, unnecessarily correspond to real-world objects. According to the negative entropy principle of implicit generators, the generated samples have high-confidence predictions and are not necessary to satisfy the same distribution as training ID samples. The visualization results presented in Fig. 3.3 and Fig. 3.5 verify that the generated samples have high-confidence predictions and semantic shift [5]. Therefore, the generated samples are almost OOD. Furthermore, even if some generated samples follow the ID, they will not affect the patching of the distributional vulnerability, as discussed in the next section.

3.2.3 Confidence Penalty on Out-of-distribution Samples

Due to distributional vulnerability, the predictions by a pretrained discriminator for OOD samples are uncertain. Some OOD samples thus have unexpected high-confidence predictions. Therefore, the high-confidence OOD samples from an implicit generator can be applied to reveal the distributional vulnerability of the corresponding pretrained discriminator. Accordingly, a natural suggestion to patch this vulnerability is to penalize the OOD samples by flattening their class probabilities. Because the implicit generator depends on the corresponding pretrained discriminator, we can improve the OOD sensitivity of the pretrained discriminator by making it difficult for the corresponding implicit generator to generate high-confidence OOD samples. Specifically, an implicit generator is proportional to negative entropy to ensure that the generated OOD samples have high confidence predictions, and we correspondingly penalize these OOD samples by encouraging them to have large entropy, i.e.,

$$\max_{\theta} \mathbb{E}_{p(\mathbf{x},y)} \log q_\theta(y|\mathbf{x}) - \mathbb{E}_{q_\theta(\mathbf{x})} \sum_{y' \in [C]} q_\theta(y'|\mathbf{x}) \log q_\theta(y'|\mathbf{x}). \tag{3.18}$$

Algorithm 2 FIG: Fine-tuning Discriminators by Implicit Generators

Input: pretrained discriminator $q_\theta(y|\mathbf{x})$,
OOD percentage K,
learning rate μ
repeat
  Draw b ID samples from $\mathcal{D}_I$
  Draw $b \cdot K$ OOD samples from LDS$(q_\theta(y|\mathbf{x}))$
  Estimate objective function: $\mathcal{L}(\theta)$
  Obtain gradients: $\nabla_\theta \mathcal{L}(\theta)$
  Update parameters: $\theta = \theta + \mu \nabla_\theta \mathcal{L}(\theta)$
until convergence
Output: fine-tuned discriminator $q_\theta(y|\mathbf{x})$

After updating parameter θ, we obtain an updated discriminator. Also, we can derive a new implicit generator and obtain the newly generated OOD samples for the next iteration. We learn parameter θ iteratively until the implicit generator barely generates the OOD samples with high confidence predictions. Although some of the generated ID samples are also encouraged to have flat class probabilities, the rest of the generated OOD samples can still patch the vulnerability, and the dominated cross-entropy maintains the classification ability of the discriminator. In an extreme case where all generated samples are ID, the objective function Eq. (3.18) degenerates into the neural network confidence penalty method [57], which has empirically been demonstrated to improve the generalization ability.

We apply the stochastic gradient descent (SGD) [17] optimization algorithm to estimate the gradient of the objective function Eq. (3.18). For the ID training dataset $\mathcal{D}_I$ containing N ID samples, we draw $N \cdot K$ samples from the implicit generator $q_\theta(\mathbf{x})$ to construct the OOD training dataset $\mathcal{D}_O$, where $K \in [0, 1]$ is a hyper-parameter indicating the percentage of the generated OOD samples. In line with the idea of Monte Carlo [58], we estimate the objective function Eq. (3.18) by

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{(\mathbf{x}_I, y_I) \in \mathcal{D}_I} \log q_\theta(y_I|\mathbf{x}_I) - \frac{1}{N \cdot K} \sum_{y' \in [C]} \sum_{\mathbf{x}_O \in \mathcal{D}_O} q_\theta(y'|\mathbf{x}_O) \log q_\theta(y'|\mathbf{x}_O). \tag{3.19}$$

Algorithm 2 summarizes the process of FIG to patch the distributional vulnerability of a pretrained discriminator with the OOD samples generated by its corresponding implicit generator.

3.3 Experiments

In this section, we demonstrate FIG effectiveness[2] in comparison with the existing methods to detect OOD samples. Furthermore, we analyze the sensitivity of hyper-parameters in the LDS and the objective function of FIG. Also, we analyze the transferability of the generated OOD samples, i.e., OOD samples drawn from an implicit generator of a discriminator cannot be applied to patch the vulnerability of other discriminators with different network architectures. Finally, we present the visualization results to confirm that the generated OOD samples can effectively train OOD-sensitive discriminators.

3.3.1 Setup

The ID datasets for pretraining and fine-tuning discriminators are SVHN [59], CIFAR10 [60], CIFAR100 [60], and MiniImageNet [61]. The number of classes in these four datasets are 10, 10, 100, and 100, respectively. We follow the standard data augmentation practice for training samples. Specifically, we apply Resize(256) and RandomCrop((224,224)) to the samples in MiniImageNet and RandomCrop(32, padding=4) and RandomHorizontalFlip() to the samples in the other three datasets.

To test OOD detection performance, the corresponding test samples of an ID training dataset are treated as ID, and the samples from the other four real image datasets are treated as OOD. The OOD datasets used are LSUN [62], TinyImageNet [63], Caltech256 [64], and COCO [65]. When the ID training dataset is MiniImageNet, we also treat the test samples from CIFAR10 and CIFAR100 as OOD samples. Because OOD samples come from distinct datasets with varying input sizes, following the methods proposed in ODIN [24], we resize or crop each OOD sample to maintain the same size as the ID samples. (r) and (c) represent resized and randomly cropped samples, respectively. For a fair comparison, following the setup of the baseline and state-of-the-art methods [6, 24, 30, 27, 37, 28], validation datasets are unavailable to validate the hyper-parameters because OOD detection should consider the detection performance on diverse OOD samples that are unobservable in the validation phase.

Four advanced neural network architectures, namely ResNet18 [1], VGG19 [66], ShuffleNetV2 [67] and DenseNet100 [68], are used to create the discriminators. In the pretrained phase, their learning rates start at 0.1 and are divided by 10 after 100 and 150 epochs, and all networks are trained for 200 epochs on the training sets with 128 samples per mini-batch.

If not specified, the FIG setup is as follows. The same ID dataset is used to train and fine-tune pretrained discriminators. The fine-tuning process uses the learning rate

[2]The source codes are available at: `https://github.com/Lawliet-zzl/FIG`.

Table 3.1 FIG: OOD detection performance of pretrained and fine-tuned discriminators with diverse detectors.

Each value represents the average AUROC across eight OOD datasets, including LSUN(r), LSUN(c), TinyImageNet(r), TinyImageNet(c), Caltech256(r), Caltech256(c), COCO(r) and COCO(c). All the values are in percentage, and the boldface values represent relatively better detection performance.

in-dist	network	Baseline	ODIN	MLB	DRF	GM
		Pretrained / Fine-tuned (FIG)				
SVHN	ResNet18	92.1 / **98.5**	93.9 / **98.8**	94.8 / **97.5**	93.8 / **98.5**	87.6 / **98.6**
	VGG19	92.0 / **98.1**	92.8 / **98.4**	92.6 / **97.2**	92.8 / **98.2**	91.3 / **98.5**
	ShuffleNetV2	96.7 / **98.8**	98.1 / **99.3**	93.1 / **96.4**	97.0 / **98.8**	98.2 / **99.4**
	DenseNet100	91.3 / **97.6**	92.8 / **97.7**	95.4 / **96.8**	91.4 / **97.4**	77.9 / **96.2**
CIFAR10	ResNet18	91.2 / **95.0**	92.3 / **95.8**	91.5 / **94.3**	91.7 / **95.0**	91.0 / **95.6**
	VGG19	88.2 / **92.2**	89.0 / **92.8**	88.5 / **90.5**	89.0 / **92.9**	89.0 / **92.3**
	ShuffleNetV2	88.7 / **92.1**	91.4 / **95.2**	89.5 / **92.7**	87.2 / **91.4**	90.4 / **94.7**
	DenseNet100	90.8 / **94.9**	90.0 / **94.2**	91.4 / **93.9**	91.5 / **93.2**	90.9 / **94.8**
CIFAR100	ResNet18	82.6 / **89.3**	84.6 / **90.2**	80.5 / **87.7**	71.2 / **74.4**	80.4 / **87.0**
	VGG19	76.1 / **82.7**	78.9 / **84.6**	77.4 / **84.3**	78.4 / **82.8**	72.3 / **81.3**
	ShuffleNetV2	74.4 / **81.4**	83.2 / **86.2**	79.7 / **82.8**	80.8 / **86.4**	81.4 / **90.2**
	DenseNet100	83.0 / **93.1**	86.3 / **93.4**	83.5 / **93.9**	75.0 / **78.1**	72.7 / **85.5**

$\mu = 0.001$ which is equal to the final learning rate in the pretraining phase. For the modified energy function Eq. (3.15), we set the constant $c = 5$ because this value is sufficient to ensure that the exponential value is within the computer numerical range. For the LDS, following the suggestions of Welling and Teh [50], we set the step-size initialization $\epsilon_0 = 0.1$, the decay rate $\gamma = 0.9$, and the decay period $L = 100$. We set the OOD percentage $K = 0.1$ to balance the effectiveness and efficiency according to prior knowledge. We further discuss the effect of K in Section 3.3.4.

3.3.2 Incorporating OOD Detectors into FIG

We incorporate diverse state-of-the-art OOD detectors into a pretrained discriminator and its fine-tuned discriminator. The pretrained discriminator is learned from a training ID dataset, and its corresponding fine-tuned discriminator is obtained by fine-tuning the discriminator with the OOD samples generated by its implicit generator.

We incorporate five different OOD detectors, the baseline [6], ODIN [24], MLB [26], DRF [27], and GM [28], into FIG. The baseline [6] directly defines the maximum softmax output value from a discriminator as the OOD score without any hyperparameters. For ODIN [24], we select the temperature in $\{1, 2, 5, 10, 20, 50, 100, 200, 500, 100\}$ and the perturbation magnitude of 21 evenly spaced numbers starting from 0 and end-

ing at 0.004, and the best results are reported. For MLB [26], we tune the magnitude of noise in $\{0, 0.0005, 0.001, 0.0014, 0.002, 0.0024, 0.005, 0.01, 0.05, 0.1, 0.2\}$. For a fair comparison, we add the scores from different layers without training a logistic regression on a validation OOD dataset in MLB. For DRF [27], the magnitude of noise is 0.05 for CIFAR10 and SVHN and 0.0025 for CIFAR100. For GM [28], the order of computing feature correlations falls in the set $\{1, \ldots, 10\}$.

We summarize the results in Table 3.1, which shows that a fine-tuned discriminator achieves a significant improvement (1.22% to 23.49%) over its corresponding pretrained discriminator. Specifically, the discriminator fine-tuned by FIG achieves significant detection performance for both the detectors that apply the softmax outputs and the feature embeddings from network layers. This shows that FIG can improve the OOD sensitivity of a pretrained discriminator and alleviate the feature collapse problem [25]. According to the learning principle of FIG, the fundamental reason for its OOD detection improvement is that the distributional vulnerability of a pretrained discriminator has been effectively patched by the samples generated by its corresponding implicit generator.

3.3.3 Comparison Results

To verify the quality of the OOD samples generated by the implicit generators, we compare FIG with five state-of-the-art methods that retrain or fine-tune pretrained discriminators, namely Gaussian (GS) [14], MIXUP [13], adversarial (AD) [36], joint confidence loss (JCL) [37], and DeConf-C (DCC) [30]. For a fair comparison, following the setup of the state-of-the-art methods [37, 28], we apply the embedded detector based on ODIN [24] for DCC and the baseline detector [6] for the other compared methods to calculate the OOD scores without loss of generality.

The settings of all the compared methods are the same as their original. To use samples drawn from the Gaussian distribution as OOD samples in GS, we adopt Algorithm 2 to fine-tune the pretrained discriminators for a fair comparison. Specifically, we replace the OOD samples drawn from the Langevin dynamic sampler with Gaussian noise samples in Algorithm 2. As for MIXUP, the mixing coefficients that control the interpolation strength between sample pairs are drawn from Beta$(1, 1)$ for all ID datasets. When using adversarial samples as the generated OOD samples in AD to retrain the pretrained discriminators, we set the perturbation magnitude as 0.1 and the weights of both the cross-entropy loss and the adversarial objective function as 0.5. Another advanced method JCL retrains a pretrained discriminator with a generative adversarial network (GAN) [56] and encourages the softmax probabilities of generated samples to satisfy a uniform distribution. For JCL, we use mini-batch size 128 and regularization coefficient 1 of the Kullback-Leibler (KL) divergence term for SVHN, and the two hyper-parameters are 64 and 0.1 respectively for the other three training ID

Table 3.2 FIG: OOD detection performance for networks learned on SVHN, CIFAR10, and CIFAR100.

The value for an OOD dataset indicates its corresponding AUROC presented as a percentage, and the values for Ave. indicate the average AUROC across all the test OOD datasets. Boldface values represent a relatively better detection performance.

In-dist	Out-of-dist	GS / MIXUP / AD / JCL / DCC / FIG			
		ResNet18	VGG19	ShuffleNetV2	DenseNet100
SVHN	LSUN(r)	99.2 / 95.4 / 94.7 / 98.7 / 94.5 / **99.4**	98.5 / 96.3 / **99.7** / 98.3 / 91.0 / 99.3	98.0 / 95.9 / 96.4 / 91.6 / 99.1 / **99.7**	95.4 / 95.9 / 90.0 / 91.4 / **98.4** / **98.4**
	LSUN(c)	98.4 / 92.7 / 95.8 / **99.7** / 97.6 / 97.1	98.9 / 94.3 / 96.4 / **98.2** / 97.0 / 98.0	96.7 / 92.7 / 95.2 / 94.3 / **98.8** / 97.7	92.9 / 95.2 / 90.8 / 91.8 / **99.1** / 95.9
	TinyImageNet(r)	99.0 / 95.2 / 95.1 / 98.9 / 94.9 / **99.4**	98.6 / 96.8 / **99.7** / 98.3 / 93.4 / 99.1	98.0 / 96.2 / 96.8 / 92.9 / **99.9** / 99.6	95.5 / 95.7 / 91.2 / 90.3 / 98.0 / **98.8**
	TinyImageNet(c)	98.7 / 94.8 / 96.6 / **99.7** / 97.4 / 98.2	99.2 / 95.7 / **99.5** / 98.4 / 95.7 / 98.8	97.3 / 96.0 / 96.1 / 95.4 / 98.4 / **99.0**	93.5 / 95.9 / 91.3 / 91.9 / **99.5** / 97.5
	Caltech256(r)	95.9 / 90.9 / 93.3 / 90.4 / 92.5 / **97.2**	95.4 / 92.9 / 93.8 / 95.2 / 91.5 / **95.5**	95.2 / 92.7 / 94.2 / 91.2 / **97.8** / 97.3	91.9 / 93.5 / 88.8 / 89.6 / **95.5** / 95.3
	Caltech256(c)	97.7 / 91.7 / 94.1 / 97.3 / 94.5 / **98.8**	96.9 / 94.1 / 92.4 / 97.7 / 88.7 / **98.3**	96.5 / 90.5 / 92.5 / 92.6 / 98.5 / **98.7**	94.8 / 94.7 / 89.5 / 90.8 / 97.0 / **99.2**
	COCO(r)	97.5 / 92.9 / 94.6 / 94.6 / 95.1 / **98.4**	96.6 / 94.9 / **99.5** / 96.3 / 91.6 / 97.2	96.7 / 94.6 / 96.3 / 91.5 / 97.4 / **98.8**	94.2 / 95.3 / 88.4 / 90.0 / 96.1 / **96.7**
	COCO(c)	97.9 / 91.1 / 94.2 / 97.6 / 93.6 / **99.1**	97.2 / 93.7 / 97.2 / 98.1 / 88.4 / **98.4**	96.6 / 90.5 / 95.2 / 92.6 / 98.7 / **99.2**	95.0 / 93.9 / 91.0 / 90.9 / 97.2 / **99.4**
	Ave.	98.0 / 93.1 / 94.8 / 97.1 / 95.0 / **98.5**	97.7 / 94.8 / 97.3 / 97.6 / 92.2 / **98.1**	96.9 / 93.7 / 95.3 / 92.8 / 98.6 / **98.8**	94.2 / 95.0 / 90.1 / 90.8 / **97.6** / **97.6**
CIFAR10	LSUN(r)	92.8 / 92.8 / 91.9 / 90.8 / 98.7 / **99.0**	89.4 / 95.3 / 80.4 / 90.8 / 96.4 / **97.4**	83.0 / 83.5 / 81.4 / 88.8 / 98.6 / **99.8**	92.2 / 87.8 / 90.6 / 94.7 / **99.4** / 99.1
	LSUN(c)	95.0 / 95.7 / 94.1 / 90.8 / 98.2 / **98.9**	92.3 / 95.7 / 86.4 / 90.1 / **97.3** / 96.7	89.0 / 86.7 / 82.5 / 91.9 / **98.0** / 93.0	93.1 / 96.1 / 91.5 / 97.3 / **98.3** / 98.0
	TinyImageNet(r)	91.9 / 89.8 / 89.1 / 92.7 / 95.4 / **99.0**	86.8 / 93.9 / 78.7 / 84.3 / 92.4 / **96.4**	82.0 / 82.6 / 77.2 / 84.7 / **97.3** / 96.2	91.5 / 87.6 / 85.9 / 93.6 / **99.1** / 97.3
	TinyImageNet(c)	93.2 / 93.4 / 93.0 / 92.7 / **96.2** / 95.7	89.7 / 94.3 / 84.4 / 92.7 / 91.3 / **94.9**	87.4 / 85.9 / 85.8 / 88.2 / **96.5** / 92.2	92.3 / 93.4 / 89.3 / 96.2 / **98.7** / 96.5
	Caltech256(r)	86.9 / 80.0 / 85.9 / **92.9** / 85.0 / 88.0	82.5 / **86.1** / 76.1 / 84.3 / 80.4 / 83.4	79.3 / 78.9 / 76.3 / 81.2 / **84.6** / 83.0	86.7 / 79.5 / 85.1 / **90.1** / 87.6 / 87.8
	Caltech256(c)	93.0 / 90.3 / 91.5 / 84.3 / 91.7 / **94.7**	88.5 / **92.7** / 79.4 / 89.5 / 87.6 / 90.7	82.5 / 80.4 / 78.1 / 79.1 / 87.1 / **91.9**	91.0 / 89.9 / 90.8 / **95.2** / 91.3 / 94.4
	COCO(r)	87.9 / 83.9 / 87.2 / **91.7** / 85.9 / 90.5	85.0 / **88.2** / 79.4 / 85.2 / 81.0 / 86.5	80.5 / 79.9 / 80.8 / 82.3 / 85.1 / **88.3**	87.6 / 83.8 / 85.8 / 88.5 / 88.8 / **89.6**
	COCO(c)	92.7 / 87.5 / 91.6 / 85.2 / 89.9 / **94.5**	88.4 / 93.8 / 79.2 / 90.8 / 87.3 / **91.7**	84.1 / 81.6 / 78.7 / 79.6 / 87.9 / **92.4**	91.0 / 89.5 / 90.7 / 93.9 / 90.6 / **96.2**
	Ave.	91.7 / 89.2 / 90.5 / 90.1 / 92.6 / **95.0**	87.8 / 92.5 / 80.5 / 88.5 / 89.2 / **92.2**	83.5 / 82.4 / 80.1 / 84.5 / 91.9 / **92.1**	90.7 / 88.4 / 88.7 / 93.7 / 94.2 / **94.9**
CIFAR100	LSUN(r)	83.6 / 78.0 / 82.7 / 87.6 / 93.4 / **93.8**	79.2 / 75.4 / 71.5 / 80.7 / **87.3** / 82.5	71.9 / 55.9 / 68.8 / 65.7 / 80.4 / **82.3**	81.9 / 75.2 / 82.6 / 86.1 / **98.7** / 98.6
	LSUN(c)	85.4 / 77.6 / 81.5 / 80.5 / **88.3** / 85.0	83.7 / 80.9 / 78.3 / 81.9 / 85.6 / **85.9**	75.1 / 71.2 / 76.7 / 77.3 / **87.7** / 82.9	81.6 / 81.9 / 81.4 / 88.4 / **95.3** / 94.6
	TinyImageNet(r)	82.9 / 74.4 / 81.5 / 87.2 / 92.8 / **97.1**	76.6 / 75.6 / 70.9 / 80.5 / **81.6** / 80.0	72.5 / 61.1 / 64.4 / 63.5 / 78.4 / **84.7**	82.5 / 74.1 / 82.3 / 83.5 / 98.6 / **98.7**
	TinyImageNet(c)	87.1 / 83.7 / 83.8 / 83.3 / **91.4** / 89.6	83.4 / 81.8 / 77.3 / 79.9 / 83.9 / **87.2**	78.9 / 78.8 / 78.5 / 75.9 / **88.5** / 86.4	84.1 / 84.9 / 84.0 / 87.8 / **97.6** / 96.6
	Caltech256(r)	75.3 / 75.2 / 76.2 / 79.7 / **83.3** / 82.4	71.5 / 71.2 / 69.1 / **87.8** / 77.3 / 76.9	67.2 / 68.9 / 68.5 / 67.8 / **74.6** / 72.5	74.4 / 72.3 / 75.8 / 81.3 / 82.9 / **83.4**
	Caltech256(c)	82.1 / 83.7 / 81.6 / 83.6 / 87.9 / **89.2**	79.7 / 79.9 / 74.7 / 80.6 / 76.7 / **84.5**	71.7 / 70.0 / 71.5 / 71.2 / 76.3 / **81.5**	81.6 / 80.1 / 81.3 / 85.7 / 86.9 / **92.9**
	COCO(r)	77.4 / 78.8 / 78.8 / 80.2 / 83.2 / **84.1**	75.6 / 77.7 / 72.9 / 76.8 / **79.7** / 79.2	70.8 / 69.7 / 71.5 / 69.2 / **77.8** / 75.6	77.0 / 79.8 / 78.5 / 80.6 / 84.5 / **85.4**
	COCO(c)	83.2 / 80.5 / 82.2 / 82.5 / 89.0 / **93.4**	81.7 / 79.3 / 75.8 / 81.7 / 78.4 / **85.6**	71.8 / 70.5 / 71.1 / 71.6 / 78.3 / **85.6**	82.0 / 79.6 / 82.3 / 84.5 / 88.1 / **94.7**
	Ave.	82.1 / 79.0 / 81.0 / 83.1 / 88.7 / **89.3**	78.9 / 77.7 / 73.8 / 81.2 / 81.3 / **82.7**	72.5 / 68.3 / 71.4 / 70.3 / 80.2 / **81.4**	80.6 / 78.5 / 81.0 / 84.7 / 91.6 / **93.1**

datasets. For DCC, we adopt the cosine similarity in the scoring function and search for the adversarial perturbation magnitude with only ID samples.

The OOD detection results on SVHN, CIFAR10, and CIFAR100 are displayed in Table 3.2. Comparing all the methods, we observe that FIG does not achieve the best OOD detection performance on some ID and OOD dataset pairs. Lee et al. offer a possible explanation, i.e., the distribution of a specific OOD dataset does not effectively cover all tested out-of-distributions [37]. We thus verify the effect of FIG on different test OOD datasets, and FIG inevitably reduces the effect on some OOD samples in order to pursue the overall OOD detection improvement. Compared with GS, FIG obtains significant improvement (5.69%). We thus verify that the generated samples from the implicit generators are not simple high-confidence noise but informative images that can reveal discriminator vulnerability. For all neural architectures, compared with the other state-of-the-art methods, we find that FIG achieves the best OOD detection performance with an average of 3.29%, 5.34%, and 9.01% AUROC improvement on the three training ID datasets, SVHN, CIFAR10, and CIFAR100, respectively. We also perform experiments on a larger resolution dataset MiniImageNet, and the results are presented in Table 3.3. FIG achieves the most significant average AUROC value across all the test OOD datasets with an average of 10.56% AUROC improvement over the other state-of-the-art methods. Therefore, FIG is applicable for high-resolution samples. As a result, FIG achieves the best OOD detection performance. This is because the generated OOD samples of FIG are specific to the ID training dataset and network architecture. The data characteristics indicate that the generated OOD samples can be applied to patch the vulnerability of a pretrained network to improve OOD sensitivity.

The harmonic means of AUROC and accuracy are shown in Table 3.4. Although JCL and DCC achieve significant performance in detecting some OOD samples, as shown in Table 3.2, the corresponding harmonic means are close to the baseline method which only applies a pretrained network without modification. The results indicate that the two methods significantly sacrifice the classification ability to improve the OOD sensitivity. However, FIG achieves the most significant harmonic means on all ID training datasets, which indicates that FIG finds the best balance between classifying ID samples and detecting OOD samples. The reasons are two-fold: (1) OOD samples of FIG are generated from the implicit generator of a given pretrained discriminator; (2) fine-tuning the pretrained discriminator with the specific generated OOD samples will not seriously disturb the learning process of classifying ID samples.

In general, our FIG can improve the OOD detection performance and maintain high ID classification accuracy. We recall the diverse vulnerability of discriminators with different architectures to understand the reason behind this. Hence, OOD samples generated by particular generators cannot correspondingly address the discriminator-specific

vulnerability. FIG patches the vulnerability of a pretrained discriminator to improve OOD detection performance by the generated samples from its implicit generator, and the implicit generator knows what kind of samples are OOD for the pretrained discriminator. These conclusions also explain why FIG can balance OOD detection and ID classification performance after being fine-tuned on the generated OOD samples. The OOD samples are data- and network-adaptive, which enables the pretrained discriminator learn the knowledge from the ID samples with less interference.

Table 3.3 FIG: OOD detection performance for networks learned from MiniImageNet.

The value for an OOD dataset indicates its corresponding AUROC presented as a percentage, and the values for "Ave." indicate the average AUROC across all the test OOD datasets. Boldface values represent a relatively better detection performance.

Out-of-dist	GS	MIXUP	AD	JCL	DCC	FIG
CIFAR10(r)	87.2	81.7	80.8	83.1	93.8	**93.9**
CIFAR100(r)	84.8	79.5	76.8	80.2	91.6	**92.5**
Caltech256(r)	77.9	78.7	63.5	75.1	83.2	**84.6**
Caltech256(c)	80.9	79.2	82.1	84.2	85.0	**89.8**
COCO(r)	78.0	80.1	63.3	76.5	82.7	**84.6**
COCO(c)	81.0	81.9	82.6	85.7	85.2	**89.9**
Ave.	81.6	80.2	74.8	80.8	86.9	**89.2**

Table 3.4 FIG: Harmonic means of AUROC and accuracy.

Boldface values represent a relatively better balance between classifying ID samples and detecting OOD samples.

In-dist	Pretrained	GS	MIXUP	AD	JCL	DCC	FIG
SVHN	47.1	48.5	47.4	47.7	48.0	47.7	**48.7**
CIFAR10	46.6	46.7	46.2	44.9	45.8	46.8	**47.5**
CIFAR100	39.9	39.7	39.4	35.9	38.9	40.5	**41.2**
MiniImageNet	38.8	38.6	39.0	36.5	37.3	39.3	**40.0**

3.3.4 Hyper-parameter Analyses

This section empirically shows the impact of the OOD percentage K on the proposed FIG method. We test the effect of K by setting it to $0, 0.01, 0.1, 0.4, 0.7, 1$ respectively. We show the widespread applicability and stability of the hyper-parameter K on CIFAR10, SVHN, CIFAR100 with network architectures Resnet18, VGG19, ShuffleNetV2, and DenseNet100 in terms of AUROC. Note that when $K = 0$, FIG only applies training ID samples to fine-tune discriminators without generating OOD samples.

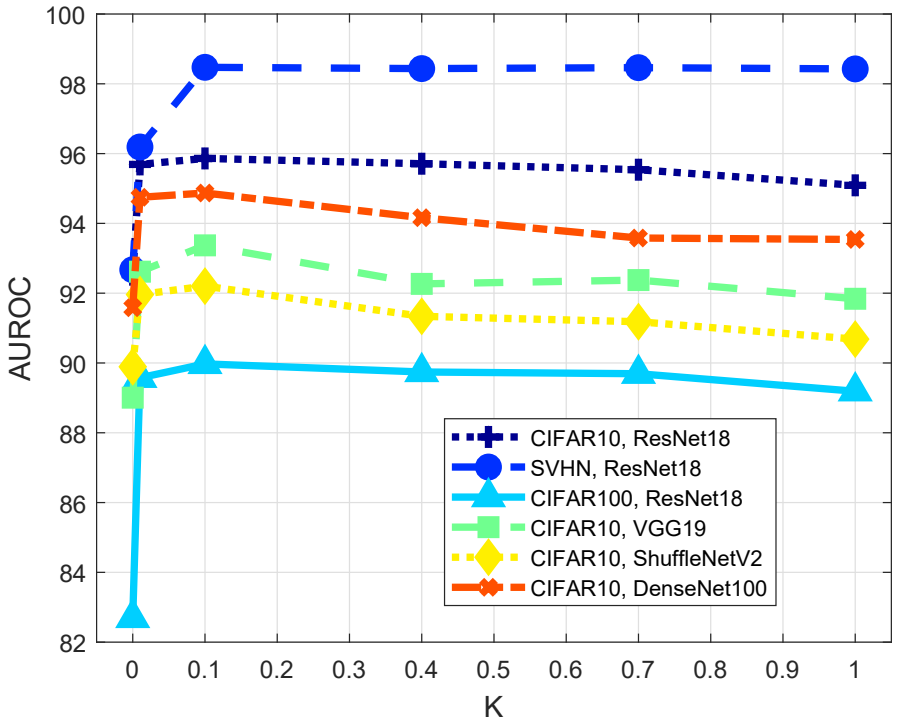


Fig. 3.1 FIG: Effect of the OOD percentage K.
Each point refers to an average AUROC score on the eight OOD datasets, including LSUN(r), LSUN(c), TinyImageNet(r), TinyImageNet(c), Caltech256(r), Caltech256(c), COCO(r) and COCO(c).

The results of verifying K are shown in Fig. 3.1. We observe that increasing the OOD percentage K can improve the detection performance, and the detection performance diminishes when K is sufficiently large ($K > 0.1$). However, having a large K with performance reduction is acceptable. Recall that an implicit generator depends on a pretrained discriminator, and the pretrained discriminator is updated by the OOD samples generated by the implicit generator. According to the design principle, implicit generators generate low-entropy samples that could be ID or OOD samples. We assume that most drawn samples are OOD because ID samples have limited classes while the generated samples are diverse. When K is sufficiently large ($K > 0.1$), more generated ID samples are encouraged to yield low-confidence predictions in the fine-tuning phase, which is contradictory to the expectation that a pretrained distribution should assign high-confidence predictions for training ID samples. Specifically, a large set of the generated samples contain more generated ID samples, which causes bias in the estimated gradients of the entropy in the objective function, and the dynamic implicit generator makes the biased estimation more serious. Therefore, a small OOD percentage such as $K \in [0.01, 0.1]$ is a better choice for FIG. Hence, we apply $K = 0.1$ to balance the effectiveness and the efficiency by default.

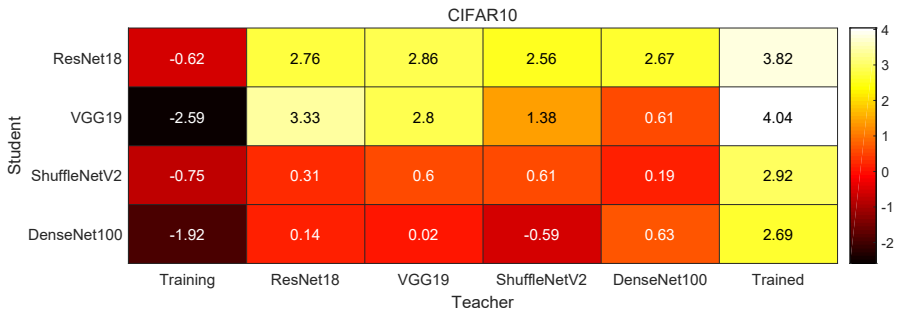


Fig. 3.2 FIG: Transferability of the generated OOD samples between two discriminators.

Each student is the discriminator in the objective function, and each teacher is the discriminator used to infer the implicit generator in the Langevin dynamic sampler. The training and trained teachers are initialized randomly by pretrained discriminators. Both training and trained teachers are continuously updated as the pretrained discriminators change. The teachers named by network architectures are randomly initialized, and their parameters are fixed during the learning process of students. A value in the colored boxes represents the percentage of AUROC improvement over the pretrained discriminator with the same student network architecture, and lighter colors reflect better results. For all columns except the first and last, diagonal entries correspond to answer **A1**, and off-diagonal entries correspond to **A2**. The entries in the first column correspond to **A3**.

3.3.5 Transferability Analyses

In FIG, the OOD samples are drawn from the implicit generator of a pretrained discriminator. The generated OOD samples are then used to patch the vulnerability of the discriminator. We analyze the transferability of the generated OOD samples to verify that (1) the implicit generator should be updated as the corresponding discriminator is updated; (2) OOD samples drawn from an implicit generator of a discriminator cannot be applied to patch the vulnerabilities of other discriminators with different network architectures; (3) FIG is not suitable for randomly initialized discriminators.

The discriminator following the Langevin dynamic sampler (LDS) and the discriminator in the objective function Eq. (3.18) can be treated as a teacher and a student, respectively. Therefore, a discriminator which learns from training ID samples is a student learning without teachers, and a discriminator trained by FIG is a student learning with a teacher. The teacher teaches the student how to find the vulnerability, and the student who receives the knowledge from the teacher then knows the previously unknown (i.e., the vulnerability). The teacher already has some knowledge of the network structure if the teacher is pretrained, and the teacher and the student learn from each other as the discriminator used in LDS is updated. Accordingly, we analyze the teacher from different perspectives and ask the following three questions:

- **Q1:** What if the teacher stops learning from the student? This corresponds to applying a fixed discriminator to infer an implicit generator in each iteration.

- **Q2:** What if the expertise of the teacher mismatches that of the student? This corresponds to generating OOD samples according to a discriminator to patch the vulnerability of other discriminators with different architectures.

- **Q3:** What if the teacher does not yet have enough knowledge or experience but still learns from the student? In this situation, the discriminator is trained from scratch, and the implicit generator is updated according to the training discriminator before each epoch.

To answer these questions, we design the following experiments and the results are shown in Fig. 3.2. We run FIG in terms of different teacher-student pairs on CIFAR10. We evaluate the improved performance over the baseline with the same network architecture as the student on detecting different OOD samples, including LSUN(r), LSUN(c), TinyImageNet(r), TinyImageNet(c), Caltech256(r), Caltech256(c), COCO(r) and COCO(c). In summary, the following findings address the above questions.

- **A1:** Similarly, we fix the discriminator in LDS and ensure this discriminator and the discriminator in the objective function have the same network structure. If pretrained discriminators are fixed in LDS for each iteration in the generation process, the detection performance will be worse than when the on-the-fly discriminators are used to infer implicit generators. The main reason for this is that vulnerability is dynamic as refining discriminators leads to new vulnerabilities, and this dynamic property requires the implicit generators to be updated continuously.

- **A2:** We replace the regularly updated discriminator in the LDS input with a fixed discriminator that is diversified with different architectures. These teachers have to be fixed because only the gradients of students are calculated in FIG, and the gradients for teachers are not available. When students and teachers have different architectures, the OOD detection performance generally declines since the generated OOD samples from a network do not match the vulnerabilities of networks with different architectures, which means that the generated OOD samples are model specific.

- **A3:** We replace the discriminator in the input list of Algorithm 2 with a randomly initialized discriminator and use the same training setup as the baseline where the discriminator is trained for 200 epochs, and the learning rates start at 0.1 and are divided by 10 after 100 and 150 epochs. It is important to give knowledge to teachers as we find fine-tuning a pretrained discriminator can achieve better performance than retraining a new one. This is because the capable discriminators deduce reliable implicit generators, which guarantees the right direction to patch the vulnerability.

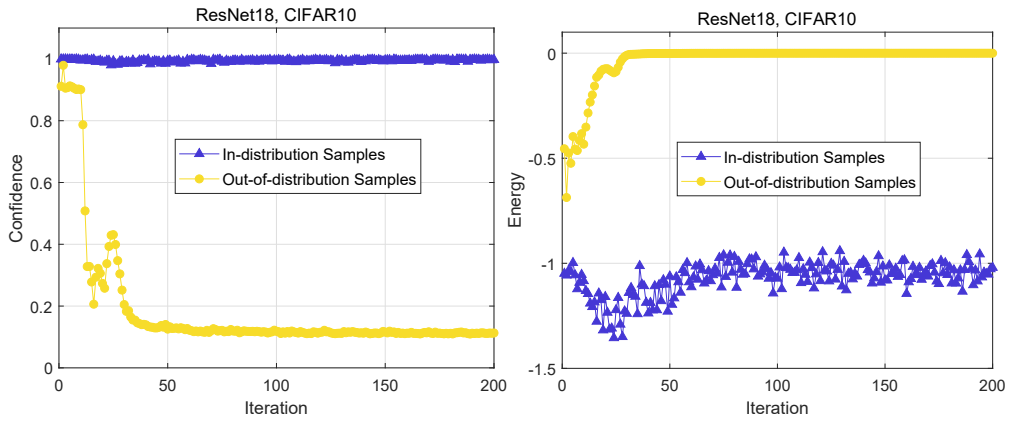


Fig. 3.3 FIG: Confidence and energy.
Each point indicates an average value of confidence or energy on the training ID dataset or a generated OOD dataset.

According to the transferability analyses, we apply FIG on a pretrained discriminator and continually update the pretrained discriminator and the corresponding implicit generator.

3.3.6 Visualization of the Results

The generated OOD samples from the implicit generators can be applied to train OOD-sensitive discriminators because these OOD samples have high confidence and are reliable and specific. This is verified by visualizing the change in the confidence and energy along the fine-tuning process, the embedding results, and the content and classes of the generated samples. The network architecture is Resnet18, and the training ID datasets are CIFAR10 and SVHN.

3.3.6.1 Confidence and Energy

We analyze FIG from the confidence and energy perspectives, respectively. We visualize the changes in confidence and energy on both training ID samples and the generated OOD samples along with the fine-tuning of the discriminators. For the OOD confidence, FIG should encourage low scores since the OOD sensitivity of discriminators can be improved by making it difficult for the corresponding implicit generators to produce OOD samples. For the OOD energy, the implicit generators should have high values according to the design principle.

The results are reported in Fig. 3.3. We find that ID samples maintain high-confidence scores and stable energy values. For the generated OOD samples, the confidence scores are close to one in the preliminary stage which then drop continuously. It is increasingly difficult for implicit generators to generate OOD samples since samples with a higher energy are explored as iterations increase. Although the energy of the generated OOD

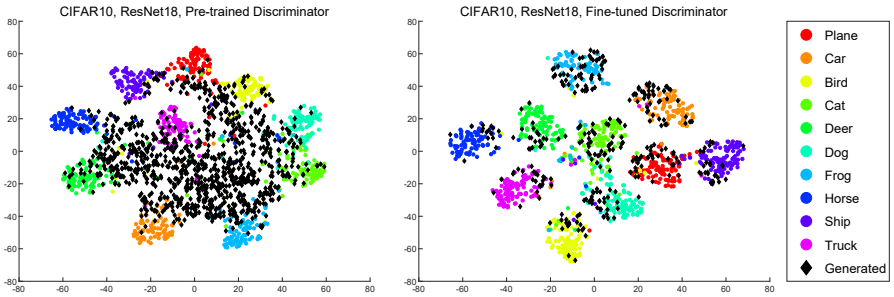


Fig. 3.4 FIG: Embedding results.
The black diamonds indicate the generated samples, and the colored circles represent the test ID samples.

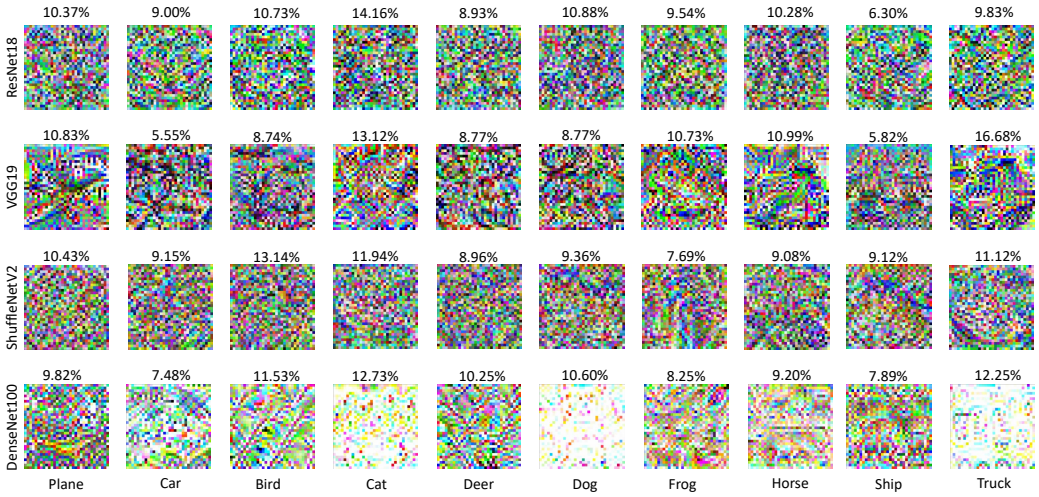
samples is higher than that of the ID samples, the distribution of the prediction probability vectors is approximate to a uniform distribution since the confidence scores are close to $0.1 = 1/\text{class-number}$ on the training dataset CIFAR10. Therefore, we conclude that implicit generators can produce high-confidence OOD samples in the preliminary stage, which then fails after the vulnerability is patched.

3.3.6.2 Embedding Visualization

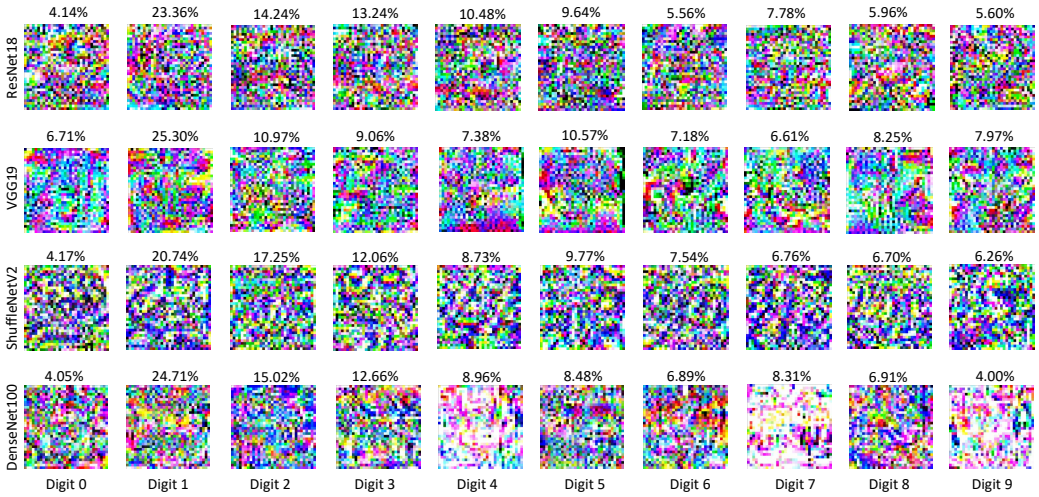
Fig. 3.4 presents the embedding results of test ID samples and the generated samples from a pretrained discriminator and a fine-tuned discriminator presented by t-SNE [9]. We randomly sample 10% of the test ID samples and draw $1,000$ samples from the implicit generators, and only the samples with confidence scores over 0.9 are plotted. The results show that the vulnerability exists in the pretrained discriminators because numerous OOD samples with high confidence are located in the external range of ID classes, and the vulnerability is commendably fixed after being fine-tuned by FIG because fewer ID samples can be drawn from implicit generators. As a result, the embedding results of the pretrained discriminators substantiate that high-confidence OOD samples can be drawn from its implicit generators. The embedding results for the fine-tuned discriminators verify that FIG effectively applies the generated OOD samples to patch the vulnerability.

3.3.6.3 Content and Class

Fig. 3.5 visually shows the 100% confidence OOD samples corresponding to different predicted classes. To generate visually meaningful images from implicit generators, we set an extensive maximum iteration number $T = 10,000$ in LDS. We then count the predicted class probabilities of the generated samples according to the outputs from the corresponding pretrained discriminators. We observe that different network architectures on different ID training datasets respond differently to the generated OOD samples and result in various predicted class distributions. For example, on CIFAR10,



(a) In-distribution dataset: CIFAR10



(b) In-distribution dataset: SVHN

Fig. 3.5 FIG: Generated OOD samples from implicit generators.
The value on top of each image represents the percentage of generated samples for the class (column) by the network (row). Each word in the horizontal direction and vertical direction represents a network architecture and a class name, respectively.

the implicit generator based on ResNet18 more likely generates samples for the class *Cat* while the generator based on VGG is more likely to generate samples for the class Truck. Furthermore, the generated samples on CIFAR10 and SVHN are quite different, even though the network architectures are the same. We thus verify that OOD samples drawn from implicit generators are discriminator-specific as different pretrained discriminators show distinct vulnerabilities. In general, various sources of vulnerability lead to diverse OOD samples, and it is essential to consider the specific OOD samples when patching the vulnerability of a pretrained discriminator.

3.4 Summary of This Chapter

In this Chapter, we propose a method of *fine-tuning discriminators by implicit generators* (FIG) to improve the OOD sensitivity of a given pretrained discriminator, which tackles the main challenge of generating discriminator-specific OOD samples. Specifically, we reveal the distributional vulnerability by the corresponding implicit generator inferred from a pretrained discriminator without extra training costs, draw OOD samples from the generator by a Langevin dynamic sampler, and patch the distributional vulnerability by penalizing the prediction confidence of these generated samples. We empirically demonstrate that FIG outperforms the existing methods in detecting OOD samples.

CHAPTER 4

Supervision Adaptation Balances In-distribution Generalization and Out-of-distribution Detection

4.1 Motivations

A network can improve OOD sensitivity by penalizing OOD samples, i.e., encouraging OOD samples to have low-confidence predictions. However, OOD samples are usually unavailable in training phases. Accordingly, chapter 3 discusses how to generate specific OOD samples for a given network by inferring its corresponding implicit generator. The OOD samples from real-world datasets or drawn from generative models should be provided labels if they are available in the training process. Although it is easy to provide manually-determined labels for OOD samples, the manually-determined labels cannot precisely describe the introduced OOD samples and could disrupt the learning process of classifying ID samples. To balance ID classification accuracy and OOD detection performance, the main challenge is to make the OOD samples adaptive to ID samples by designing adaptive supervision information for OOD samples. Therefore, an important requirement of the adaptive supervision information is to separate OOD from ID samples, which minimizes the impact of OOD samples on learning classifying ID samples.

Furthermore, to better ensure the above balance between ID generalization and OOD detection performance, we expect to improve the accuracy while pursuing high OOD detection performance. The adaptive supervision information can achieve this target if it can associate OOD samples with specific aspects of the data space (i.e., specific areas outside the coverage of ID samples), which makes ID samples with different labels more separable [69]. Note that the adaptive supervision information aims to make OOD samples compatible with ID samples, which ensures that ID and OOD samples can be mapped to different areas in the data space by networks. It further minimizes the impact of OOD samples on learning classifying ID samples. We thus have to explore the relationships between ID samples and between ID and OOD samples, which leads to that the supervision information can address the above two challenges: (1) adapting OOD samples to ID ones; and (2) making ID samples with different labels more

distinguishable.

According to the above discussion, this chapter manages the above challenges by introducing a *supervision adaptation* (SA) method to balance ID generalization ability and OOD detection capacity. Based on the assumption that there are no OOD samples in the data space, the traditional cross-entropy loss for ID samples is derived by mutual information maximization [41]. To understand the uncovered area of ID samples, we consider extending the mutual information maximization to mixed data space which contains both ID and OOD samples. Accordingly, we infer a lower bound of the mutual information measuring the dependency between ID samples and the corresponding labels in the mixed data space. Furthermore, we apply a tractable optimization problem over a parameterized discriminator to replace the intractable conditional distribution within the bound. This lower bound shows that the form of supervision information for OOD samples is the negative probabilities of all classes. To estimate the supervision information, we further improve the generalization ability by exploring the data relationships between ID and OOD samples. The main idea [69] is that applying an OOD dataset as a reference can make different classes of ID samples more distinguishable. Specifically, we solve a binary regression problem to separate OOD samples from a class of ID samples. We obtain a compact objective function of the SA method by simplifying the combined results of the lower bound on the mutual information and the estimated OOD supervision information.

4.2 Supervision Adaptation (SA)

The objective function is obtained from the two components *mixed space mutual information* (MSMI) and *multiple binary cross-entropy* (MBCE). MSMI and MBCE reveal the form of the supervision information of an OOD sample and estimate the supervision information, respectively. The objective function of SA is obtained by computing the lower bound of the combined results of MSMI and MBCE.

4.2.1 Problem Statement

Let $\mathbf{x}$ be a sample feature vector and y be a label. Let $P_I(\mathbf{x})$ and $P_O(\mathbf{x})$ denote in-distribution and out-of-distribution respectively. Accordingly, $P(\mathbf{x}) = (1-\epsilon)P_I(\mathbf{x}) + \epsilon P_O(\mathbf{x})$ represents a mixture distribution of ID and OOD samples where ϵ is a component parameter controlling the proportions of ID and OOD samples in the mixed data space. Because only ID samples have ground-truth labels, label-free OOD samples should not change the label distribution. We thus further assume the label distributions in the ID samples and the mixture data are the same, i.e., $P_I(y) = P(y)$, the join distributions of the two random variables $\mathbf{x}$ and y for the ID and the mixture dis-

tribution are the same, i.e., $P_I(\mathbf{x}, y) = P(\mathbf{x}, y)$, and the marginal distribution of $\mathbf{x}$ is $P_I(\mathbf{x}) = \sum_{y=1}^{K} P_I(\mathbf{x}, y) = \sum_{y=1}^{K} P_I(y|\mathbf{x})P_I(y)$, where K is the number of classes.

SA trains a parameterized discriminator $Q_\theta(y|\mathbf{x})$ with an ID dataset $\mathcal{D}^I$ and an OOD dataset $\mathcal{D}^O$ to estimate the conditional distribution $P(y|\mathbf{x})$ in the mixed data space. θ denotes the model parameter. We focus on the following research question: given a sample drawn from the mixture distribution $P(\mathbf{x})$, can the discriminator $Q_\theta(y|\mathbf{x})$ determine whether the sample follows the out-of-distribution $P_O(\mathbf{x})$ and accurately assign a label to this sample if it is drawn from the in-distribution $P_I(\mathbf{x})$?

4.2.2 MSMI: Mixed Space Mutual Information

SA characterizes the supervision information by exploring the data relationship between samples and labels in the mixed data space. The underlying idea is that an ID sample is strongly associated with its corresponding label, while an OOD sample is weakly coupled with any classes. Because of the weak couplings between classes and OOD samples, it is difficult to define the supervision information of OOD samples. We thus implicitly infer the supervision information of OOD samples by strengthening the association between ID samples and their labels in the mixed data space. This makes the networks aware of OOD samples and minimizes the OOD interference in classifying ID samples.

Mutual information (MI) is a quantity used to measure the relationship between random variables. Traditional approaches [48] assume that the data space only contains ID samples. They approximate the conditional distribution of label y given sample $\mathbf{x}$ in MI by a parametric discriminator. This assumption mismatches the complex situations where OOD samples exist. We thus extend MI to the mixed data space and measure the dependence between ID samples and its labels in the mixed space by

$$\mathcal{I}(X;Y) = \mathbb{E}_{P(\mathbf{x},y)}\left[\log \frac{P(\mathbf{x},y)}{P(\mathbf{x})P(y)}\right] = \mathbb{E}_{P_I(\mathbf{x},y)}\left[\log \frac{P_I(\mathbf{x},y)}{P(\mathbf{x})P_I(y)}\right]. \tag{4.1}$$

The data distribution in the traditional MI only considers ID samples. Differing from that, the distribution $P(\mathbf{x})$ in the mixed space MI is a mixture distribution of ID and OOD samples. Note that only ID samples have the ground-truth labels, and the class distributions of OOD samples are unavailable. We can thus only explicitly measure the relationships between ID samples and their labels in Eq. (4.1). Furthermore, we can implicitly derive the supervision information of OOD samples by maximizing Eq. (4.1) by the mixture distribution $P(\mathbf{x})$, which adapts OOD samples to ID samples.

For an observed sample drawn from the mixture distribution $P(\mathbf{x})$, it could be an ID sample linked to a label or a label-free OOD sample. According to the mixed space MI, we aim to estimate the conditional distribution $P(y|\mathbf{x})$ by learning a parametric

discriminator $Q_\theta(y|\mathbf{x})$ to maximize Eq. (4.1). Calculating the MI is challenging due to the unaccessible underlying distributions. However, computing the gradients of a lower bound [70] on MI concerning the parameter θ does not require directly estimating MI. To establish a lower bound on mutual information, we factorize MI and introduce a tractable discriminator $Q_\theta(y|\mathbf{x})$ to approximate the unknown conditional distribution $P(y|\mathbf{x})$, and we have

$$\begin{aligned}
\mathcal{I}(X;Y) &= \mathbb{E}_{P_I(\mathbf{x},y)}\left[\log \frac{P_I(\mathbf{x},y)}{P(\mathbf{x})}\right] - \mathbb{E}_{P_I(\mathbf{x})}\mathbb{E}_{P_I(y)}\left[\log P_I(y)\right] && (4.2)\\
&= \mathbb{E}_{P_I(\mathbf{x},y)}\left[\log \frac{P_I(\mathbf{x},y)}{P(\mathbf{x})}\right] + H(y) && (4.3)\\
&= \mathbb{E}_{P_I(\mathbf{x},y)}\left[\log \frac{P_I(\mathbf{x},y)Q_\theta(y|\mathbf{x})}{P(\mathbf{x})Q_\theta(y|\mathbf{x})}\right] + H(y), && (4.4)\\
& && (4.5)
\end{aligned}$$

where $H(y) = -\mathbb{E}_{P_I(y)}\left[\log P_I(y)\right] \in [0, \log K]$ is the entropy of variable y. According to the Bayes' theorem and the assumption $P(\mathbf{x}, y) = P_I(\mathbf{x}, y)$, we have

$$P(y|\mathbf{x}) = \frac{P(\mathbf{x},y)}{P(\mathbf{x})} = \frac{P_I(\mathbf{x},y)}{P(\mathbf{x})}. \tag{4.6}$$

Due to the nonnegativity of the entropy $H(y)$ and Eq. (4.6), we have

$$\begin{aligned}
\mathcal{I}(X;Y) \geq& \mathbb{E}_{P_I(\mathbf{x},y)}\left[\log Q_\theta(y|\mathbf{x})\right] + \mathbb{E}_{P_I(\mathbf{x},y)}\left[\log \frac{P_I(\mathbf{x},y)}{P(\mathbf{x})Q_\theta(y|\mathbf{x})}\right] && (4.7)\\
=& \mathbb{E}_{P_I(\mathbf{x},y)}\left[\log Q_\theta(y|\mathbf{x})\right] + \mathbb{E}_{P(y|\mathbf{x})P(\mathbf{x})}\left[\log \frac{P(y|\mathbf{x})}{Q_\theta(y|\mathbf{x})}\right] && (4.8)\\
=& \mathbb{E}_{P_I(\mathbf{x},y)}\left[\log Q_\theta(y|\mathbf{x})\right] + \mathbb{E}_{P(\mathbf{x})}\left[D_{\mathrm{KL}}\left(P(y|\mathbf{x}) \| Q_\theta(y|\mathbf{x})\right)\right]. && (4.9)\\
& && (4.10)
\end{aligned}$$

In the above, the last equality is attributed to the Kullback-Leibler divergence. By retaining the terms that relate to $Q_\theta(y|\mathbf{x})$, we can obtain the objective function of MSMI,

$$\mathcal{L}_{\mathrm{MSMI}}(\theta) = \mathbb{E}_{P_I(\mathbf{x},y)}\left[\log Q_\theta(y|\mathbf{x})\right] + \beta \mathbb{E}_{P(\mathbf{x})}\left[\sum_{y=1}^{K} -P(y|\mathbf{x}) \log Q_\theta(y|\mathbf{x})\right], \tag{4.11}$$

where β is a hyper-parameter controlling the strength of restriction on the outputs of ID and OOD samples. The estimate of the first term requires the ground-truth labels of ID samples. The second term represents the form of the supervision information of both ID and OOD samples, i.e., the negative probabilities of all classes $[-P(1|\mathbf{x}), \ldots, -P(y|\mathbf{x}), \ldots, -P(K|\mathbf{x})]$. Accordingly, ID samples need two different kinds of supervision information (i.e., ground-truth labels and the negative probabili-

ties of all classes), and OOD samples treat the negative probabilities of all classes as adaptive supervision information. Therefore, the supervision information also restricts the behavior of $Q_\theta(y|\mathbf{x})$ for ID samples for no adverse effects on their classification. The reason is that, if the data space only contains ID samples, i.e., $P_I(\mathbf{x}) = P(\mathbf{x})$, the objective function Eq. (4.11) degenerates into the confidence penalty [57], which has empirically been proved to improve the generalization ability.

4.2.3 MBCE: Multiple Binary Cross Entropy

By strengthening the association between ID samples and their labels, the supervision information $[-P(1|\mathbf{x}), \ldots, -P(y|\mathbf{x}), \ldots, -P(K|\mathbf{x})]$ adapts OOD samples to ID samples. This strategy makes the networks aware of OOD and minimizes the OOD interference in classifying ID samples. However, providing this supervision information of OOD samples in the training process is impracticable due to the unknown conditional distribution $P(y|\mathbf{x})$ of OOD samples. Accordingly, we assume that OOD samples differ from ID ones in terms of being affiliated with respective classes and estimate the supervision information by exploring the data relationships between ID and OOD samples. The estimated supervision information makes ID samples associated with different labels more separable.

Note that the parametric discriminator $Q_\theta(y|\mathbf{x})$ is applied to estimate $P(y|\mathbf{x})$ for predicting a label to a given sample. Accordingly, $Q_\theta(y|\mathbf{x})$ can be reformulated to specify the adaptive supervision information for OOD samples. For instance, the traditional method applies the standard softmax function to $Q_\theta(y|\mathbf{x})$ in estimating $P_I(y|\mathbf{x})$. However, this method cannot estimate $P(y|\mathbf{x})$ because it ignores the data relationships between ID and OOD. It thus fails to make OOD samples adaptive to ID samples, which consequently misleads the learning process of correctly predicting labels for ID samples.

To further improve the ID classification accuracy, we separate different class of ID samples [71]. Specifically, we formulate $Q_\theta(y|\mathbf{x})$ to estimate the supervision information of OOD samples and approximate the conditional distribution of the MI in the mixed data space. Precisely, we form K binary regression problems where K is the number of classes where each binary regression problem is for the corresponding set of ID samples with the same label against all OOD samples. We then integrate all the K problems to obtain a compact expression of $Q_\theta(y|\mathbf{x})$.

To estimate the conditional distribution $P(y|\mathbf{x})$ by the parameterized discriminator $Q_\theta(y|\mathbf{x})$,

$$Q_\theta(y|\mathbf{x}) = \frac{P_I(\mathbf{x}, y)}{P(\mathbf{x})} = \frac{P_I(\mathbf{x}, y)}{(1-\epsilon)\sum_{y=1}^{K} P_I(\mathbf{x}, y) + \epsilon P_O(\mathbf{x})}. \tag{4.12}$$

However, it is infeasible to design $Q_\theta(y|\mathbf{x})$ to estimate $P(y|\mathbf{x})$ according to Eq. (4.12)

because estimating $P(\mathbf{x})$ requires the probability $P_I(\mathbf{x}, y)$ which is unknown for OOD samples. To avoid this problem, we firstly rewrite Eq. (4.12),

$$Q_\theta(y|\mathbf{x}) = \frac{P_I(\mathbf{x}, y)}{P_I(\mathbf{x}, y) + P(\mathbf{x})} \frac{P_I(\mathbf{x}, y) + P(\mathbf{x})}{P(\mathbf{x})}, \tag{4.13}$$

and estimate the following density ratio [72] by an auxiliary function $D(\mathbf{x}, y)$,

$$\frac{P_I(\mathbf{x}, y)}{P(\mathbf{x}) + P_I(\mathbf{x}, y)} = D(\mathbf{x}, y) = \sigma(\log \varphi(\mathbf{x}, y)) \tag{4.14}$$

where $\sigma(s) = 1/(1 + \exp(-s))$ is the sigma function and $\varphi(\mathbf{x}, y)$ is the normalized output by the softmax function

$$\varphi(\mathbf{x}, y) = \frac{\exp\left(f_\theta(\mathbf{x}, y)\right)}{\sum_{y=1}^{K} \exp\left(f_\theta(\mathbf{x}, y)\right)}, \tag{4.15}$$

where f_θ is a parametric neural network which maps each sample $\mathbf{x}$ to a K-dimensional output vector $(f_\theta(\mathbf{x}, 1), \ldots, f_\theta(\mathbf{x}, y), \ldots, f_\theta(\mathbf{x}, K))$, and each $f_\theta(\mathbf{x}, y)$ represents the classification score of the corresponding class.

Substituting Eq. (4.14) into Eq. (4.13), we have

$$\begin{aligned} Q_\theta(y|\mathbf{x}) =& \frac{D(\mathbf{x}, y)}{1 - D(\mathbf{x}, y)} = \exp\left(\log \frac{D(\mathbf{x}, y)}{1 - D(\mathbf{x}, y)}\right) \\ =& \exp\left(\sigma^{-1}(D(\mathbf{x}, y))\right) = \exp\left(\sigma^{-1}(\sigma(\log \varphi(\mathbf{x}, y)))\right) \\ =& \exp\left(\log \varphi(\mathbf{x}, y)\right) = \varphi(\mathbf{x}, y). \end{aligned} \tag{4.16}$$

The third equality of Eq. (4.16) is due to the property about the sigma function $\sigma^{-1}(s) = \log(s/(1 - s))$. We thus end with a simple and elegant expression $Q_\theta(y|\mathbf{x}) = \varphi(\mathbf{x}, y)$. Although this expression is the same as that in the traditional method, its meaning is different. This is because it is an intermediate result in solving the auxiliary function $D(\mathbf{x}, y)$ in Eq. (4.14) rather than the target in the traditional objective function $\max \mathbb{E}_{P_I(\mathbf{x},y)} [\log \varphi(\mathbf{x}, y)]$. Specifically, solving $D(\mathbf{x}, y)$ to estimate $P(y|\mathbf{x})$ causes a specific $Q_\theta(y|\mathbf{x})$, and this estimation process is a component of optimizing the objective function Eq. (4.11), which trains $Q_\theta(y|\mathbf{x})$ to approximate $P(y|\mathbf{x})$.

However, deciding the expression for $Q_\theta(y|\mathbf{x})$ leads to the question of how to address $D(\mathbf{x}, y)$. We know that the density ratio $D(\mathbf{x}, y)$ is the probability that a given sample $\mathbf{x}$ is sampled from $P_I(\mathbf{x}, y)$ in the mixture distribution $P(\mathbf{x})$. Accordingly, the problem of estimating this density ratio is transformed into a binary classification problem [73]. For each binary classifier, the samples from the joint distribution $P_I(\mathbf{x}, y)$ are positive while those from the mixture distribution $P(\mathbf{x})$ are negative. For calculating the loss for negative samples, we require to access the label distribution of the

mixture distribution $P(\mathbf{x})$ due to the density ratio $D(\mathbf{x}, y)$. However, the label distribution of OOD samples in the mixed data space is unknown. Therefore, we assume the label distribution $P(y)$ of the mixture distribution $P(\mathbf{x})$ is consistent with $P_I(y)$. This assumption avoids the disturbance of OOD samples in classifying ID samples by misleading the label distribution $P_I(y)$. According to the logistic regression [74], the density ratio $D(\mathbf{x}, y)$ is trained by maximizing the following objective function MBCE,

$$\mathcal{L}_{\text{MBCE}}(\theta) = \mathbb{E}_{P_I(\mathbf{x},y)} \left[\log D(\mathbf{x}, y)\right] - \mathbb{E}_{P_I(y)} \mathbb{E}_{P(\mathbf{x})} \left[\log \left(1 - D(\mathbf{x}, y)\right)\right]. \tag{4.17}$$

To understand this objective function, we rewrite the first term according to the product rule of probability and have,

$$\mathbb{E}_{P_I(y)} \left[\mathbb{E}_{P_I(\mathbf{x}|y)} \left[\log D(\mathbf{x}, y)\right] - \mathbb{E}_{P(x)} \left[\log \left(1 - D(\mathbf{x}, y)\right)\right]\right].$$

Accordingly, the objective function integrates multiple binary regression problems where each binary cross-entropy loss is applied to a corresponding problem. The number of binary cross-entropy losses is equal to the number of ID labels. Each binary cross-entropy loss is applied to distinguish ID samples from the conditional distribution $P_I(\mathbf{x}|y)$ and the samples from the mixture distribution $P(\mathbf{x})$. The OOD samples shared among different binary regression problems can be treated as a bridge to make ID samples with different labels more separable [69], leading to an improved ID classification ability. Note that these binary regression problems have the same parametric neural network f_θ. Therefore, we can optimize their binary cross-entropy losses in the integrated loss Eq. (4.17). We obtain the estimated supervision information $[-P(1|\mathbf{x}), \ldots, -P(y|\mathbf{x}), \ldots, -P(K|\mathbf{x})]$ for OOD samples and the expression of $Q_\theta(y|\mathbf{x})$ by solving the objective function. We can obtain a discriminator $Q_\theta(y|\mathbf{x})$ by optimizing $D(\mathbf{x}, y)$ in Eq. (4.17). However, the discriminator should actually be learned from Eq. (4.11) for maximizing the mutual information between ID samples and their labels in the mixed space. This is because Eq. (4.17) is a constraint involving the data relationships between ID and OOD on $Q_\theta(y|\mathbf{x})$ in Eq. (4.11).

4.2.4 The SA Algorithm

The learning process of the proposed SA method is summarized in Algorithm 3. SA takes advantage of MSMI and MBCE to achieve a balance between ID classification and OOD detection. The objective function MSMI reveals the form of required OOD supervision information, which aims to make networks aware of OOD and minimize their interference in classifying ID samples. Furthermore, the objective function MBCE presents the method to estimate the supervision information and expression of the parametric discriminator in MSMI to improve the generalization. Therefore, MSMI and

Algorithm 3 Supervision Adaptation (SA)

1: **repeat**
2: Sample $\{(\mathbf{x}_1^I, y_1^I), \ldots, (\mathbf{x}_M^I, y_M^I)\}$ from $P_I(\mathbf{x}, y)$
3: Sample $\{(\mathbf{x}_1^O), \ldots, (\mathbf{x}_N^O)\}$ from $P_O(\mathbf{x})$
4: Estimate objective function:

$$\widetilde{\mathcal{L}}_{SA}(\theta) = \frac{1}{M}\sum_{m=1}^{M}\log\varphi(\mathbf{x}_m^I, y_m^I) + \frac{\alpha}{N}\sum_{n=1}^{N}\left[\sum_{y=1}^{K}\left(P_I(y) - \varphi(\mathbf{x}_n^O, y)\right)\log\varphi(\mathbf{x}_n^O, y)\right]$$

5: Obtain gradients $\nabla_\theta\widetilde{\mathcal{L}}_{SA}(\theta)$ to update parameters θ
6: **until** convergence
7: **Output:** discriminator $Q_\theta(y|\mathbf{x})$

MBCE are complementary. To take advantage of MSMI and MBCE, we combine these two components to obtain the objective function for SA.

To apply the adaptive supervision information $[-P(1|\mathbf{x}), \ldots, -P(y|\mathbf{x}), \ldots, -P(K|\mathbf{x})]$ of OOD samples to train a discriminator $Q_\theta(y|\mathbf{x})$ and design the specific $Q_\theta(y|\mathbf{x})$ by estimating the supervision information, we linearly combine MSMI and MBCE because MBCE can be regarded as a constraint on $Q_\theta(y|\mathbf{x})$ in MSMI,

$$(1-\alpha)\mathcal{L}_{\text{MI}}(\theta) + \alpha\mathcal{L}_{\text{MBCE}}(\theta) \tag{4.18}$$

where $\alpha \in [0, 1]$ is a combination parameter to balance the impact of the two components. Combining Eqs. (4.13), (4.16) and (4.18), we have

$$\begin{aligned}&\underbrace{(1-\alpha)\mathbb{E}_{P_I(\mathbf{x},y)}\left[\log\varphi(\mathbf{x}, y)\right]}_{:=A}\underbrace{-(1-\alpha)\beta\mathbb{E}_{P(\mathbf{x})}\left[\sum_y \varphi(\mathbf{x}, y)\log\varphi(\mathbf{x}, y)\right]}_{:=B}\\ &\underbrace{+\alpha\mathbb{E}_{P_I(\mathbf{x},y)}\left[\log\sigma(\log\varphi(\mathbf{x}, y))\right]}_{:=C}\underbrace{-\alpha\mathbb{E}_{P(\mathbf{x})}\mathbb{E}_{P_I(y)}\left[\log\left(1-\sigma(\log\varphi(\mathbf{x}, y))\right)\right]}_{:=D}.\end{aligned} \tag{4.19}$$

However, the complex objective function Eq. (4.19) causes the high cost of calculating the gradients for optimization. To simplify the above function, we first obtain the following lower bound,

$$\begin{aligned}A + C =&(1-\alpha)\mathbb{E}_{P_I(\mathbf{x},y)}\left[\log\varphi(\mathbf{x}, y)\right] + \alpha\mathbb{E}_{P_I(\mathbf{x},y)}\left[\log\frac{\varphi(\mathbf{x}, y)}{\varphi(\mathbf{x}, y)+1}\right]\\ =&\mathbb{E}_{P_I(\mathbf{x},y)}\left[\log\varphi(\mathbf{x}, y)\right] - \alpha\mathbb{E}_{P_I(\mathbf{x},y)}\left[\log\left(\varphi(\mathbf{x}, y)+1\right)\right]\\ \geq&\mathbb{E}_{P_I(\mathbf{x},y)}\left[\log\varphi(\mathbf{x}, y)\right] - \alpha\log 2,\end{aligned} \tag{4.20}$$

where the first equality is due to the property of the sigma function

$$\sigma(\log s) = \frac{s}{s+1}, s \geq 0 \tag{4.21}$$

and the first inequality holds since $\varphi(\mathbf{x}, y) \in (0, 1]$. Also, we have the following lower bound,

$$\begin{aligned} B + D \geq & -(1-\alpha)\beta \mathbb{E}_{P(\mathbf{x})} \left[\sum_y \varphi(\mathbf{x}, y) \log \varphi(\mathbf{x}, y) \right] \\ & + \alpha \mathbb{E}_{P(\mathbf{x})} \mathbb{E}_{P_I(y)} \left[\log \left(\varphi(\mathbf{x}, y) + 1 \right) \right] \\ \geq & \mathbb{E}_{P(\mathbf{x})} \left[\sum_y \left(\alpha P_I(y) - (1-\alpha)\beta \varphi(\mathbf{x}, y) \right) \log \varphi(\mathbf{x}, y) \right], \end{aligned} \tag{4.22}$$

where the first inequality holds owing to Eq. (4.21) and the second inequality holds since $\log(x)$ is a monotonically increasing function. To obtain a compact result, we assume $\beta = \alpha/(1-\alpha)$ without loss of generality. Substituting Eq. (4.20) and Eq. (4.22) into Eq. (4.19), we obtain the objective function of the SA method,

$$\mathcal{L}_{SA}(\theta) = \mathbb{E}_{P_I(\mathbf{x},y)} \left[\log \varphi(\mathbf{x}, y) \right] + \alpha \mathbb{E}_{P(\mathbf{x})} \left[\sum_{y=1}^{K} \left(P_I(y) - \varphi(\mathbf{x}, y) \right) \log \varphi(\mathbf{x}, y) \right], \tag{4.23}$$

where $P_I(y)$ is an ID class probability which can be estimated by exploring ID samples in experiments. From the derived result Eq. (4.23), we observe that the adaptive supervision information for a given OOD sample with respect to class y is $P_I(y) - \varphi(\mathbf{x}, y)$ after combining MSMI with MBCE.

Recall that $P(\mathbf{x}) = (1-\epsilon)P_I(\mathbf{x}) + \epsilon P_O(\mathbf{x})$ is the mixture distribution of ID and OOD samples, accordingly, to apply the stochastic gradient descent (SGD) optimization algorithm to estimate the gradient of the objective function Eq. (4.23), we assume the mini-batch size for mixed data is B which includes $M = (1-\epsilon)B$ ID samples $\{(\mathbf{x}_1^I, y_1^I), \ldots, (\mathbf{x}_N^I, y_N^I)\}$ and $N = \epsilon B$ OOD samples $\{(\mathbf{x}_1^O), \ldots, (\mathbf{x}_M^O)\}$.

4.3 Experiments

We demonstrate the effectiveness of the proposed SA method using four network architectures, two ID datasets and eleven OOD datasets. Specifically, we compare the SA method with several state-of-the-art methods, analyze the effect of parameters, run a set of ablation study experiments, and make a qualitative analysis.

Table 4.1 SA: OOD detection performance.

Each value indicates the average AUROC score on the eleven OOD datasets, and the boldface values indicate the relatively better OOD detection performance.

ID dataset / Measure	Network	Baseline	EC / UF / PN / OE / SA		
			TinyImageNet	LSUN	CelebA
CIFAR10 AUROC	ResNet18	91.4	96.4 / 96.4 / 96.5 / 95.0 / **98.1**	96.9 / 96.8 / 96.8 / 96.2 / **98.9**	94.2 / 95.1 / 91.1 / 91.0 / **96.1**
	VGG19	89.3	**95.4** / 95.3 / 95.4 / 92.9 / 92.4	**96.6** / 96.5 / 96.5 / 94.9 / 96.4	93.8 / 92.5 / **95.8** / 89.2 / 90.7
	MobileNetV2	87.9	95.3 / **95.8** / 94.8 / 93.9 / 94.6	96.2 / 96.3 / 96.2 / 95.9 / **97.6**	86.4 / 92.7 / 91.2 / 90.6 / **92.8**
	EfficientNet	91.3	94.9 / 94.7 / 94.7 / 92.5 / **95.2**	96.0 / 96.6 / 95.9 / 95.1 / **97.1**	91.6 / 93.6 / 91.2 / 90.0 / **94.5**
CIFAR100 AUROC	ResNet18	80.1	91.5 / 92.3 / 92.9 / 89.0 / **93.0**	93.6 / 93.7 / 94.0 / 93.5 / **96.4**	86.3 / 88.6 / 88.8 / 87.3 / **92.1**
	VGG19	72.0	87.8 / 89.5 / **89.7** / 86.7 / 89.4	91.6 / 92.2 / 94.0 / 92.2 / **94.4**	85.2 / 86.1 / 86.6 / 84.0 / **86.9**
	MobileNetV2	72.6	87.0 / 90.4 / **90.7** / 88.3 / 87.7	92.3 / 91.7 / 93.1 / **93.7** / 92.8	81.9 / 79.4 / **85.2** / 83.9 / 83.3
	EfficientNet	74.2	86.8 / 89.8 / **91.1** / 87.2 / 89.9	90.3 / 92.9 / 92.9 / 92.7 / **94.1**	84.8 / 81.4 / 85.0 / 83.5 / **86.5**

Table 4.2 SA: Classification accuracy (compared with retraining methods).

Each value indicates the ACC score on the corresponding test ID dataset, and boldface values indicate the relatively better classification performance.

ID dataset / Measure	Network	Baseline	EC / UF / PN / OE / SA		
			TinyImageNet	LSUN	CelebA
CIFAR10 ACC	ResNet18	95.0	94.7 / 95.0 / 94.9 / 94.9 / **95.1**	95.0 / 94.7 / 95.0 / 94.9 / **95.1**	95.2 / 95.2 / 95.0 / 94.9 / **95.3**
	VGG19	93.5	93.3 / **93.8** / 93.4 / 92.7 / 93.6	93.5 / 93.5 / 93.4 / 93.2 / **93.7**	93.1 / 93.2 / 93.3 / 93.5 / **93.5**
	MobileNetV2	91.3	91.0 / 91.5 / 91.2 / 90.7 / **91.7**	90.8 / 91.5 / 91.2 / 91.1 / **91.6**	91.0 / 91.6 / 91.0 / 91.3 / **91.7**
	EfficientNet	90.6	89.8 / 90.1 / 90.1 / **90.2** / 90.1	90.2 / 90.5 / 90.2 / 89.6 / **90.7**	90.1 / **90.5** / 90.1 / 89.9 / 90.3
CIFAR100 ACC	ResNet18	77.7	77.7 / 77.8 / 77.3 / 77.6 / **78.6**	78.0 / 77.5 / 77.1 / 76.5 / **78.6**	77.3 / 77.0 / 77.4 / 77.3 / **79.0**
	VGG19	71.4	71.5 / 71.7 / 71.5 / 72.0 / **73.0**	71.3 / 71.6 / 72.1 / 71.5 / **72.7**	71.5 / 71.7 / 71.2 / 71.5 / **72.3**
	MobileNetV2	71.2	70.1 / 70.4 / 70.5 / 70.8 / **71.4**	70.8 / 70.9 / 70.3 / 70.6 / **72.2**	69.6 / 70.0 / 70.7 / 69.9 / **71.1**
	EfficientNet	69.1	68.8 / 68.9 / 69.1 / 68.7 / **70.8**	68.8 / 69.0 / 68.7 / 69.2 / **69.6**	69.2 / 68.0 / 68.1 / 68.4 / **70.3**

4.3.1 Setup

We introduce the network architectures incorporated into different OOD detection methods, the ID and OOD datasets for experiments, and the evaluation metrics.

4.3.1.1 Network Architectures and Datasets

To verify the general applicability of the proposed SA method for different neural architectures, we apply it to four advanced convolutional neural networks ResNet18 [1], VGG19 [66], MobileNetV2 [75], and EfficientNet [76]. All networks are implemented in PyTorch trained in a single GPU. The setups of all networks follow the setups used in FIG method.

Networks are trained to classify two ID datasets CIFAR10 and CIFAR100 [60]. The OOD datasets include SVHN [59], iSUN [77], LSUN [62], TinyImageNet [63], CelebA [78], VisDA [79], Caltech256 [64], PASC [80], COCO [65], Gaussian, and Uniform. For an OOD dataset used in the training phase of the SA method, according to the definition of mixture distribution $P(\mathbf{x})$, the number of OOD samples required in the training process is #training ID samples $\times$ ϵ, and the rest OOD samples are used to test the OOD detection performance. For an OOD dataset unused for training, the whole dataset is applied to test the OOD detection performance.

4.3.2 Comparison Results

We compare the proposed SA methods with methods of improving OOD sensitivity and methods of improving generalization. We aim to verify that the SA method can achieve

a competitive ID classification accuracy while also pursuing a high OOD detection performance.

4.3.2.1 Comparison Methods of Improving OOD Sensitivity

We compare the performance of the baseline, extra class (EC) [15], KL with the uniform distribution (UD) [37], prior network (PN) [16], outlier exposure (OE) [35], and our proposed SA method in terms of measures ACC and AUROC. For a trained model, we calculate ACC on the ID test dataset corresponding to its ID training dataset. Furthermore, by adopting maximum softmax output as OOD scores, we report the average AUROC across eleven OOD datasets, namely, SVHN, iSUN, LSUN, TinyImageNet, CelebA, Caltech256, VisDA, PASC, COCO, Gaussian, and Uniform.

The baseline method applies a pretrained network which is optimized from a cross-entropy loss on a training ID dataset. The rest methods retrain the network on a pair of an ID dataset and an OOD dataset. For the EC method, we add an extra class to the OOD dataset. For UD [37], PN [16] and OE [35], we apply the setups suggested in their corresponding papers, respectively. Specifically, for UD, we select the regularization coefficient for the KL divergence with the uniform distribution among $\{0.1, 1\}$. The best OOD detection performance and the corresponding classification accuracy are reported. For PN, we set 1 for the hyper-parameter of the dense Dirichlet distribution for OOD samples. For OE, we set the regularization coefficient 0.5 to a margin ranking loss on the log probabilities of ID and OOD samples. For SA, we set $\alpha = 0.2$ which achieves a trade-off between the high OOD detection performance and the high classification accuracy. For fair comparisons, the component parameter ϵ of OOD samples is 0.05 for all considered models.

The comparison results of the OOD detection performance are summarized in Table 4.1. We find that the methods introducing an OOD dataset to training processes can significantly improve the OOD detection performance over the baseline method. SA achieves the best results in most cases and competitive performance in the remaining situations. The average rank of SA 1.91 across all the neural architecture and OOD dataset pairs. The reason is that the OOD datasets restrict networks to make certain and low-probability predictions for samples differing from ID samples. Specifically, the adaptive supervision information of OOD samples in SA can sufficiently describe OOD samples, which causes that more valuable information behind an introduced OOD dataset can be utilized to recognize other unobserved OOD samples.

The comparison results of the classification accuracy are summarized in Table 4.2. Compared with the baseline method, all the other methods except SA can improve the OOD detection performance by sacrificing classification accuracy. SA method performs even better on classifying ID samples with its average rank 1.12 across all the neural

Table 4.3 SA: Classification accuracy (compared with generalization improvement methods).

The SA method is trained with the TinyImageNet OOD dataset, and the other methods focusing on improving the generalization performance are trained without any OOD datasets. The boldface values represent the relatively better classification performance.

Network	Baseline / GS / MIXUP / LS / Tf-KD / SA	
Measure	CIFAR10	CIFAR100
ResNet18 ACC	95.0 / 94.7 / **95.8** / 95.1 / 95.2 / 95.1	77.7 / 76.5 / 78.2 / 78.8 / **79.0** / 78.6
VGG19 ACC	93.5 / 92.9 / **94.3** / 93.3 / 93.2 / 93.7	71.4 / 70.4 / 72.8 / **72.8** / 72.4 / 72.7
MobileNetV2 ACC	91.3 / 90.8 / 91.3 / 91.2 / 91.0 / **91.6**	71.2 / 69.1 / 68.6 / 71.0 / 71.6 / **72.2**
EfficientNet ACC	90.6 / 89.9 / 88.9 / 90.6 / 89.3 / **90.7**	69.1 / 67.7 / 66.2 / 69.0 / 69.1 / **69.6**

architecture and OOD dataset pairs. This is because that the manually-determined labels for OOD samples mislead the label distribution to corrupt the training data and make networks less attentive to classification tasks. Furthermore, SA applies the supervision information of OOD samples to improve the generalization capacity. Specifically, the supervision information minimizes the OOD interference in classifying ID samples and makes ID classes more distinguishable. Compared with the state-of-the-art methods involving OOD datasets in the training process, SA achieves a balance between the generalization capacity for ID samples and the detection performance for OOD samples.

4.3.2.2 Comparison Methods of Improving Generalization

To demonstrate the significant classification improvement by involving OOD samples in SA, we further compare SA with other methods without involving extra OOD datasets. Recall that the SA learning with OOD samples is for increasing the OOD sensitivity of discriminators. The improvement of SA on classification accuracy is a by-product. The comparison methods include the baseline, Gaussian noise (GN), MIXUP [13], label smoothing (LS) [57], and the teacher-free knowledge distillation (Tf-KD) [81]. For all the methods, we follow the setups suggested in their corresponding papers, respectively.

The experimental results are presented in Table 4.3 and Table 4.4. The two tables show that SA obtains average 8.52% and $26,76\%$ improvements across all the neural architecture and OOD dataset pairs over the baseline method on the datasets CIFAR10 and CIFAR100, respectively, which indicates that the involved OOD datasets significantly improve the network ability to distinguish ID and OOD samples. Other methods mainly focusing on improving generalization are less capable of detecting OOD sam-

Table 4.4 SA: OOD detection performance (compared with generalization improvement methods).

The SA method is trained with the TinyImageNet OOD dataset, and the other methods focusing on improving the generalization performance are trained without any OOD datasets. The boldface values represent the relatively better OOD detection performance.

Network / Measure	Test OOD	Baseline / GS / MIXUP / LS / Tf-KD / SA	
		CIFAR10	CIFAR100
ResNet18 AUROC	SVHN	92.0 / 89.4 / 89.6 / 88.8 / 83.4 / **97.1**	83.2 / 83.7 / 76.6 / 78.6 / 80.0 / **90.8**
	iSUN	91.8 / 92.1 / 91.6 / 86.0 / 79.7 / **100.0**	84.8 / 81.3 / 73.4 / 80.2 / 82.9 / **99.9**
	LSUN	94.1 / 92.9 / 95.5 / 93.6 / 93.8 / **98.4**	82.4 / 81.6 / 77.6 / 81.7 / 80.3 / **93.3**
	TinyImageNet	93.2 / 92.5 / 93.8 / 90.7 / 89.5 / **97.1**	84.9 / 84.2 / 83.7 / 85.7 / 85.5 / **93.2**
	CelebA	86.9 / 82.7 / 85.5 / 79.5 / 82.3 / **100.0**	84.8 / 85.2 / 79.2 / 83.3 / 83.3 / **99.9**
	Caltech256	92.1 / 90.8 / 90.3 / 86.5 / 87.2 / **99.3**	82.9 / 82.2 / 79.0 / 82.9 / 82.6 / **98.1**
	VisDA	91.9 / 89.5 / 89.8 / 84.0 / 88.6 / **99.8**	80.2 / 81.5 / 76.7 / 78.4 / 77.7 / **99.6**
	PASC	91.2 / 88.8 / 86.1 / 85.7 / 87.8 / **98.4**	81.1 / 82.0 / 78.4 / 83.7 / 81.6 / **93.3**
	COCO	91.0 / 90.6 / 88.0 / 85.7 / 87.0 / **97.4**	82.9 / 81.6 / 80.5 / 83.1 / 82.1 / **92.2**
	Gaussian	88.8 / 92.5 / 99.9 / 83.4 / 93.5 / **100.0**	57.1 / 70.8 / 98.2 / 85.8 / 66.0 / **99.9**
	Uniform	92.9 / 93.9 / 99.5 / 87.4 / 94.5 / **100.0**	76.5 / 82.3 / 98.7 / 89.3 / 77.6 / **100.0**
VGG19 AUROC	SVHN	85.8 / 78.7 / **91.9** / 75.1 / 72.9 / 90.5	72.4 / 76.2 / 80.2 / 73.3 / 73.0 / **84.1**
	iSUN	90.4 / 87.9 / 92.1 / 79.9 / 81.1 / **99.9**	59.4 / 68.4 / 74.5 / 68.0 / 64.0 / **99.6**
	LSUN	93.3 / 91.4 / 95.2 / 83.2 / 85.7 / **97.8**	75.9 / 77.6 / 80.9 / 77.0 / 77.0 / **92.8**
	TinyImageNet	90.8 / 88.4 / **93.0** / 77.3 / 76.2 / 92.5	77.0 / 78.0 / 81.8 / 76.5 / 74.9 / **89.1**
	CelebA	82.7 / 73.4 / 87.7 / 67.6 / 67.2 / **99.4**	76.4 / 79.1 / 82.1 / 74.1 / 76.5 / **98.2**
	Caltech256	88.2 / 85.1 / 91.1 / 75.7 / 75.9 / **96.4**	75.5 / 76.4 / 79.9 / 73.9 / 74.0 / **95.7**
	VisDA	87.6 / 82.1 / 90.8 / 73.2 / 76.5 / **97.4**	75.4 / 76.5 / 79.9 / 74.3 / 75.8 / **96.5**
	PASC	86.2 / 81.9 / 89.4 / 75.6 / 67.5 / **94.6**	71.4 / 77.8 / 79.2 / 66.6 / 72.3 / **91.9**
	COCO	88.2 / 84.6 / **92.2** / 73.4 / 73.5 / 92.1	76.6 / 76.7 / 79.3 / 73.6 / 74.0 / **90.8**
	Gaussian	95.0 / 92.1 / 99.1 / 84.6 / 92.9 / **100.0**	70.0 / 63.2 / 73.0 / 85.9 / 79.4 / **100.0**
	Uniform	94.6 / 91.7 / 99.5 / 78.6 / 86.3 / **100.0**	61.7 / 64.1 / 62.5 / 90.7 / 88.8 / **100.0**
MobileNetV2 AUROC	SVHN	86.1 / 84.2 / 89.6 / 85.8 / 87.3 / **96.4**	72.2 / 74.9 / 74.9 / 77.4 / 81.5 / **87.8**
	iSUN	89.7 / 90.7 / 90.4 / 88.9 / 89.1 / **99.6**	71.8 / 65.8 / 62.1 / 75.6 / 75.2 / **98.2**
	LSUN	91.6 / 89.2 / 89.7 / 88.4 / 88.3 / **97.8**	76.3 / 72.8 / 73.9 / 76.3 / 79.5 / **86.1**
	TinyImageNet	90.5 / 89.6 / 90.3 / 84.5 / 85.6 / **94.4**	80.8 / 78.3 / 81.5 / 81.9 / 79.6 / **86.7**
	CelebA	88.9 / 84.5 / 85.8 / 83.6 / 83.6 / **99.4**	77.7 / 81.2 / 78.7 / 80.1 / 80.8 / **98.5**
	Caltech256	89.9 / 88.0 / 86.4 / 86.0 / 86.3 / **97.5**	75.4 / 75.8 / 74.6 / 77.2 / 76.9 / **94.9**
	VisDA	90.9 / 86.5 / 85.9 / 86.5 / 85.8 / **98.4**	76.0 / 80.1 / 74.7 / 75.2 / 77.3 / **96.8**
	PASC	88.4 / 84.6 / 88.3 / 83.7 / 84.8 / **97.6**	71.4 / 75.2 / 73.4 / 73.2 / 75.8 / **88.4**
	COCO	89.8 / 88.2 / 86.9 / 85.4 / 84.0 / **96.4**	77.3 / 76.0 / 75.9 / 77.9 / 77.7 / **90.1**
	Gaussian	76.2 / 82.3 / 90.7 / 89.1 / 90.8 / **97.3**	56.2 / 55.7 / 89.4 / 85.4 / 20.7 / **99.1**
	Uniform	84.6 / 86.0 / 96.2 / 87.4 / 85.7 / **99.2**	63.2 / 63.7 / 93.9 / 84.4 / 32.8 / **93.9**
EfficientNet AUROC	SVHN	90.7 / 88.4 / 83.2 / 87.5 / 87.3 / **94.8**	71.8 / 78.1 / 73.1 / 79.3 / 79.5 / **88.8**
	iSUN	92.3 / 89.4 / 90.4 / 88.6 / 84.9 / **99.4**	72.6 / 71.7 / 82.3 / 75.4 / 73.6 / **99.3**
	LSUN	89.9 / 88.4 / 86.1 / 84.2 / 83.3 / **96.7**	74.4 / 72.8 / 74.1 / 75.9 / 77.4 / **87.7**
	TinyImageNet	89.9 / 88.9 / 88.6 / 83.7 / 84.4 / **93.2**	76.4 / 75.6 / 77.9 / 76.7 / 75.9 / **89.5**
	CelebA	87.6 / 85.2 / 80.5 / 77.5 / 78.3 / **99.1**	78.0 / 80.7 / 79.1 / 79.3 / 80.2 / **99.0**
	Caltech256	89.0 / 86.7 / 83.9 / 84.0 / 83.9 / **97.5**	76.7 / 77.2 / 75.3 / 77.4 / 77.0 / **96.0**
	VisDA	89.3 / 87.1 / 82.7 / 83.4 / 83.0 / **97.4**	79.7 / 79.4 / 75.2 / 77.9 / 77.3 / **96.2**
	PASC	89.0 / 86.4 / 86.2 / 83.3 / 82.4 / **97.9**	72.5 / 75.9 / 76.1 / 76.4 / 77.1 / **94.5**
	COCO	89.9 / 87.7 / 86.1 / 83.1 / 82.9 / **97.4**	76.7 / 76.6 / 73.8 / 78.2 / 77.5 / **92.5**
	Gaussian	97.9 / 89.3 / 96.1 / **98.4** / 69.4 / 97.6	80.8 / 85.0 / 77.4 / 92.8 / 89.6 / **97.0**
	Uniform	**99.0** / 88.8 / 95.7 / 98.5 / 83.0 / 96.8	82.3 / 84.4 / 59.4 / 90.8 / 87.7 / **94.3**

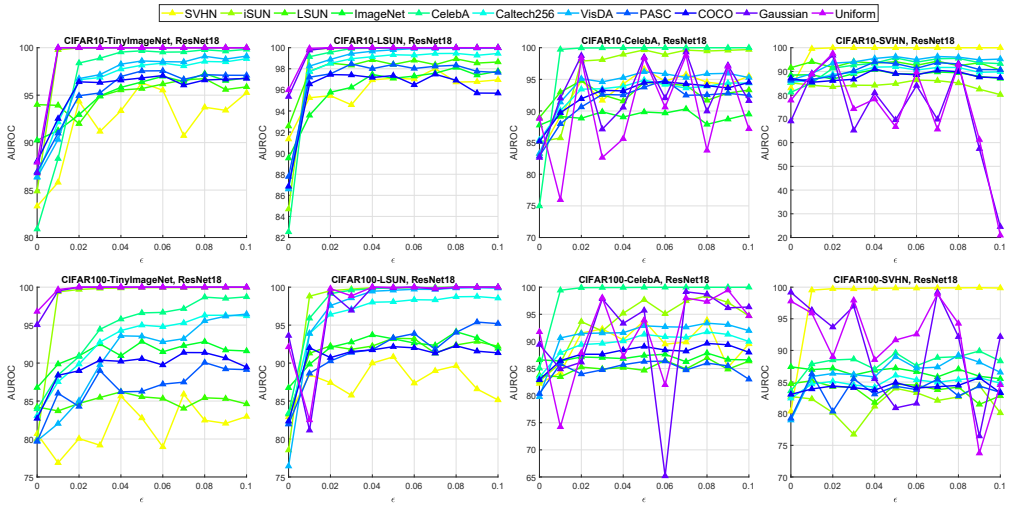


Fig. 4.1 SA: The Effect of Component Parameter ϵ ($\alpha = 0.2$).
Each title includes the corresponding training ID dataset, training OOD dataset, and network architecture. Eleven different OOD datasets are used to test the OOD detection performance, and each broken line corresponds to a test OOD dataset. Each point indicates an AUROC score on a test OOD dataset, and larger values are better.

ples than the baseline method. The main reason is that all methods lead to the decreased confidence prediction on ID samples without any constraints to OOD samples, which causes the boundary between the two kinds of samples is unclear.

Similar to the comparison methods, SA method also encourages ID samples to have smooth output probabilities. However, SA tends to assign lower confidence predictions to OOD samples by distinguishing ID and OOD samples according to the design principle of the component MBCE. In terms of classification accuracy, the SA method is competitive with the other compared methods and surpasses the baseline method by 0.22% and 1.26% on CIFAR10 and CIFAR100, respectively. The reasons for this phenomenon are two-fold (1) the adaptive nature of the supervision information guarantees that OOD samples will not disturb the learning process of classifying ID samples; (2) applying an OOD dataset to separate ID samples with different labels promotes the improvement further.

4.3.3 Effects of Parameters

We analyze the effects of the component parameter ϵ and the combination parameter α of SA in terms of ACC and AUROC.

4.3.3.1 Effect of the Component Parameter

We show the effect of the component parameter ϵ by selecting it from 11 evenly-spaced numbers starting from 0 and ending at 0.1 with $\alpha = 0.2$ in Fig. 4.1. A larger ϵ yields better OOD detection performance, and this effect diminishes when the component pa-

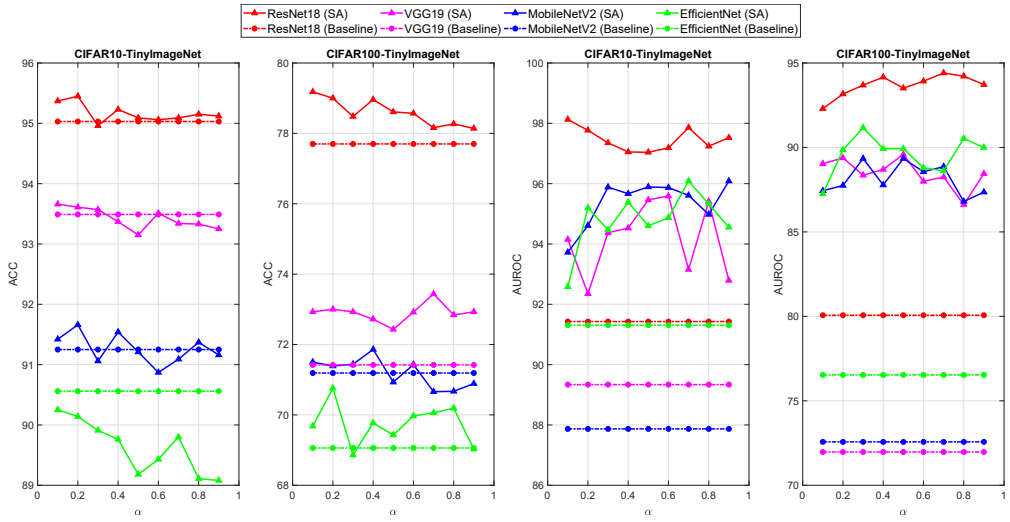


Fig. 4.2 SA: The Effect of Combination Parameter α ($\epsilon = 0.05$).
Each title contains the information about the training ID dataset and the network architecture. Solid lines and dotted lines correspond to the results of SA and the baseline method, respectively. Each point in the first two sub-figures indicates an ACC score on the test dataset corresponding to the training ID dataset, and each point in the last two sub-figures indicates an average AUROC score on the eleven OOD datasets.

rameter is too large. However, the detection performance is sensitive to ϵ as expected when the models are trained on OOD datasets like CelebA and SVHN. This is because simple samples without complex textures and shapes from the two datasets are quite different from unrecognizable samples. In contrast, we also observe that small ϵ (like 0.05) leads to a high OOD detection performance. Therefore, we conclude that training the SA method with a complex OOD dataset can better improve its sensitivity to different OOD samples. Furthermore, a simple OOD dataset used in the training process is sufficient to guarantee a high OOD detection performance.

4.3.3.2 Effect of the Combination Parameter

The effect of the combination parameter α is shown in Fig. 4.2. The combination parameter α is selected from nine evenly-spaced numbers starting from 0.1 and ending at 0.9 with $\epsilon = 0.05$. The experimental results indicate that $\alpha \in [0.1, 0.2]$ causes improved ACC over the baseline for almost all ID dataset and network architecture pairs. However, ACC cannot be improved on CIFAR10 with the network architecture EfficientNet. The main reason is that it is difficult to improve the generalization ability of lightweight models focusing on enhancing the test efficiency like EfficientNet on datasets with few classes like CIFAR10 [76]. Furthermore, SA significantly outperforms the baseline in terms of AUROC for any choice of α. This result verifies the rationality and feasibility of learning with OOD samples to improve the network sensitivity to OOD samples. We observe that SA achieves the best OOD detection performance when α is approximately

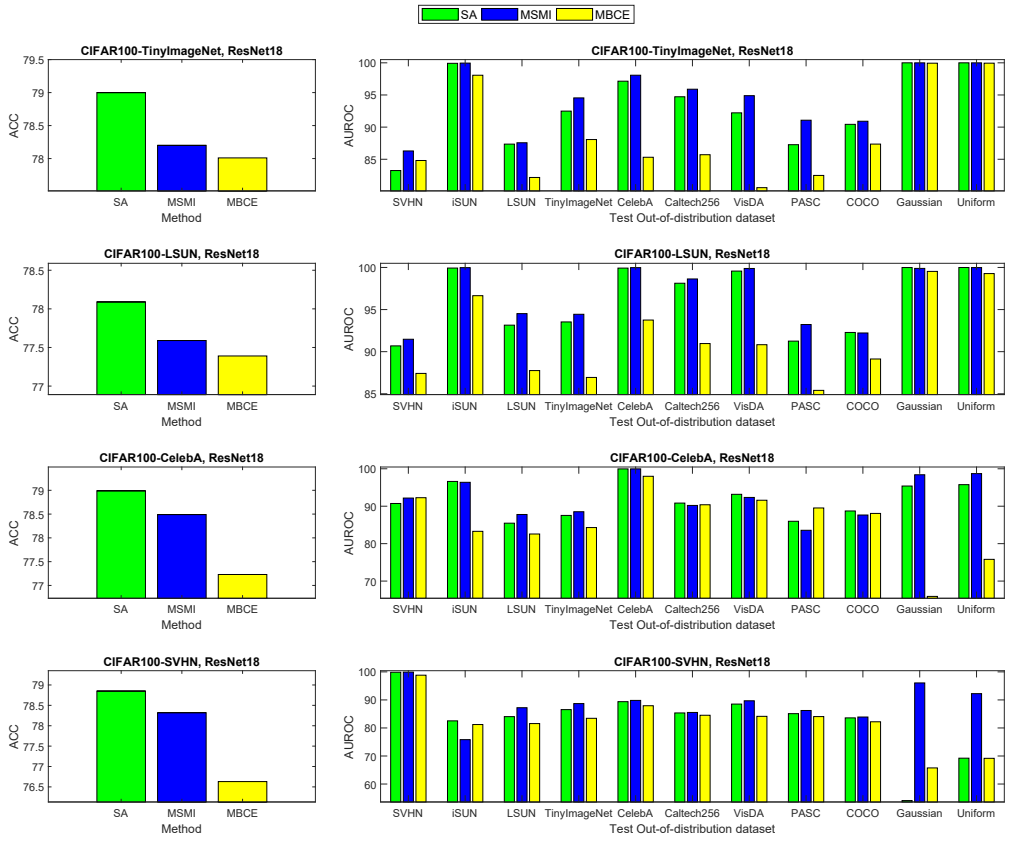


Fig. 4.3 SA: Results of the Ablation Study.
Each title contains the information about the training ID dataset and the network architecture. Each bar in the sub-figures on the left indicates the classification accuracy on the corresponding test ID dataset, and each bar in the sub-figures on the right indicates the detection performance for an OOD dataset. Higher bars are better.

equal to 0.5. However, a small α would be a better choice for a high OOD detection performance because it can also improve the classification accuracy over the baseline.

4.3.4 Ablation Study

SA method is based on two components: MSMI and MBCE. The two components reveal the form of the supervision information required for OOD samples and estimate this adaptive supervision information in training OOD-sensitive networks, respectively. We run a set of ablation study experiments to verify that both MSMI and MBCE are complementary and indispensable.

In the experimental setup, MSMI optimizes the objective function Eq. (4.11) where the adaptive supervision information $P(y|x)$ is directly approximated by the parametric discriminator $Q_\theta(y|\mathbf{x})$. MBCE optimizes the objective function Eq. (4.17) to learn $D(\mathbf{x}, y)$ and infers the discriminator $Q_\theta(y|\mathbf{x})$ according to Eq. (4.16). For a fair comparison, following the discussion in Section 4.3.3, we apply component parameter $\epsilon = 0.05$ for all methods and the combination parameter $\alpha = 0.2$ for SA method.

The ablation study experimental results are shown in Fig 4.3. Networks obtained by MBCE have poor classification accuracy. However, MSMI can achieve further improvement by combining with MBCE to derive the SA method. Therefore, considering the data relationships between ID and OOD samples by MBCE when estimating the supervision information in MSMI is necessary. Furthermore, MSMI achieves the best results in terms of AUROC, which verifies the effectiveness of the adaptive supervision information for OOD samples. The OOD detection capacity of MBCE is relatively low, which indicates separating ID and OOD samples without considering the data relationships between samples and labels is insufficient for learning OOD-sensitive networks. Taking advantage of both MSMI and MBCE, the SA method has a similar OOD detection capability to MSMI and a similar ID classification ability to MBCE. Therefore, we conclude that MSMI and MBCE form a complementary solution to balance the ID generalization capacity and the OOD detection ability.

4.3.5 Qualitative Analyses

To intuitively illustrate that the SA method can clearly distinguish ID and OOD samples, we consider the qualitative analyses of its prediction confidence trend, the final prediction probability, and the feature distribution.

The prediction confidence trends of different test datasets are presented in Fig. 4.4. For an ID test dataset containing correctly and wrongly classified samples, we evaluate the average prediction confidence in each iteration for each test dataset. By increasing the iteration times, the average prediction confidence of the SA correctly and wrongly classified samples increases gradually, and an opposite trend is observed on all test OOD datasets. However, the baseline increases the average prediction confidence on the test OOD datasets as the same as the test ID dataset. Hence, the increasing gap in the average prediction confidence between ID and OOD samples leads to that SA is easier to distinguish between the two kinds of samples.

We show the final prediction probability of individual samples in Fig. 4.5. We observe that both the baseline and SA provide high confidence on the correct class for an ID sample. However, SA is more sensitive to OOD samples than the baseline. This because all class confidence of the three OOD samples are suppressed in SA compared with that in the baseline.

We show the feature distribution in Fig. 4.6. We extract the output features of test samples from the baseline and SA and obtain the corresponding embeddings by t-SNE [9]. Recall that the number of classes of CIFAR10 is 10. For the baseline method, only 9 classes can be clearly observed, and the remaining ones has severely overlap with OOD samples. For SA, all 10 classes can be clearly observed. Accordingly, the SA method can better distinguish ID and OOD samples.

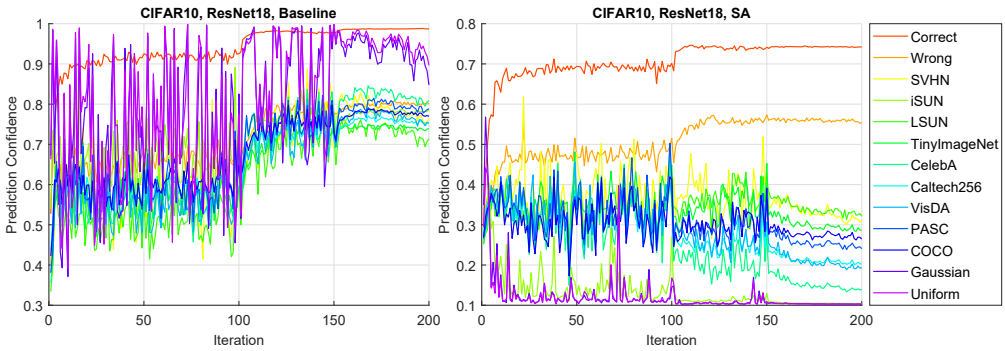


Fig. 4.4 SA: Prediction Confidence.
Each solid line represents the change of average prediction confidence of samples from the corresponding test set over time. In the legend, 'Correct' and 'Wrong' indicate correctly and wrongly classified samples, respectively, and the other names denote different OOD samples.

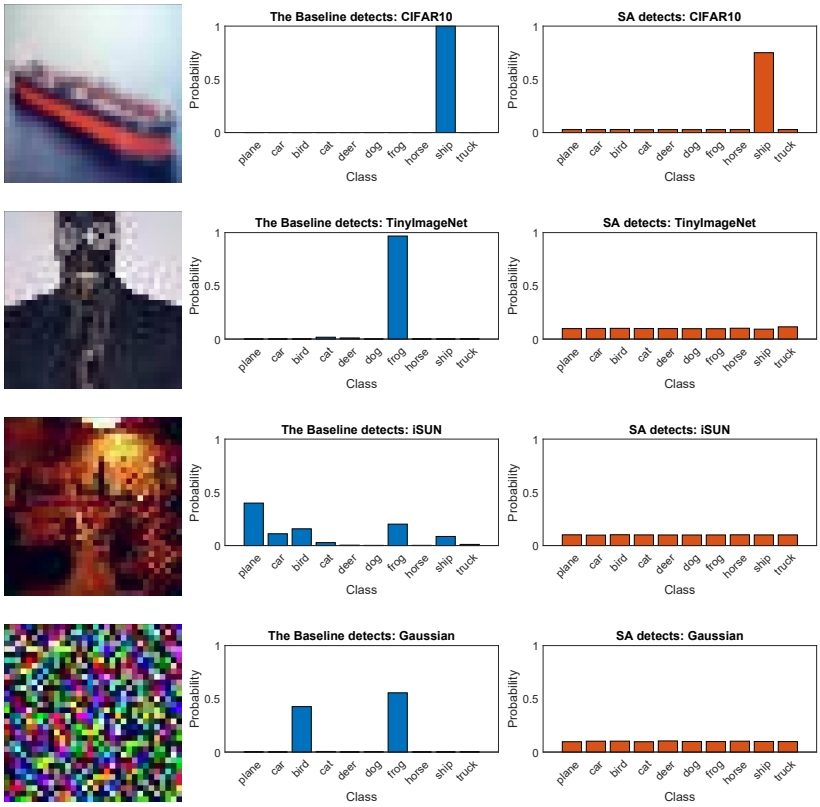


Fig. 4.5 SA: Inputs and Their Corresponding Prediction Probabilities.
The two models are trained by ResNet18 on CIFAR10, and the extra OOD Dataset used in SA is TinyImageNet. The blue and orange bars represent the result from the baseline and SA, respectively.

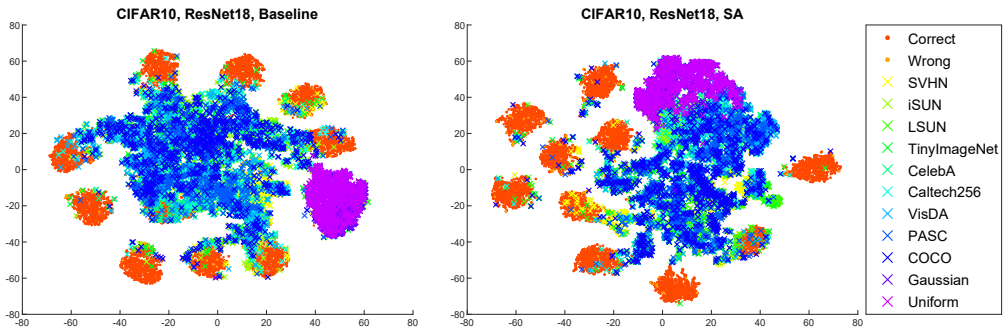


Fig. 4.6 SA: t-SNE Visualization of ResNet18 Features.
The color of points indicates the datasets of the corresponding samples. 'Correct' and 'Wrong' indicate correctly and wrongly classified samples, respectively, and the other names denote different OOD samples.

4.4 Summary of This Chapter

In this chapter, we propose a supervision adaptation (SA) approach to balancing the classification ability of ID samples and the detection capacity of OOD samples when OOD samples are available in the training process. SA solves the primary issue of defining the supervision information for OOD samples for adapting ID samples. We measure the data relationships between ID samples and their labels in terms of the mixed space mutual information to reveal the form of the supervision information. Furthermore, we consider the data correlations between the two kinds of samples to estimate this supervision information in terms of multiple binary regression problems. For different neural architectures and diverse datasets, we empirically demonstrate that SA consistently outperforms the baseline method in detecting OOD samples and achieves an improved classification performance.

CHAPTER 5

Out-of-distribution Detection via Cross-class Vicinity Distribution

5.1 Motivations

Chapter 4 discusses how to balance OOD detection and ID classification performance when OOD samples are involved in training phases by constructing adaptive supervision information. When OOD samples from real-world datasets are unavailable, generated OOD samples drawn from generative models can be involved, as discussed in Chapter 3. However, constructing generative models and sampling processes are usually expensive. Accordingly, we consider an orthogonal direction, i.e., improving OOD sensitivity by exploring the information from ID samples. Specifically, we explore generating augmented ID samples from vicinity distributions [82, 13] of training ID samples. The augmented ID samples can be treated as data-dependent OOD samples for the training ID dataset. However, data-dependent OOD samples cannot be drawn from the vicinity distributions constructed by the standard methods [83], including horizontal reflection, rotation and rescaling. These augmented samples can only be used as extra ID samples and mapped to the same label as their original one to improve generalization. The standard constructions usually assume that the samples from the same vicinity distribution own the same label. Furthermore, they ignore the vicinity relations across samples of different classes. Therefore, standard constructions cannot generate OOD samples. In this chapter, we break down this barrier to construct a vicinity distribution of an ID sample with involving the ID samples of other classes.

Accordingly, we propose a *Learning from Cross-class Vicinity Distribution* (LCVD) approach to explore OOD samples related to a given ID sample according to the following insight:

> *An OOD input generated by mixing multiple ID inputs does not belong to the same classes as its constituents.*

To draw an OOD sample drawn from the cross-class vicinity distribution of a given ID sample, we can linearly combine the ID sample with multiple ID samples associated with other classes. We assume the ID sample no longer belongs to the original class after being contaminated by those ID ones from other classes. The reason is that there

is more than one class in the contaminated input (the generated OOD input). Therefore, we encourage a network to make low-confidence predictions for samples located in the regions outside the training ID samples. Accordingly, the generated OOD input can be associated with a *complementary label*[1] [84] which could be any of the labels of its constituents. To improve the network capacity of discriminating between ID and OOD samples, we can finetune the pretrained network to reject the mapping between OOD inputs and the corresponding complementary labels.

5.2 Cross-class Vicinity Distribution

5.2.1 Generic Expected Risk

We introduce a generic expected risk [85] which is used to train a network from both ID and OOD samples. We define $\mathbf{x}$ and y as the input and label, respectively, and the number of classes is K. Each ID / OOD sample corresponds to a ground truth label / complementary label. The marginal distribution of $\mathbf{x}$, the marginal distribution of y, and the joint distribution are $P_I(\mathbf{x})$, $P_I(y)$ and $P_I(\mathbf{x}, y)$ for ID samples, respectively. Similarly, we define $P_O(\mathbf{x})$, $P_O(y)$ and $P_O(\mathbf{x}, y)$ for OOD samples. We assume both ID and OOD samples share the same conditional distribution $P(y|\mathbf{x})$ estimated by a parameterized network $Q_\theta(y|\mathbf{x})$ with model parameter θ. This is a mild assumption. This is because the two kinds of samples share the same label space and $P(y|\mathbf{x})$ depends on the given input x. Further, estimating $Q_\theta(y|\mathbf{x})$ for ID and OOD samples can recognize different classes of ID inputs and discriminate between ID and OOD samples. Recall that mutual information [48] is a quantity which measures the relationship between random variables. The mutual information for ID and OOD sample are defined as $\mathcal{I}_I(\mathbf{x}; y)$ and $\mathcal{I}_O(\mathbf{x}; y)$, respectively.

The pretrained network focuses on measuring the relationship between a random ID input and the corresponding ground truth label in the pretraining phase. Then, it focuses on rejecting OOD samples in the finetuning phase to improve the OOD sensitivity. Based on the definitions of ID and OOD samples, the mutual information of the two kinds of samples should be enhanced and reduced, respectively. We thus have,

$$\max \quad \mathcal{I}_I(\mathbf{x}; y) - \mathcal{I}_O(\mathbf{x}; y). \tag{5.1}$$

To maximize Eq. (5.1) and introduce $Q_\theta(y|\mathbf{x})$ to estimate $P(y|\mathbf{x})$, we obtain the lower

[1] Contrary to the definition of ground truth labels, complementary labels indicate the classes a given input does not belong to.

bound of $\mathcal{I}_I(\mathbf{x}; y)$

$$\begin{aligned}
\mathcal{I}_I(\mathbf{x}; y) =& \mathbb{E}_{P_I(\mathbf{x},y)}\left[\log \frac{P_I(y|\mathbf{x})}{P_I(y)}\right] \\
=& \mathbb{E}_{P_I(\mathbf{x},y)}\left[\log \frac{P_I(y|\mathbf{x})Q_\theta(y|\mathbf{x})}{P_I(y)Q_\theta(y|\mathbf{x})}\right] \\
=& \mathbb{E}_{P_I(\mathbf{x},y)}\left[\log Q_\theta(y|\mathbf{x})\right] + D_{KL}\left(P_I(\mathbf{x}, y) \| Q_\theta(y|\mathbf{x})\right) + H(P_I(y)) \\
\geq& \int \log Q_\theta(y|\mathbf{x}) \mathrm{d}P_I(\mathbf{x}, y) + H(P_I(y))
\end{aligned} \tag{5.2}$$

where the inequality is due to the nonnegative property of the Kullback-Leibler divergence, and the the upper bound of $\mathcal{I}_O(\mathbf{x}; y)$

$$\begin{aligned}
\mathcal{I}_O(\mathbf{x}; y) &= \mathbb{E}_{P_O(\mathbf{x},y)}\left[\log \frac{P_O(y|\mathbf{x})}{P_O(y)}\right] \\
&= \mathbb{E}_{P_O(\mathbf{x},y)}\left[\log \frac{P_O(y|\mathbf{x})\left(1 - Q_\theta(y|\mathbf{x})\right)}{P_O(y)\left(1 - Q_\theta(y|\mathbf{x})\right)}\right] \\
&= -\mathbb{E}_{P_O(\mathbf{x},y)}\left[\log\left(1 - Q_\theta(y|\mathbf{x})\right)\right] + \mathbb{E}_{P_O(\mathbf{x},y)}\left[\log \frac{P_O(y|\mathbf{x})\left(1 - Q_\theta(y|\mathbf{x})\right)}{P_O(y)}\right] \\
&\leq -\mathbb{E}_{P_O(\mathbf{x},y)}\left[\log\left(1 - Q_\theta(y|\mathbf{x})\right)\right] + \mathbb{E}_{P_O(\mathbf{x},y)}\left[\log \frac{\left(1 - Q_\theta(y|\mathbf{x})\right)}{P_O(y)}\right] \\
&\leq -\mathbb{E}_{P_O(\mathbf{x},y)}\left[\log\left(1 - Q_\theta(y|\mathbf{x})\right)\right] - 1 \\
&\quad + \mathbb{E}_{P_O(\mathbf{x},y)}\left[\frac{1 - Q_\theta(y|\mathbf{x})}{P_I(y)}\right] + \mathbb{E}_{P_O(\mathbf{x},y)}\left[\log \frac{P_I(y)}{P_O(y)}\right] \\
&\leq -\mathbb{E}_{P_O(\mathbf{x},y)}\left[\log\left(1 - Q_\theta(y|\mathbf{x})\right)\right] + \mathbb{E}_{P_O(\mathbf{x},y)}\left[\frac{1}{P_I(y)}\right] - D_{KL}\left(P_O(y) \| P_I(y)\right) \\
&\leq -\int \log(1 - Q_\theta(y|\mathbf{x})) \mathrm{d}P_O(\mathbf{x}, y) + \mathbb{E}_{P_O(y)}\left[\frac{1}{P_I(y)}\right],
\end{aligned} \tag{5.3}$$

where the first inequality is due to $P_O(y|\mathbf{x}) \geq 1$; the second inequality uses the logarithm inequality $\log(x) \leq \frac{x}{a} + \log(a) - 1$ for all $x, a \geq 0$; the third inequality is due to $Q_\theta(y|\mathbf{x}) \geq 0$; the last is due to the nonnegative property of the Kullback-Leibler divergence; $H(\cdot)$ is the entropy. Substituting Eq. (5.2) and Eq. (5.3) into Eq. (5.1) and ignoring the constant terms, we obtain the generic expected risk for learning from both ID and OOD samples,

$$\mathcal{R}(\theta) = -\int \log Q_\theta(y|\mathbf{x}) \mathrm{d}P_I(\mathbf{x}, y) + \int \log(1 - Q_\theta(y|\mathbf{x})) \mathrm{d}P_O(\mathbf{x}, y). \tag{5.4}$$

The first term in $\mathcal{R}(\theta)$ is the expected risk for learning a pretrained $Q_\theta(y|\mathbf{x})$ without considering the OOD sensitivity. We finetune $Q_\theta(y|\mathbf{x})$ to improve the ability in dis-

criminating between ID and OOD samples according to $\mathcal{R}(\theta)$.

5.2.2 Cross-class Vicinity Distribution

5.2.2.1 Empirical Distribution

Optimizing $Q_\theta(y|\mathbf{x})$ by minimizing $\mathcal{R}(\theta)$ is intractable because we usually cannot obtain analytic expressions for $P_I(\mathbf{x}, y)$ and $P_O(\mathbf{x}, y)$. We can only access a training ID dataset $\mathcal{D}_I = \{(\mathbf{x}_i^I, y_i^I)\}_{i=1}^{N_I}$, where each sample $(\mathbf{x}_i^I, y_i^I)$ is assumed to be IID drawn from the unknown $P_I(\mathbf{x}, y)$, and N_I is the number of ID samples. According to the vicinal risk minimization principle [82], we can approximate $\mathcal{R}(\theta)$ by replacing $P_I(\mathbf{x}, y)$ and $P_O(\mathbf{x}, y)$ with the corresponding empirical distributions $\widetilde{P}_I(\mathbf{x}, y)$ and $\widetilde{P}_O(\mathbf{x}, y)$. $\widetilde{P}_I(\mathbf{x}, y)$ and $\widetilde{P}_O(\mathbf{x}, y)$ are constructed by the corresponding vicinity distributions $\mathcal{VI}$ and $\mathcal{VO}$ based on the training ID dataset $\mathcal{D}_I$, respectively,

$$\begin{aligned} \widetilde{P}_I(\mathbf{x}, y) &= \frac{1}{N_I} \sum_{i=1}^{N_I} \mathcal{VI}(\mathbf{x}, y|\mathbf{x}_i^I, y_i^I), \\ \widetilde{P}_O(\mathbf{x}, y) &= \frac{1}{N_I} \sum_{i=1}^{N_I} \mathcal{VO}(\mathbf{x}, y|\mathbf{x}_i^I, y_i^I). \end{aligned} \tag{5.5}$$

To obtain the two empirical distributions (i.e., $\widetilde{P}_I(\mathbf{x}, y)$ and $\widetilde{P}_O(\mathbf{x}, y)$) for exploring ID and OOD samples, we have to define the corresponding *vicinity distributions* (i.e., $\mathcal{VI}$ and $\mathcal{VO}$) measuring the probability of finding the *virtual* input-label pairs in the vicinity based on a given ID sample $(\mathbf{x}^I, y^I)$ drawn from $P_I(\mathbf{x}, y)$. Note that $\widetilde{P}_O(\mathbf{x}, y)$ is an empirical distribution for generating OOD samples, which are also built on the ID samples as $\widetilde{P}_I(\mathbf{x}, y)$. This is because the samples outside the ID dataset could be OOD, and the vicinity distribution finding the neighborhood around an ID sample could explore the data-dependent OOD samples.

5.2.2.2 Dirac Delta Vicinity Distribution

In the pretraining phase, we only apply the ID samples to train a network without exploring the samples outside the training ID dataset. Therefore, the corresponding *dirac delta vicinity distribution* for ID samples is defined as

$$\mathcal{VI}(\mathbf{x}, y|\mathbf{x}^I, y^I) = \delta(\mathbf{x} = \mathbf{x}^I, y = y^I), \tag{5.6}$$

where δ is the Dirac delta function. However, the vicinity distribution $\mathcal{VI}(\mathbf{x}, y|\mathbf{x}^I, y^I)$ cannot be applied to find samples different from the ID samples, which causes uncertain predictions for OOD samples and unexpected high-confidence predictions for some of them.

5.2.2.3 Cross-class Vicinity Distribution

We thus construct another vicinity distribution $\mathcal{VO}(\mathbf{x}, y|\mathbf{x}^I, y^I)$ to explore OOD samples by considering the vicinity relations among the ID samples of different classes. A pretrained network is taught to reject the OOD samples drawn from $\widetilde{P}_O(\mathbf{x}, y)$, which improves the OOD sensitivity. Considering the vicinity relations across ID samples of different classes, we combine different classes of ID inputs to generate an OOD input. However, it is difficult to determine the ground truth label for the OOD input. This is because the input contains more than one label information. We constrain networks to provide low-confidence predictions for inputs containing different label information for improving the discriminability of ID and OOD samples. Consequently, determining the complementary labels for the OOD input is easy, i.e., the ground truth labels of the ID inputs. The construction method of the vicinity distribution of ID samples for exploring OOD samples is based on the insight: an OOD input generated by mixing multiple ID inputs does not belong to the classes of these ID inputs.

For a given ID sample $(\mathbf{x}^I, y^I)$, we firstly generate an OOD input $\mathbf{x}^O$ by combining $\mathbf{x}^I$ with other $M-1$ randomly-selected ID inputs $\{\mathbf{x}_1^I, \ldots, \mathbf{x}_{M-1}^I\}$,

$$\mathbf{x}^O = \frac{1}{M}(\mathbf{x}^I + \sum_{i=1}^{M-1} \mathbf{x}_i^I), \tag{5.7}$$

the corresponding label set of the M selected samples is $C(M) = (\bigcup_{i=1}^{M-1} y_i^I) \cup y^I \subseteq [K]$. Recall that the number of classes of ID samples is K. We assume the number of selected classes in $C(M)$ is K_C.

Theorem 5.2.1. *With a high probability, the number of selected classes K_C is less than K, which indicates the label set $C(M)$ is nearly impossible to contain all the labels of ID samples.*

Proof. The problem of calculating the number of selected classes K_C for M selected ID samples and K classes is equivalent to calculating the number of allocation schemes for M balls and K boxes where the boxes could be empty [86]. According to dynamic programming [87], the number of selected classes K_C among the M samples satisfies the following distribution,

$$P(K_C) = \frac{d(M, K_C)}{\sum_{i=1}^{K} d(M, i)}, \tag{5.8}$$

where

$$\begin{aligned} d(M, K_C) &= \sum_{i=1}^{K_C} d(M - K_C, i), M \geq K_C, \\ d(M, K_C) &= 0, M < K_C, d(M, 1) = d(M, M) = 1. \end{aligned} \tag{5.9}$$

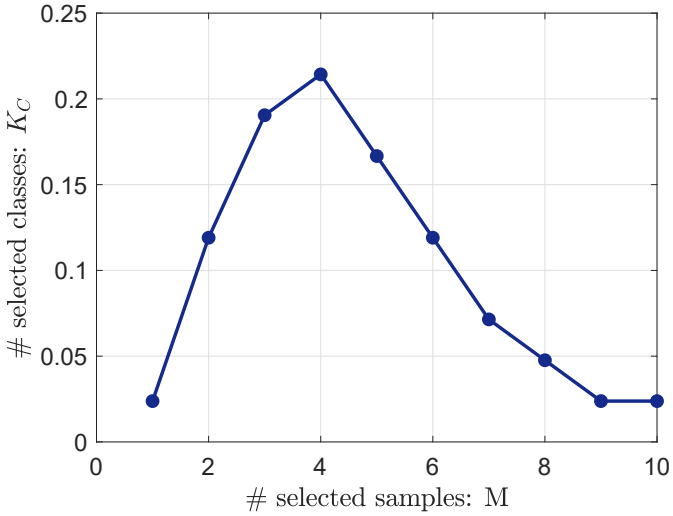


Fig. 5.1 LCVD: An illustration of the distribution over K_C when the number of classes K and the number of selected samples M are equal to 10.

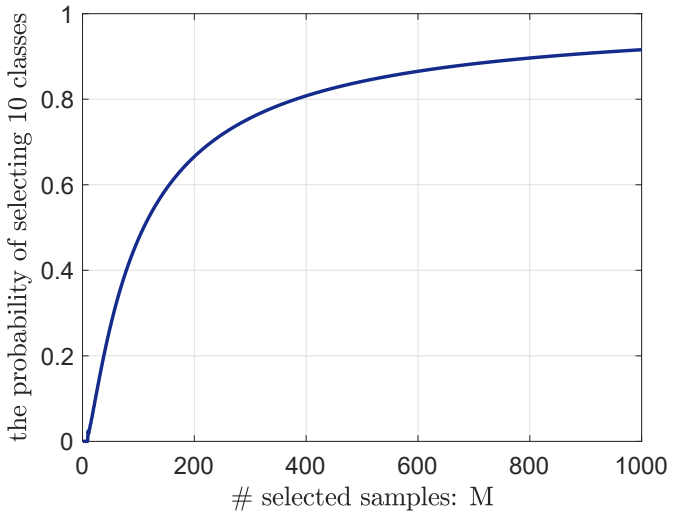


Fig. 5.2 LCVD: An illustration of the probability of $K_C = K = 10$ for different number of selected samples M.

The distribution over K_C when $M = K = 10$ is presented in Fig. 5.1. When the number of components in $\mathbf{x}^O$ equals $K = 10$, the number of labels contained in $\mathbf{x}^O$ is $K_C(K_C \leq K)$, and the highest probability corresponds to 4 different classes. The probability of $K_C = K = 10$ for different number of selected samples M is presented in Fig. 5.2, which indicates that the M selected samples can only contain all classes of samples when M is extremely large $M > 1000$. □

According to Theorem 5.2.1, an OOD input $\mathbf{x}^O$ integrating M different ID inputs cannot contain all the label information with a high probability. The $\mathbf{x}^O$ constructed by linearly combining M ID samples of K_C classes does not belong to the K_C classes, and the complementary label set of $\mathbf{x}^O$ is $C(M)$. For the constructed OOD input $\mathbf{x}^O$, the

complementary label is randomly-selected from the complementary label set $C(M)$,

$$y^O \sim C(M) = (\bigcup_{i=1}^{M-1} y_i^I) \cup y^I. \tag{5.10}$$

From the vicinity distribution perspective, we draw the OOD sample $(\mathbf{x}^O, y^O)$ from an empirical distribution constructed by a vicinity distribution of the ID samples $(\mathbf{x}^I, y^I)$ which considers the vicinity relations among samples of different classes. According to the input Eq. (5.7) and the complementary label Eq. (5.10) of the OOD sample, we obtain the following *cross-class vicinity distribution* for exploring OOD samples based on the given ID sample $(\mathbf{x}^I, y^I)$.

$$\mathcal{VO}(\mathbf{x}, y|\mathbf{x}^I, y^I) = \mathbb{E}_{\mathbf{x}_1^I} \ldots \mathbb{E}_{\mathbf{x}_{M-1}^I} \left[\delta\left(\mathbf{x} = \mathbf{x}^O, y = y^O\right)\right], \tag{5.11}$$

where $\mathbf{x}^O$ and y^O are constructed by Eq (5.7) and Eq (5.10), respectively.

We can obtain the two empirical distributions $\widetilde{P}_I(\mathbf{x}, y)$ and $\widetilde{P}_O(\mathbf{x}, y)$ by substituting Eq (5.6) and Eq (5.11) into Eq (5.5) and have

$$\begin{aligned}
\widetilde{P}_I(\mathbf{x}, y) &= \frac{1}{N_I} \sum_{i=1}^{N_I} \delta(\mathbf{x} = \mathbf{x}_i^I, y = y_i^I), \\
\widetilde{P}_O(\mathbf{x}, y) &= \frac{1}{N_I} \sum_{i=1}^{N_I} \mathbb{E}_{\mathbf{x}_1^I} \ldots \mathbb{E}_{\mathbf{x}_{M-1}^I} \left[\delta\left(\mathbf{x} = \mathbf{x}^O, y = y^O\right)\right].
\end{aligned} \tag{5.12}$$

Both empirical distributions are estimated by ID samples. The dirac delta vicinity distribution $\mathcal{VI}$ in $\widetilde{P}_I(\mathbf{x}, y)$ degenerates into a simple Dirac delta function without exploring the samples outside the training dataset $\mathcal{D}$. Conversely, the cross-class vicinity distribution $\mathcal{VO}$ in $\widetilde{P}_O(\mathbf{x}, y)$ combines different classes of ID samples to find the OOD samples outside $\mathcal{D}$.

5.2.3 Generic Empirical Risk

In the pretrained phase, only ID samples are available, with $P_I(\mathbf{x}, y) \approx \widetilde{P}_I(\mathbf{x}, y)$, we approximate the expected risk $\mathcal{R}(\theta)$ by the following empirical risk

$$\mathcal{R}(\theta) \approx -\sum_{i=1}^{N_I} \log Q_\theta(y_i^I|\mathbf{x}_i^I), \tag{5.13}$$

and learn the pretrained network Q_θ by minimizing Eq. (5.13) on ID samples. To improve the OOD sensitivity of the pretrained network Q_θ in the finetuning phase, we consider a generic expected risk Eq. (5.4) introducing OOD samples for learning to reject the mapping between the inputs and the corresponding complementary labels.

Using $P_I(\mathbf{x}, y) \approx \widetilde{P}_I(\mathbf{x}, y)$ and $P_O(\mathbf{x}, y) \approx \widetilde{P}_O(\mathbf{x}, y)$, we approximate the generic expected risk $\mathcal{R}(\theta)$ by the following generic empirical risk

$$\mathcal{R}(\theta) \approx \widetilde{\mathcal{R}}(\theta) = -\sum_{i=1}^{N_I} \log Q_\theta(y_i^I|\mathbf{x}_i^I) - \sum_{j=1}^{N_O} \log\left(1 - Q_\theta(y_j^O|\mathbf{x}_j^O)\right), \tag{5.14}$$

where N_O is the number of OOD samples drawn from $\widetilde{P}_O(\mathbf{x}, y)$. Based on Monte Carlo [58], we apply the stochastic gradient descent optimization algorithm [17] to estimate the gradients of Eq. (5.13) and Eq. (5.14). The pseudo-code of the finetuning procedure is summarized in Algorithm 4.

Algorithm 4 The pseudo-code of LCVD.

Input: pretrained network Q_θ,
ID training dataset $\mathcal{D} = \{(\mathbf{x}_i^I, y_i^I)\}_{i=1}^{N_I}$,
batch size b, learning rate μ
repeat
  Draw $b_I = b/2$ ID samples from $\widetilde{P}_I(\mathbf{x}, y)$: $\{(\mathbf{x}_i^I, y_i^I)\}_{i=1}^{b_I}$
  Draw $b_O = b/2$ OOD samples from $\widetilde{P}_O(\mathbf{x}, y)$: $\{(\mathbf{x}_j^O, y_j^O)\}_{j=1}^{b_O}$
  Estimate the objective function:

$$\widetilde{\mathcal{R}}(\theta) = -\sum_{i=1}^{b_I} \log Q_\theta(y_i^I|\mathbf{x}_i^I) - \sum_{j=1}^{b_O} \log\left(1 - Q_\theta(y_j^O|\mathbf{x}_j^O)\right)$$

  Obtain gradients: $\nabla_\theta \widetilde{\mathcal{R}}(\theta)$
  Update parameters: $\theta = \theta + \mu \nabla_\theta \widetilde{\mathcal{R}}(\theta)$
until convergence
Output: finetuned network Q_θ

5.3 Experiments

In this section, we verify the effectiveness of the proposed LCVD method. We compare it with different OOD detectors and retraining methods in terms of OOD detection performance and ID classification accuracy. Furthermore, we analyze the effect of the number of selected ID samples M for constructing OOD samples, compare different training mechanisms with the generated OOD samples, and run a set of ablation study experiments about the inputs and labels of the generated OOD samples.

5.3.1 Setup

We adopt the ResNet18 architecture [1] for all the experiments and implement it in PyTorch. The network setups follow that used in FIG method. If not specified, we

Table 5.1 LCVD: OOD detection performance (compared with four detectors).

All values are in percentage, and boldface values show the relatively better detection performance.

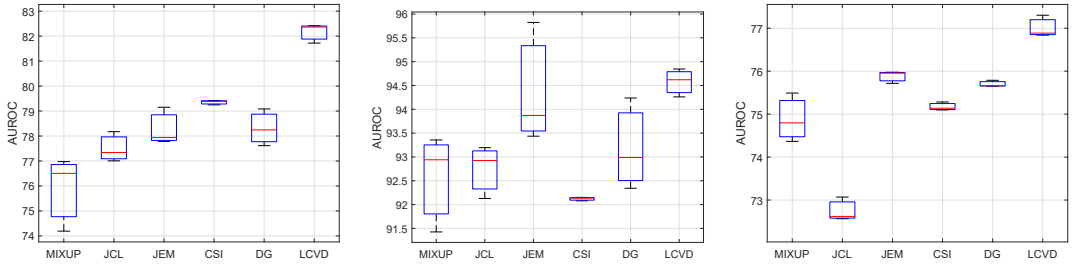
In-dist	Out-of-dist	Baseline	ODIN	MLB	Energy	LCVD
CIFAR10	CIFAR100	87.0	86.0	85.8	86.5	**89.7**
	CUB200	61.5	56.0	66.5	57.1	**67.1**
	StanfordDogs120	68.0	64.9	**72.4**	66.9	70.7
	OxfordPets37	62.9	60.2	65.3	61.6	**66.6**
	Oxfordflowers102	88.1	87.5	88.9	87.3	**90.4**
	Caltech256	86.0	86.4	85.2	85.6	**88.2**
	DTD47	89.4	91.0	89.2	90.6	**94.5**
	COCO	87.3	87.8	86.6	87.6	**90.1**
SVHN	CIFAR100	92.6	92.7	**94.7**	93.5	94.3
	CUB200	92.4	93.4	**94.7**	94.5	94.2
	StanfordDogs120	92.6	93.2	94.2	94.0	**94.8**
	OxfordPets37	92.8	93.6	94.2	94.3	**94.7**
	Oxfordflowers102	95.3	96.3	94.8	**96.7**	95.3
	Caltech256	91.2	91.8	**94.5**	92.4	93.6
	DTD47	92.3	89.5	94.3	92.6	**94.4**
	COCO	92.6	92.9	94.8	93.7	**95.3**
Mini-Imagenet	CIFAR100	84.3	85.8	84.1	83.9	**90.3**
	CUB200	71.8	71.5	**77.1**	70.1	73.8
	StanfordDogs120	65.3	63.6	62.0	64.0	**67.1**
	OxfordPets37	70.2	68.6	64.5	69.5	**71.3**
	Oxfordflowers102	79.8	80.4	76.5	77.9	**83.0**
	Caltech256	78.1	79.9	71.2	78.5	**80.7**
	DTD47	72.5	73.9	76.1	71.9	**82.7**
	COCO	78.5	79.2	73.9	77.9	**79.7**

randomly select $M = 10$ ID samples to construct an OOD sample in the proposed method and finetune the pretrained network until convergence.

We can only access ID datasets to train networks in the training phase. In the test phase, we evaluate the OOD detection performance on diverse real-world OOD datasets and the test ID datasets corresponding to the training ones. The training ID datasets used in our experiments to train neural networks include CIFAR10 [60], SVHN [59], and Mini-Imagenet [63]. For data augmentation methods, we apply random cropping and random horizontal flipping to CIFAR10 and SVHN and resizing and random cropping to Mini-Imagenet. The test OOD datasets evaluating the detection performance include CIFAR100 [60], CUB200 [88], StanfordDogs120 [89], OxfordPets37 [90], Oxfordflowers102 [91], Caltech256 [64], DTD47 [92], and COCO [65]. We resize the test OOD samples to match the size of training ID samples.

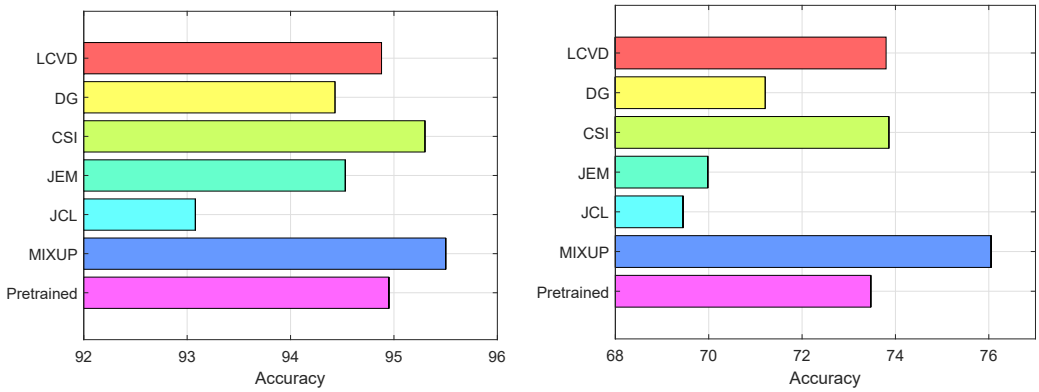
5.3.1.1 Comparison with Detectors

We compare the proposed LCVD method with four OOD detectors, including the Baseline [6], ODIN [24], MahaLanoBis (MLB) [26], and energy-based detector (Energy) [29], in terms of AUROC. All comparison methods follow the same setups as the original ones. All the detectors utilize the outputs from pretrained networks without



(a) CIFAR10 **(b)** SVHN **(c)** Mini-Imagenet

Fig. 5.3 LCVD: OOD detection performance (compared with retraining methods). We calculate the average AUROC value of each comparison method across the eight test OOD datasets. Each box is drawn over five random trials of a method. On each box, the central mark indicates the median, and the bottom and top edges of the box indicate the 25th and 75th percentile, respectively.



(a) CIFAR10 **(b)** Mini-Imagenet

Fig. 5.4 LCVD: ID classification accuracy.
A longer bar indicates a better classification result.

modifying the network parameters.

The comparison results are presented in Table 5.1. We observe that LCVD achieves the highest AUROC values on 18 of 24 pairs of an ID dataset and an OOD dataset. LCVD thus significantly outperforms the detectors on detecting OOD samples, which is mainly achieved by learning to reject OOD samples. Specifically, LCVD improves the OOD sensitivity of a pretrained network in the finetuning phase by rejecting the generated samples that are data-dependent OOD samples with complementary labels from cross-class vicinity distribution of ID samples.

5.3.1.2 Comparison with Retraining Methods

We compare the proposed LCVD method with five retraining methods, including Joint Confidence Loss (JCL) [37], Joint Energy-based Model [29], MIXUP [13], Contrasting Shifted Instances (CSI) [39], and Deep Gambler (DG) [33], in terms of AUROC. The target of the retraining methods is to improve the OOD sensitivity of a given pretrained

network by modifying its training process and objective function with extra knowledge. The settings of all the comparison methods follow the original ones.

The comparison results are presented in Table 5.3. We evaluate the performance of each method over five random trials. LCVD achieves averagely 5.54%, 1.70% and 2.87% improvement over the other state-of-the-art retraining methods in terms of AUROC on datasets CIFAR10, SVHN, Mini-Imagenet, respectively. The results indicate that LCVD can obtain the best performance. Furthermore, the performance gap of LCVD is within the narrow range $[-0.4, 0.2]$, which indicates that the performance of LCVD is stable. Therefore, for a pretrained network learned from a training ID dataset, the OOD samples drawn from the cross-class vicinity distribution of training ID samples can effectively improve the OOD sensitivity. This is because the generated OOD samples are specific to the training ID samples, which can explore the outside ranges of the training samples. The pretrained networks refuse to map data-dependent OOD inputs to the corresponding complementary labels, which indicates samples in the outside ranges are encouraged to have low-confidence predictions.

5.3.1.3 Comparison of Classification Accuracy

We compare the proposed LCVD method with the pretrained network and the five retraining methods in terms of classification accuracy. The pretrained network can represent the performance of the OOD detectors that do not modify the training process and objective function.

The comparison results are presented in Table 5.4. We observe that CSI and MIXUP methods only outperform the baseline method on the two datasets. The reasons include: (1) MIXUP is applied to improve the generalization by generating more ID samples by convex combinations; (2) the rotated samples used in CSI are also ID samples. The traditional augmented ID samples aim to learn invariable features to improve the performance of recognizing different classes of ID samples. We also observe that the classification performance of LCVD is similar to the baseline method on CIFAR10 and slightly better on Mini-Imagenet. However, the rest methods achieve poor classification performance. Therefore, LCVD improves the OOD sensitivity significantly by sacrificing only tiny classification accuracy. This is because LCVD generates OOD samples by augmenting the training ID samples, and the adaptively generated samples mildly affect the classification learning process.

5.3.2 Parameter Analyses

We analyze the affect of the number of selected ID samples M for constructing an OOD sample. We select the value of M from $\{1, 5, 10, 15, 20\}$. According to Fig. 5.1 and Fig. 5.2, we observe that the selected ID samples cannot cover all labels with a high

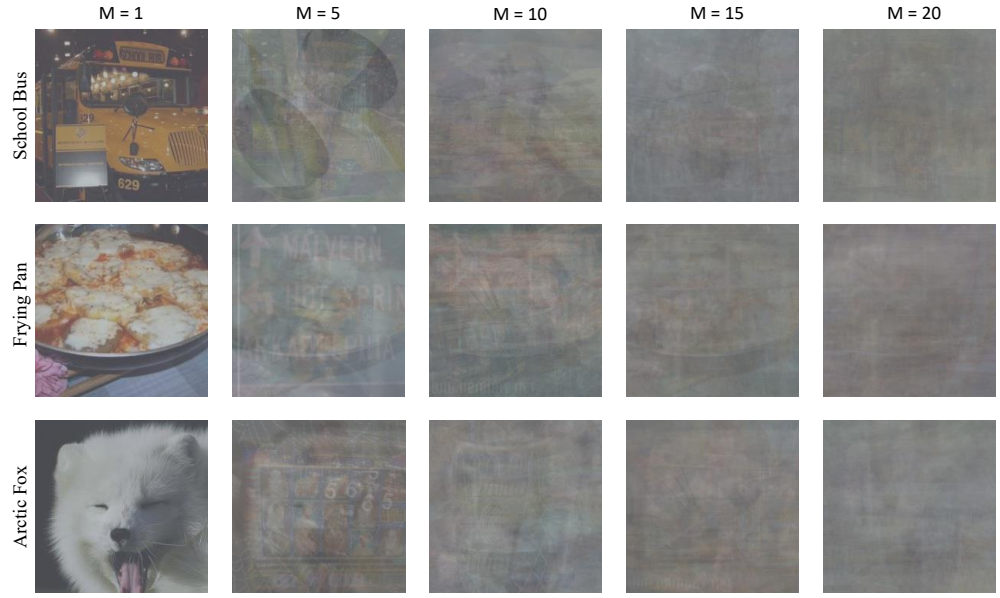


Fig. 5.5 LCVD: OOD samples drawn from the cross-class vicinity distribution.

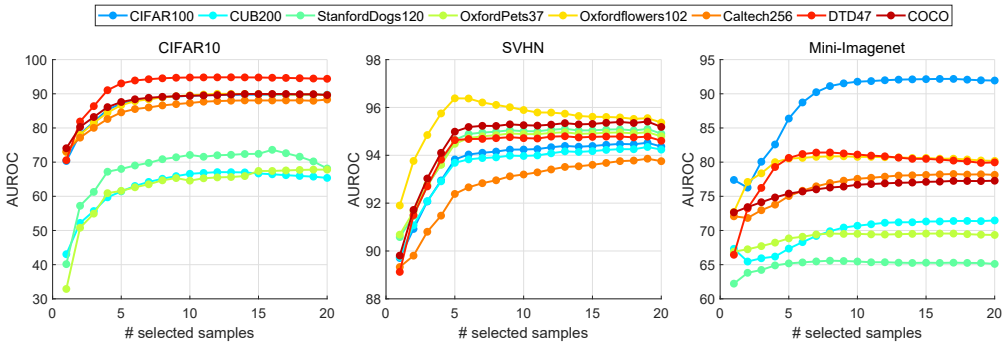


Fig. 5.6 LCVD: Effect of the number of selected ID samples M for constructing an OOD sample.

Each point indicates an AUROC value, and each line represents an OOD dataset.

probability even if $M = 20$ on the three training ID datasets.

The generated OOD samples with different M are shown in Fig. 5.5. Note that when $M = 1$, the original training ID samples are regarded as OOD samples. When $M = 5$, we observe the generated samples are still similar to the corresponding ID samples, which indicates the generated samples tend to be ID. When $M \geq 10$, the generated samples significantly differ from the corresponding ID samples, which indicates the generated samples tend to be OOD. Furthermore, different ID samples lead to specific generated OOD samples.

The experimental results are summarized in Fig. 5.6. We observe that the OOD detection performance is enhanced with the increase of M. Furthermore, this increasing trend diminishes when M is sufficiently large (e.g. $M \geq 20$). Combining tremendous ID samples to construct an OOD sample is expensive. We thus apply $M = 10$ for LCVD to balance efficiency and effectiveness. When M is small, the generated samples tend

Table 5.2 LCVD: Effect of the retraining and finetuning mechanisms.

The results are the average AUROC value across the eight test OOD datasets. All values are in percentage, and boldface values show the relatively better detection performance.

In-distribution	retrain	finetune
CIFAR10	81.77	**82.16**
SVHN	94.4	**94.57**
Mini-Imagenet	76.81	**77.01**

to become ID. The pretrained network rejecting these samples reduces the prediction confidence for ID samples, which narrows the confidence gap between ID and OOD samples. Conversely, when M is large, the generated samples tend to become OOD. The pretrained network rejecting these samples increases the prediction confidence on ID samples which enlarges the confidence gap between ID and OOD samples.

5.3.3 Training Mechanism

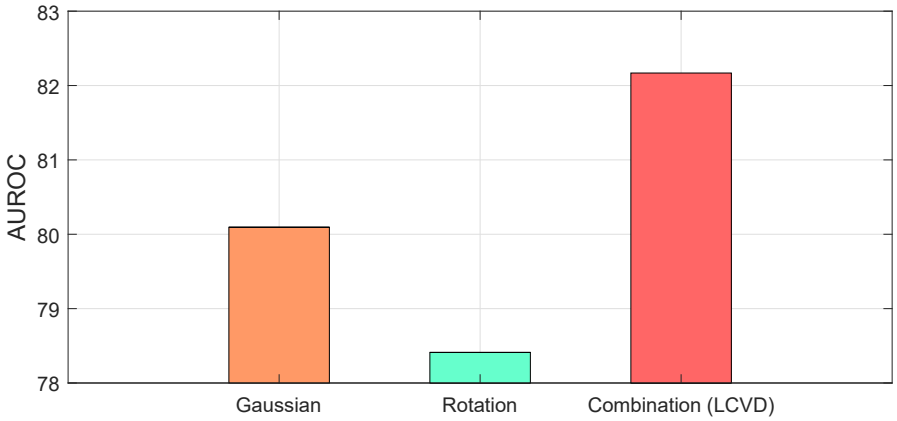
We analyze the effect of the two different training mechanisms, i.e., retraining and finetuning, for the proposed method. The results are summarized in Table 5.2. We observe narrow performance gaps between the two different mechanisms. Furthermore, the finetuning method is slightly better (0.26% to 0.48%) than the retraining method. Recall that both the pretrained network and the generated OOD samples depend on the training ID dataset. Therefore, for rejecting OOD samples, the network should have acquired knowledge about the ID samples. According to the learned knowledge, the pretrained network can discriminate between the OOD samples. In contrast, the retrained network decreases the prediction confidence in ID samples to narrow the confidence gap between ID and OOD samples. It is because the OOD samples generated by mixing multiple ID samples still contain the ID information. Therefore, we apply the finetuning mechanism for LCVD to balance effectiveness and efficiency.

5.3.4 Ablation Study

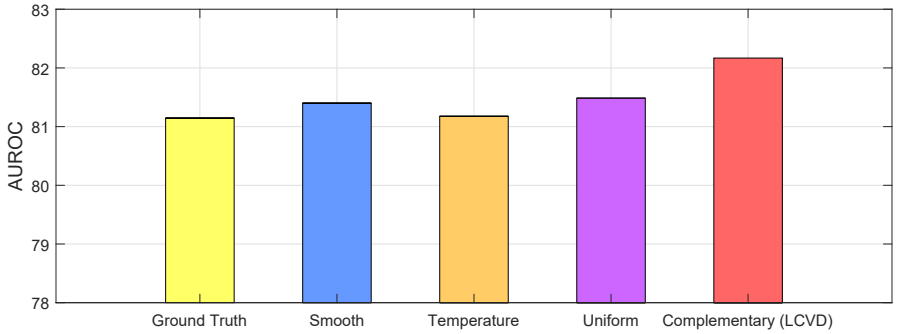
To verify that the input linearly combining multiple ID samples and the corresponding complementary labels are indispensable for generating effective OOD samples, we run a set of ablation study experiments.

5.3.4.1 Diverse Out-of-distribution Inputs

For the generated OOD samples in LCVD, the inputs are replaced with Gaussian noise and rotation of ID inputs [38] without changing the complementary labels to generate the other two variants. The Gaussian noise and rotation inputs correspond to complementary labels. Therefore, the same objective function Eq. (5.14) is applied as LCVD



(a) Effect of OOD Input



(b) Effect of OOD Labels

Fig. 5.7 LCVD: Results of the ablation study.
(a) Replacing the inputs (that linearly combine multiple ID inputs) of the constructed OOD samples in LCVD with other kinds of inputs. (b) Replacing the (complementary) labels of the constructed OOD samples in LCVD with other kinds of labels. Each bar indicates the average AUROC value across the eight test OOD datasets. A higher bar indicates a better detection result.

to refine the pretrained network on the two variants. The comparison results of diverse OOD inputs are summarized in Fig. 5.7a. We observe that rotation leads to the worst result because the rotated inputs are still essentially ID, which should not be rejected by the pretrained network. Gaussian obtains better results because Gaussian noise inputs are OOD. However, Gaussian noise inputs are independent of the training ID samples. Therefore, LCVD considers the relations among samples of different classes to generate specific OOD samples and achieves the best result.

5.3.4.2 Diverse Out-of-distribution Labels

For the generated OOD samples in LCVD, without changing the inputs, the complementary labels are replaced with different pseudo labels, including ID Ground Truth labels, Smooth ground truth labels [57], ground truth labels with Temperature [23], and randomly selected label probability vectors from a Uniform distribution, to generate other four variants. The inputs of these four variants correspond to pseudo labels

rather than complementary labels. Therefore, the objective function used to refine the pretrained network on the four variants is the objective function Eq. (5.13) used in the pretraining phase rather than the same objective function Eq. (5.14) as LCVD.

The comparison results of diverse OOD inputs are summarized in Fig. 5.7b. We observe that LCVD obtains 1% improvement over the other generated OOD samples. The main reason is that the other methods directly define the ground truth labels for the OOD inputs. Because of the complexity of inputs, the defined ground truth labels cannot precisely match the OOD inputs. In contrast, LCVD defines the complementary labels for the OOD inputs. It is difficult to decide the ground truth labels. However, based on the construction method of the cross-class vicinity distribution, it is easy to determine the classes a generated OOD input does not belong to. LCVD thus indirectly defines the learning targets for OOD inputs, which is more conservative and achieves the best result.

5.4 Summary of This Chapter

In this chapter, we propose the Learning from Cross-class Vicinity Distribution (LCVD) method which makes the first attempt to generate specific OOD samples by augmenting ID samples. Considering the vicinity relations between samples of different classes, the cross-class vicinity distribution of ID samples explores OOD samples. An OOD input is generated by linearly combining multiple ID inputs corresponding to a complementary label different from those labels of the constituent ID samples. We finetune a pretrained network to reject the generated OOD samples drawn from the cross-class vicinity distribution of training ID samples, which improves the OOD sensitivity of the pretrained network. Experiments show that LCVD significantly improves OOD detection than the state-of-the-art methods on diverse ID and OOD datasets.

CHAPTER 6

Label and Distribution-discriminative Dual Representation Learning for Out-of-distribution Detection

6.1 Motivations

Chapter 5 discusses how to improve OOD samples by utilizing augmented ID samples. Specifically, it constructs OOD samples by exploring the vicinity distributions of ID samples. Accordingly, we can further consider how to improve OOD sensitivity by exploring the OOD-sensitive information from ID samples. Furthermore, Chapters 3, Chapters 4, and Chapters 5 improve the OOD sensitivity of a given pretrained network by retraining or finetuning. Accordingly, we consider applying an auxiliary network for the pretrained network to explore the OOD-sensitive information from ID samples.

Many existing methods improve the OOD sensitivity of a pretrained network when OOD samples are unavailable in the training process [6, 28, 37, 43], including pretrained and retraining methods. However, these methods ignore the complementary distribution-discriminative representations. For pretrained methods, a pretrained network is learned from ID samples without considering the predictions for OOD samples, an OOD detector is then applied for the pretrained network without modification. The OOD detector does not learn new knowledge from training ID samples, which causes the OOD detection performance to be heavily dependent on the knowledge about label-discriminative representations learned by the pretrained network [93]. To address this issue, retraining methods improve the OOD sensitivity of a pretrained network by retraining it with extra prior knowledge about OOD samples [37, 38, 39, 30]. Retraining methods restrict the output distribution of the network to encourage high- and low-confidence predictions on ID and OOD samples, respectively [16]. However, retraining methods rely heavily on prior knowledge about OOD samples, which indicates that they may not be applicable to unknown OOD samples. Therefore, the existing pretrained and retraining methods suffer from limited OOD detection performance. To address the limitations of the two methods, we explore OOD-sensitive representations from ID samples and distinguish ID and OOD samples according to the different informativeness properties.

We propose a *dual representation learning* (DRL) approach to learn both label- and distribution-discriminative representations, which explores the different informativeness properties of ID and OOD samples. For the generality and flexibility of DRL, we assume a pretrained network, focusing on learning the label-discriminative representations to classify ID samples, is given. We apply an *auxiliary network* to learn the complementary distribution-discriminative representations corresponding to the label-discriminative representations. Accordingly, the pretrained network is trained solely on ID samples, while the auxiliary network is trained on both ID samples and the corresponding label-discriminative representations from the pretrained one. The pretrained and auxiliary networks have the same backbone, and their target is extracting label-related information from inputs to learn representations. However, the label- and distribution-discriminative representations of the two networks are strongly and weakly related to labeling, respectively. To verify the properties of the two representations, for a given ID sample, we set a restriction to ensure that its distribution-discriminative representation is significantly different from its label-discriminative representation and sensitive to the same label.

An implicit constraint is incorporated into the auxiliary network, integrating multiple *intermediate representations* into a complementary distribution-discriminative representation. An intermediate representation less similar to the label-discriminative representation is given a higher weight. After obtaining the auxiliary network depending on the pretrained network, DRL averages the softmax outputs of these two representations and calculates an *out-of-distribution score* (OOD score) for each test sample. Therefore, the OOD scores are then used to distinguish ID and OOD samples for OOD detection. The learning process is summarized in Fig. 6.1.

6.2 Dual Representation Learning

6.2.1 Learning Principle of Label-discriminative Representations

The information bottleneck principle [20] measures a tradeoff between the compression of input and the prediction of its label. The mutual information [48, 94] measures the shared information between variables. Therefore, for a network, the learning process of extracting label-related information from inputs to learn the corresponding representations can be interpreted from the information-theoretic view [42].

Given a dataset $\mathcal{X}$ and its label set $\mathcal{Y}$, an ID sample $\mathbf{x}$ contains all the information about its corresponding label $\mathbf{y}$. The information bottleneck limits the information to provide a prediction $\mathbf{y}$ by compressing $\mathbf{x}$ to learn its label-discriminative representation $\mathbf{d}$. Therefore, the information of the label-discriminative representations $\mathcal{D}$ shares with the labels $\mathcal{Y}$ should be maximized and the information between $\mathcal{D}$ and the inputs $\mathcal{X}$

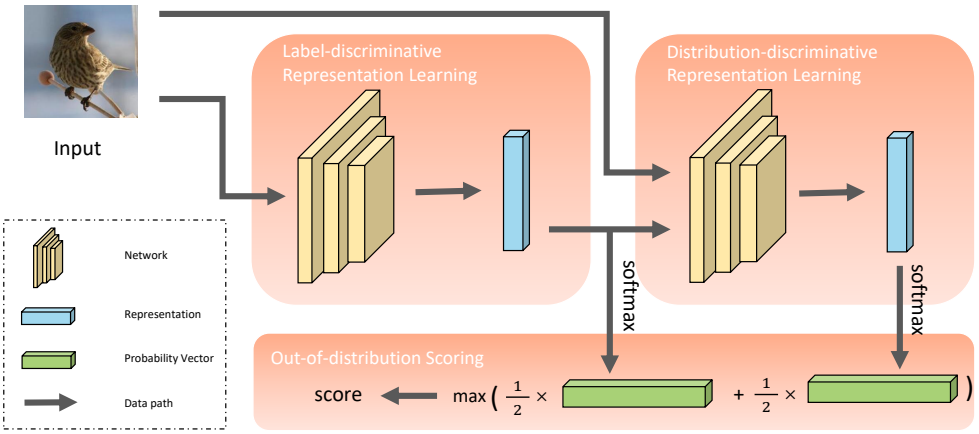


Fig. 6.1 DRL: Learning process.
It includes: (1) learning a pretrained network on training ID samples for label-discriminative representations; (2) learning an auxiliary network on training ID samples and their corresponding label-discriminative representations for distribution-discriminative representations; and (3) calculating the OOD scores by combining these two representations.

should be minimized, i.e.,

$$\max \mathcal{I}(\mathcal{D};\mathcal{Y}) - \beta_{\mathcal{D}}\mathcal{I}(\mathcal{X};\mathcal{D}), \tag{6.1}$$

where $\mathcal{I}(\cdot;\cdot)$ refers to the mutual information, and $\beta_{\mathcal{D}}$ controls the trade-off between learning more information from the labels $\mathcal{Y}$ and retaining less information of the original inputs $\mathcal{X}$ for learning label-discriminative representations $\mathcal{D}$. Specifically, $\mathcal{I}(\mathcal{D};\mathcal{Y})$ determines how much label information is accessible from the label-discriminative representation, and $\mathcal{I}(\mathcal{X};\mathcal{D})$ denotes how much information the label-discriminative representation can acquire from the original input.

Note that $\mathbf{x}$ contains all the information of $\mathbf{d}$ because $\mathbf{d}$ is a representation learned from $\mathbf{x}$. Therefore, we have

$$\mathcal{I}(\mathcal{X};\mathcal{Y}) = \mathcal{I}(\mathcal{X},\mathcal{D};\mathcal{Y}), \tag{6.2}$$

where $\mathcal{I}(\mathcal{X},\mathcal{D};\mathcal{Y})$ denotes the shared information between the labels $\mathcal{Y}$ and the union of $\mathcal{X}$ and $\mathcal{D}$. According to the chain rule [95] of the mutual information, we have,

$$\mathcal{I}(\mathcal{X};\mathcal{Y}) = \mathcal{I}(\mathcal{D};\mathcal{Y}) + \mathcal{I}(\mathcal{X};\mathcal{Y}|\mathcal{D}), \tag{6.3}$$

where $\mathcal{I}(\mathcal{X};\mathcal{Y}|\mathcal{D})$, representing the shared information between $\mathcal{X}$ and $\mathcal{Y}$ given the label-discriminative representations $\mathcal{D}$, is greater than or equals 0. Accordingly, we have $\mathcal{I}(\mathcal{X};\mathcal{Y}) \geq \mathcal{I}(\mathcal{D};\mathcal{Y})$. Therefore, for an ID input $\mathbf{x}$, its label-discriminative representation $\mathbf{d}$ is insufficient for $\mathbf{y}$ because $\mathbf{d}$ does obtain all the information about $\mathbf{y}$.

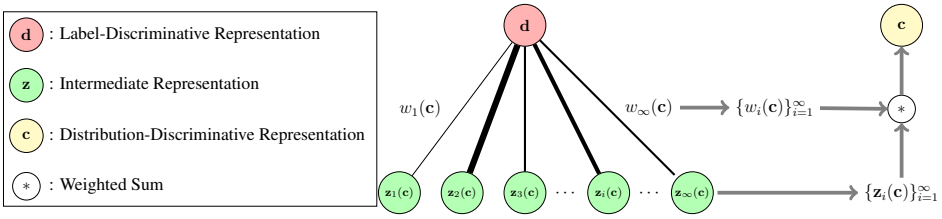


Fig. 6.2 DRL: Constructing a distribution-discriminative representation. A thicker black line indicates a larger weight or vice versa. A distribution-discriminative representation consists of multiple intermediate representations where an intermediate representation less similar to the label-discriminative representation is given a higher weight.

Therefore, we assume there exists another representation for an ID sample $\mathbf{x}$ differing from the label-discriminative representation and containing the remaining information about $\mathbf{y}$, i.e., a complementary distribution-discriminative representation $\mathbf{c}$. The distribution-discriminative representations C also satisfies Eq. (6.2) and Eq. (6.3) since $\mathbf{c}$ is also a representation of the input $\mathbf{x}$. Accordingly, we can obtain the following equation by simply replacing $\mathcal{D}$ in Eq. (6.3) with $\mathcal{C}$,

$$\mathcal{I}(\mathcal{X};\mathcal{Y}) = \mathcal{I}(\mathcal{C};\mathcal{Y}) + \mathcal{I}(\mathcal{X};\mathcal{Y}|\mathcal{C}). \tag{6.4}$$

Since $\mathbf{d}$ is insufficient for selecting $\mathbf{y}$ and there is no shared information between $\mathbf{c}$ and $\mathbf{d}$, we thus assume $\mathcal{I}(\mathcal{X};\mathcal{Y}|\mathcal{D}) = \mathcal{I}(\mathcal{C};\mathcal{Y})$ and have the following equation according to Eqs. (6.3) and (6.4),

$$\mathcal{I}(\mathcal{X};\mathcal{Y}) = \mathcal{I}(\mathcal{D};\mathcal{Y}) + \mathcal{I}(\mathcal{C};\mathcal{Y}). \tag{6.5}$$

Based on Eq. (6.5), both the label- and distribution-discriminative representations corresponding to the same label co-exist for an ID sample. Furthermore, a label-discriminative representation alone cannot contain all the label information of an ID sample. Specifically, for ID samples, its label-discriminative representation and distribution-discriminative representation are strongly and weakly related to labeling, respectively. Note that an OOD sample with high-confidence prediction has a label-discriminative representation which is sensitive to a label. However, its distribution-discriminative representation can correspond to other labels or even none of any labels. Accordingly, by exploiting the label-discriminative representations and exploring the distribution-discriminative representations, we distinguish ID and OOD samples according to different informativeness properties.

6.2.2 Learning Principle of Distribution-discriminative Representations

We present the information bottleneck principle of learning distribution-discriminative representations. For an ID sample $\mathbf{x}$, the pretrained network g_ϕ learns its label-discriminative representation $\mathbf{d}$ to predict the label $\mathbf{y}$ based on the information bottleneck principle in Eq. (6.1). Besides, the corresponding distribution-discriminative representation $\mathbf{c}$ also contains the information about the label $\mathbf{y}$ because $\mathbf{d}$ is insufficient to contain all label information, which indicates that $\mathcal{C}$ also follows the learning principle of Eq. (6.1). Further, according to Eq. (6.5), there is no shared label information between the two kinds of representations, i.e., the mutual information $I(\mathcal{D}, \mathcal{C})$ is expected to be equal to zero. However, in practice, it is impossible to separate the two representations entirely. Based on Eq. (6.1), we thus minimize the amount of shared information $I(\mathcal{D}, \mathcal{C})$ and have

$$\max \mathcal{I}(\mathcal{C}; \mathcal{Y}) - \beta_{\mathcal{C}} \mathcal{I}(\mathcal{X}; \mathcal{C}) - \alpha \mathcal{I}(\mathcal{D}; \mathcal{C}). \tag{6.6}$$

Similar to $\beta_{\mathcal{D}}$, $\beta_{\mathcal{C}}$ controls the amount of information propagated from $\mathcal{X}$ to $\mathcal{C}$. A larger $\beta_{\mathcal{C}}$ leads to more information being extracted from $\mathcal{X}$, which also indicates more label-unrelated information will be extracted to reduce the overlap information between $\mathcal{C}$ and $\mathcal{Y}$. Furthermore, α is a difference coefficient controlling the trade-off between extracting label-related information from the original inputs and enlarging the difference between the label- and distribution-discriminative representations. A larger α causes to less overlap information between $\mathcal{D}$ and $\mathcal{C}$ but less label information to be extracted, or vice versa.

To find a restriction for learning distribution-discriminative representations according to $\mathcal{I}(\mathcal{D}; \mathcal{C})$, we quantify this mutual information term and have

$$\mathcal{I}(\mathcal{D}; \mathcal{C}) = \mathbb{E}_{P(\mathcal{D})} \left[\mathrm{KL} \left(P(\mathcal{C}|\mathcal{D}) \| P(\mathcal{C}) \right) \right], \tag{6.7}$$

where $P(\mathcal{D})$, $P(\mathcal{C})$ and $P(\mathcal{C}|\mathcal{D})$ denote the respective probability distributions, and $\mathrm{KL}(\cdot|\cdot)$ represents the Kullback-Leibler (KL) divergence. However, measuring Eq. (6.7) is intractable because we cannot obtain an analytic expression for $P(\mathcal{C})$. Based on the variational inference [96], we can solve this problem by using a tractable proposal distribution [74] $Q(\mathcal{C})$ to approximate $P(\mathcal{C})$. Therefore, we have

$$\begin{aligned} \mathcal{I}(\mathcal{D}; \mathcal{C}) &= \mathbb{E}_{P(\mathcal{D})} \left[\mathrm{KL} \left(P(\mathcal{C}|\mathcal{D}) \| Q(\mathcal{C}) \right) \right] - \mathrm{KL} \left(Q(\mathcal{C}) \| P(\mathcal{C}) \right) \\ &\leq \mathbb{E}_{P(\mathcal{D})} \left[\mathrm{KL} \left(P(\mathcal{C}|\mathcal{D}) \| Q(\mathcal{C}) \right) \right], \end{aligned} \tag{6.8}$$

where the inequality is due to the nonnegative property of the KL divergence. According to Eq. (6.8), we know that the distribution-discriminative representations rely on

the corresponding label-discriminative representations. Furthermore, the distribution of distribution-discriminative representations $P(\mathcal{C}|\mathcal{D})$ should be close to the proposal distribution $Q(\mathcal{C})$. However, we should decide $Q(\mathcal{C})$ according to prior knowledge which is expected to be similar with the unknown $P(\mathcal{C})$. Therefore, it is hard to construct an explicit constraint, e.g., regularizer, for the distribution-discriminative representation learning due to the unknown proposal distribution $Q(\mathcal{C})$ in Eq. (6.8). The unknown proposal distribution inspires us to find an implicit constraint to ensure that a distribution-discriminative representation is complementary to its label-discriminative representation, i.e., a distribution-discriminative representation contains weakly label-related information discarded by the label-discriminative representation. The three sources of mutual information $\mathcal{I}(\mathcal{C};\mathcal{Y})$, $\mathcal{I}(\mathcal{X};\mathcal{C})$ and $\mathcal{I}(\mathcal{D};\mathcal{C})$ in Eq. (6.6) can be illustrated by a loss function, a network, and a constraint, respectively. Both the loss function and the network are not directly obtained from their mutual information terms. Specifically, the loss function ensures that (1) the distribution-discriminative representations contain label-related information; (2) the network compresses the information from inputs to learn their representations because of the down-sampling nature. Therefore, based on $\mathcal{I}(\mathcal{D};\mathcal{C})$, we implicitly restrict that a distribution-discriminative representation differs from its corresponding label-discriminative representation in the learning process. The implicit constraint encourages the network to explore weakly related label information because the strongly related label information has been explored by label-discriminative representations $\mathcal{D}$. Without considering this constraint, networks will explore the same strongly label-related information in $\mathcal{D}$ from inputs to learn representations according to $\max \mathcal{I}(\mathcal{C};\mathcal{Y}) - \beta_{\mathcal{C}}\mathcal{I}(\mathcal{X};\mathcal{C})$. Because $\mathcal{I}(\mathcal{C};\mathcal{Y})$, $\mathcal{I}(\mathcal{X};\mathcal{C})$ and $\mathcal{I}(\mathcal{D};\mathcal{C})$ in Eq. (6.6) are implicitly modeled, it is unnecessary to explicitly set the hyper-parameters $\beta_{\mathcal{C}}$ and α in Eq. (6.8).

6.2.3 Learning the Auxiliary Network

Following the information bottleneck principle in Eq. (6.6), we apply an auxiliary network f_θ with an implicit constraint to learn a complementary distribution-discriminative representation for its corresponding label-discriminative representation, i.e.,

$$\mathbf{c} = f_\theta(\mathbf{x}, \mathbf{d}). \tag{6.9}$$

Directly modeling $f_\theta(\mathbf{x}, \mathbf{d})$ by a network cannot ensure that $\mathbf{d}$ and $\mathbf{c}$ are different because it is difficult to design an implicit constraint for $f_\theta(\mathbf{x}, \mathbf{d})$ due to the unknown proposal distribution $Q(\mathcal{C})$ in Eq. (6.8). We develop an indirect modeling method by an implicit constraint. The basic idea is to decompose the distribution-discriminative representation $\mathbf{c}$ into multiple intermediate representations where an intermediate representation $\mathbf{z}(\mathbf{c})$ less similar to the label-discriminative representation $\mathbf{d}$ is given a higher

weight $w(\mathbf{c})$, as shown in Fig 6.2.

6.2.3.1 Decompose Distribution-discriminative Representations

We firstly decompose Eq. (6.10) into a linear combination,

$$\mathbf{c} = \sum_{i=1}^{\infty} w_i(\mathbf{c}) \cdot \mathbf{z}_i(\mathbf{c}), \tag{6.10}$$

We assume $\mathbf{z}_i(\mathbf{c})$ is drawn from the Gaussian distribution $\mathcal{N}(\mu_{\mathcal{Z}}(\mathbf{c}), \Sigma_{\mathcal{Z}}(\mathbf{c}))$ without loss of generality. Inspired by the determinant point processes [97, 98] selecting diverse samples according to the kernel-based distance [99], we construct $\mathbf{c}$ by integrating diverse intermediate representations $\{\mathbf{z}_1(\mathbf{c}), \ldots, \mathbf{z}_\infty(\mathbf{c})\}$ where an intermediate representation $\mathbf{z}(\mathbf{c})$ less similar with the corresponding label-discriminative representation $\mathbf{d}$ has a higher weight. Simulating the idea of the attention mechanism [100], the inner product is adopted as the similarity metric. We thus define the weight $w(\mathbf{c})$ as

$$w(\mathbf{c}) = 1 - \epsilon \cdot \mathbf{z}(\mathbf{c})^T \times \mathbf{d}. \tag{6.11}$$

where ϵ is a small perturbation coefficient to ensure that the weight is positive. Substituting Eq. (6.11) into Eq. (6.10), we obtain an implicit constraint on $\mathbf{c}$,

$$\begin{aligned}
\mathbf{c} =& \mathbb{E}_{\mathbf{z}(\mathbf{c})}\left[\mathbf{z}(\mathbf{c}) - \epsilon \cdot \mathbf{z}(\mathbf{c})^T \times \mathbf{d} \times \mathbf{z}(\mathbf{c})\right] \\
=& \mathbb{E}_{\mathbf{z}(\mathbf{c})}\left[\mathbf{z}(\mathbf{c})\right] - \epsilon \cdot \mathbb{E}_{\mathbf{z}}\left[(\mathbf{z}(\mathbf{c}) - \mu_{\mathcal{Z}}(\mathbf{c}))^T \times \mathbf{d} \times (\mathbf{z}(\mathbf{c}) - \mu_{\mathcal{Z}}(\mathbf{c}))\right] \\
& - \epsilon \cdot \mu_{\mathcal{Z}}(\mathbf{c})^T \times \mathbf{d} \times \mu_{\mathcal{Z}}(\mathbf{c}) \\
=& \mu_{\mathcal{Z}}(\mathbf{c}) - \epsilon \cdot \left(\Sigma_{\mathcal{Z}}(\mathbf{c}) \times \mathbf{d} + \mu_{\mathcal{Z}}(\mathbf{c}) \times \mu_{\mathcal{Z}}(\mathbf{c}) \times \mathbf{d}\right).
\end{aligned} \tag{6.12}$$

6.2.3.2 Estimate Distribution-discriminative Representations

According to Eq. 6.12, we can estimate a distribution-discriminative representation $\mathbf{c}$ by estimating the expectation $\mu_{\mathcal{Z}}(\mathbf{c})$ and the covariance matrix $\Sigma_{\mathcal{Z}}(\mathbf{c})$. We estimate $\mu_{\mathcal{Z}}(\mathbf{c})$ by applying an intermediate network z_θ which maps an input $\mathbf{x}$ to the intermediate representation expectation. i.e., $z_\theta(\mathbf{x}) = \mu_{\mathcal{Z}}(\mathbf{c})$. Note that f_θ and z_θ share the same parameter θ since we indirectly construct f_θ by Eq. (6.10), Eq. (6.12) and $z_\theta(\mathbf{x}) = \mu_{\mathcal{Z}}(\mathbf{c})$. The indirect construction method considers the implicit constraint to ensure that the distribution-discriminative representations differ from the corresponding label-discriminative representations. However, the covariance matrix $\Sigma_{\mathcal{Z}}(\mathbf{c})$ is still unknown. We can define the covariance matrix according to our prior knowledge. This is because $\Sigma_{\mathcal{Z}}(\mathbf{c})$ which represents the dispersion degree of intermediate representation $\mathbf{z}(\mathbf{c})$ does not play an important role in learning distribution-discriminative representations. Because the covariance matrix $\Sigma_{\mathcal{D}}$ of ID samples can be easily estimated by

the pretrained network g_θ and z_θ and g_θ own the same network architecture, we assume $\Sigma_{\mathcal{Z}}(\mathbf{c}) = \Sigma_{\mathcal{D}}$ without loss of generality.

6.2.3.3 Objective Function

Following the learning principle of label-discrimintive representations, we use the cross-entropy loss to model $\mathcal{I}(\mathcal{C};\mathcal{Y})$ and assume $h(\cdot,\cdot)$ is the softmax function. Therefore, the loss function for learning distribution-discriminative representations is,

$$\mathcal{L}(\theta) = -\mathbb{E}_{(\mathbf{x},y)\sim P(\mathcal{X},\mathcal{Y})} \log h(f_\theta(\mathbf{x}, g_\phi(\mathbf{x})), y), \tag{6.13}$$

where the implicit constraint is

$$f_\theta(\mathbf{x}, g_\phi(\mathbf{x})) = z_\theta(\mathbf{x}) - \epsilon \cdot \left(\Sigma_{\mathcal{D}} \times g_\phi(\mathbf{x}) + z_\theta(\mathbf{x})^T \times z_\theta(\mathbf{x}) \times g_\phi(\mathbf{x})\right). \tag{6.14}$$

The parameter ϕ is fixed for learning the parameter θ in f_θ because g_ϕ is a pretrained network. Based on Monte Carlo [58], we apply the stochastic gradient descent optimization algorithm [17] to estimate the gradient of Eq. (6.13), where the batch size is B.

6.2.4 Out-of-distribution Score

For an ID sample, both its label- and distribution-discriminative representations contain information corresponding to the same label. For an OOD sample with high-confidence prediction, its label-discriminative representation is sensitive to a label. However, its distribution-discriminative representation is sensitive to other labels or even none of any labels. Accordingly, the labeling information in the label- and distribution-discriminative representations are complementary for ID samples while are inconsistent for OOD samples. Therefore, we detect OOD samples by combining these two representations. We get a softmax output of label y for input $\mathbf{x}$ by simply averaging the softmax outputs of the two representations, i.e.,

$$O(\mathbf{x}, y) = \frac{1}{2} h(f_\theta(\mathbf{x}, g_\phi(\mathbf{x})), y) + \frac{1}{2} h(g_\phi(\mathbf{x}), y). \tag{6.15}$$

We classify ID samples according to the softmax outputs $\{O(\mathbf{x}, 1), \ldots, O(\mathbf{x}, K)\}$. Following the baseline method [6] of detecting OOD samples which uses the confidence as the OOD score, we calculate the OOD score for input $\mathbf{x}$ by

$$\mathcal{S}(\mathbf{x}) = \max_{i\in[1,K]} O(\mathbf{x}, y). \tag{6.16}$$

Algorithm 5 Dual Representation Learning (DRL)

Input: pretrained network g_ϕ, perturbation coefficient ϵ, covariance $\Sigma_{\mathcal{D}}$, batch size B
repeat
  Sample $\{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_B, \mathbf{y}_B)\}$ from $P(\mathcal{X}, \mathcal{Y})$
  Receive $\mathbf{d}_i = g_\phi(\mathbf{x}_i), \forall i \in [B]$
  Calculate $\mathbf{c}_i = f_\theta(\mathbf{x}_i, \mathbf{d}_i), \forall i \in [B]$
  Estimate the objective function:

$$\widetilde{\mathcal{L}}(\theta) = -\frac{1}{B}\sum_{i=1}^{B} \log h(\mathbf{c}_i, y_i)$$

  Obtain gradients $\nabla_\theta \widetilde{\mathcal{L}}(\theta)$ to update parameters θ
until convergence
Calculate out-of-distribution score:

$$\mathcal{S}(\mathbf{x}) = \max_{i \in [1,K]} \left(\frac{1}{2} h(\mathbf{c}, y) + \frac{1}{2} h(\mathbf{d}, y) \right)$$

Output: $\mathcal{S}(\mathbf{x})$

Table 6.1 DRL: OOD detection performance (compared with pretrained methods).

All the reported values are averaged AUROC over five trials. The subscript values denote the standard deviation. The boldface values represent the relatively better detection performance.

In-dist	Method	CIFAR100	CUB200	StanfordDogs120	OxfordPets37	Oxfordflowers102	Caltech256	DTD47	COCO
CIFAR10	Baseline	$87.0_{\pm 0.2}$	$61.5_{\pm 0.4}$	$68.0_{\pm 0.7}$	$62.9_{\pm 1.7}$	$88.1_{\pm 0.6}$	$86.0_{\pm 0.2}$	$89.4_{\pm 1.5}$	$87.3_{\pm 0.1}$
	ODIN	$86.0_{\pm 0.0}$	$56.0_{\pm 0.0}$	$64.9_{\pm 0.0}$	$60.2_{\pm 0.0}$	$87.5_{\pm 0.0}$	$86.4_{\pm 0.0}$	$91.0_{\pm 0.0}$	$87.8_{\pm 0.0}$
	Energy	$86.5_{\pm 0.0}$	$57.1_{\pm 0.0}$	$66.9_{\pm 0.0}$	$61.6_{\pm 0.0}$	$87.3_{\pm 0.0}$	$85.6_{\pm 0.0}$	$90.6_{\pm 0.0}$	$87.6_{\pm 0.0}$
	Mahalanobis	$85.8_{\pm 0.1}$	**66.5**$_{\pm 0.1}$	$72.4_{\pm 0.1}$	$65.3_{\pm 0.1}$	$88.9_{\pm 0.0}$	$85.2_{\pm 0.0}$	$89.2_{\pm 0.0}$	$86.6_{\pm 0.0}$
	DRL	**89.5**$_{\pm 0.1}$	$63.7_{\pm 0.4}$	**73.1**$_{\pm 0.1}$	**67.8**$_{\pm 0.2}$	**91.2**$_{\pm 0.1}$	**88.0**$_{\pm 0.1}$	**92.6**$_{\pm 0.1}$	**89.2**$_{\pm 0.1}$
Mini-Imagenet	Baseline	$84.7_{\pm 0.2}$	$71.8_{\pm 0.4}$	$65.6_{\pm 0.7}$	$70.4_{\pm 1.7}$	$79.5_{\pm 0.6}$	$78.2_{\pm 0.2}$	$72.8_{\pm 1.5}$	$78.9_{\pm 0.1}$
	ODIN	$85.8_{\pm 0.0}$	$71.5_{\pm 0.0}$	$63.6_{\pm 0.0}$	$68.6_{\pm 0.0}$	$80.4_{\pm 0.0}$	**79.8**$_{\pm 0.0}$	$73.9_{\pm 0.0}$	$79.2_{\pm 0.0}$
	Energy	$84.0_{\pm 0.0}$	$70.1_{\pm 0.0}$	$64.0_{\pm 0.0}$	$69.6_{\pm 0.0}$	$78.0_{\pm 0.0}$	$78.5_{\pm 0.0}$	$71.9_{\pm 0.0}$	$78.0_{\pm 0.0}$
	Mahalanobis	$84.1_{\pm 0.1}$	**77.1**$_{\pm 0.1}$	$62.0_{\pm 0.1}$	$64.5_{\pm 0.1}$	$76.5_{\pm 0.0}$	$71.2_{\pm 0.0}$	**76.1**$_{\pm 0.0}$	$73.9_{\pm 0.0}$
	DRL	**86.4**$_{\pm 0.1}$	$73.8_{\pm 0.4}$	**67.3**$_{\pm 0.1}$	**72.2**$_{\pm 0.2}$	**81.4**$_{\pm 0.1}$	**79.8**$_{\pm 0.1}$	$74.5_{\pm 0.1}$	**79.9**$_{\pm 0.1}$

where an ID sample is expected to have a higher score, whereas an OOD sample is expected to have a lower score. The pseudo-code of the DRL training procedure is summarized in Algorithm 5.

6.3 Experiments

In this section, we demonstrate the effectiveness of the proposed DRL method[1]. We compare DRL with pretrained, retraining and ensemble methods. Furthermore, we analyze the effect of the hyper-parameters in DRL, run a set of ablation study experiments, and show the sensitivity of labeling information of label- and distribution-discriminative representations.

[1]The source codes are at: `https://github.com/Lawliet-zzl/DRL`

Table 6.2 DRL: OOD detection performance (compared with retraining methods).

Each value is averaged across all eight OOD datasets. The symbol ↑ indicates a larger value is better, and the symbol ↓ indicates a lower value is better. The boldface value represents the relatively better detection performance.

Dataset	Methods	AUROC ↑	FPR(95) ↓	Detection ↓
CIFAR10	JCL	77.5	73.9	26.7
	CSI	79.2	67.8	24.6
	SSL	78.1	**62.1**	26.6
	DeConf-C	78.4	64.8	25.9
	MOS	77.8	68.2	27.5
	DRL	**81.9**	65.0	**22.5**
Mini-Imagenet	JCL	72.8	86.5	32.0
	CSI	75.2	85.9	29.9
	SSL	75.6	**82.5**	28.9
	DeConf-C	75.3	85.8	29.7
	MOS	75.3	86.3	29.1
	DRL	**76.9**	83.2	**28.3**

6.3.1 Setup

We adopt the ResNet18 architecture [1] for all the experiments and implement it in PyTorch. The network setup follows that used in FIG. For the pretrained network, we train it with a cross-entropy loss on an ID dataset. For the auxiliary network, we adopt $\Sigma_{\mathcal{Z}}(\mathbf{c}) = \Sigma_{\mathcal{D}}$ for constructing distribution-discriminative representations where $\Sigma_{\mathcal{D}}$ is the covariance matrix of label-discriminative representations of ID samples from the pretrained network. For training ID datasets, we adopt CIFAR10 [60] and Mini-Imagenet [63] to train neural networks. For data augmentation methods, we apply random crop and random horizontal flip for CIFAR10 and resizing and random crop for Mini-Imagenet. For OOD datasets, we adopt CIFAR100 [60], CUB200 [88], StanfordDogs120 [89], OxfordPets37 [90], Oxfordflowers102 [91], Caltech256 [64], DTD47 [92], and COCO [65] to evaluate the OOD detection performance in the test phase.

6.3.2 Comparison Results

6.3.2.1 Comparison with Pretrained Methods

We compare DRL with different pretrained methods that utilize an OOD detector according to the outputs from a pretrained network. All the compared methods do not modify pretrained networks. The pretrained networks used in the pretrained methods are the same as that in DRL. The pretrained methods include the Baseline [6],

ODIN [24], the energy-based detector (Energy) [29], and the Mahalanobis distance (Mahalanobis) [26]. For a fair comparison, we add the scores from different layers without training a logistic regression on a validation OOD dataset in Mahalanobis because the other compared methods do not access OOD samples in the training phase. For the other comparison methods, the setups follow the same setups as their original ones.

The comparison results are presented in Table 6.1. When the training ID dataset is CIFAR10, DRL obtains the best OOD detection performance on seven of the nine OOD datasets. Furthermore, when the training ID dataset is Mini-Imagenet, which is a more complex dataset with more classes and higher resolution, DRL obtains the best detection performance on six of the nine OOD datasets and achieves the second-best detection performance on the rest three OOD datasets. Therefore, DRL outperforms the state-of-the-art pretrained methods on both datasets. The reasons include: (1) The pretrained methods can only access the label-discriminative representations that are only sensitive to labeling. (2) For a given pretrained network, DRL trains an auxiliary network to explore the complementary distribution-discriminative representations that are sensitive to OOD samples.

6.3.2.2 Comparison with Retraining Methods

We compare DRL with retraining methods in terms of AUROC, FPR(95) and Detection. The retraining methods retrain a pretrained network to improve the OOD sensitivity when OOD samples are unavailable in the training process. The retraining methods include JCL [37], CSI [39], SSL [38], DeConf-C [30] and MOS [34]. The settings of all the comparison methods follow the original ones. For a fair comparison, following the grouping method in MOS, we use K-Means clustering on feature representations to divide the training ID datasets into groups with similar concepts.

The comparison results are presented in Table 6.2. We observe that DRL achieves significant improvement (4.74% on CIFAR10 and 2.06% on Mini-Imagenet) over the other state-of-the-art retraining methods in terms of AUROC. Furthermore, DRL achieves significantly improved performance (14.20% on CIFAR10 and 5.29% on Mini-Imagenet) in terms of Detection. We also observe that DRL does not obtain the best results in terms of FPR(95). However, its performance is close to the best one with a narrow gap (2.9 on CIFAR10 and 0.7 on Mini-Imagenet). Therefore, the proposed DRL method outperforms the state-of-the-art retraining methods on both datasets in terms of the two metrics. The results show that exploring distribution-discriminative representations from information weakly related to labeling and integrating label- and distribution-discriminative representations form an effective strategy to detect OOD samples. The reason is that the distribution-discriminative representations coupled with the corre-

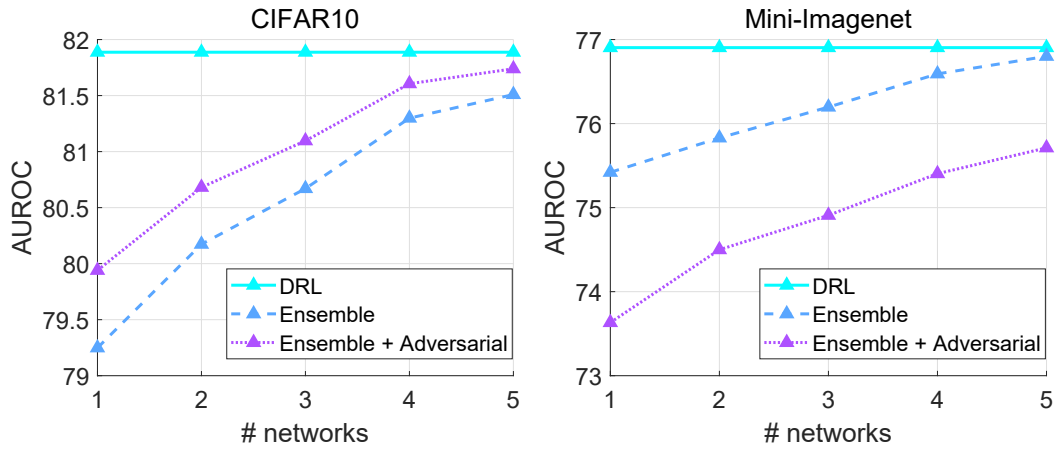


Fig. 6.3 DRL: OOD detection performance (compared with ensemble methods. The ensemble networks gradually increase independent pretrained networks with randomized initialization, and DRL only owns two depent networks, i.e, a pretrained and an auxiliary network. Each point indicates the average AUROC across all eight OOD datasets.

sponding label-discriminative representations contain more label-related information than any of them. The combined representation reduces the prediction confidence for an OOD sample owning minimum labeling-sensitive information and enhances the prediction confidence for an ID sample owning all the labeling information.

6.3.2.3 Comparison with Ensemble Methods

We compare DRL with ensemble methods integrating independent networks with different initialization parameters. Specifically, we utilize the traditional deep ensemble method which combines the output predictions by averaging softmax outputs. Besides, we apply the adversarial samples to the deep ensemble method [101]. Note that the traditional ensemble method degrades into the baseline method if the number of networks is one. All the methods apply multiple networks to explore more information from the ID samples to improve OOD sensitivity.

The comparison results are shown in Fig. 6.3. For the two ensemble methods, the OOD detection is improved as the number of networks increases on both CIFAR10 and Mini-Imagenet datasets. Furthermore, considering adversarial samples for the deep ensemble method results in improved and declined performance on CIFAR10 and Mini-Imagenet, respectively. Therefore, DRL obtains the best detection performance on both datasets, and the performance of the two ensemble methods is similar to that of DRL when the number of networks is five. Accordingly, DRL only containing two dependent networks outperforms the ensemble methods containing more independent networks. The results guarantee that the proposed DRL method is different from the ensemble method substantially. Further, DRL combines label- and distribution-discriminative representations to fetch the OOD-sensitive information from training ID samples.

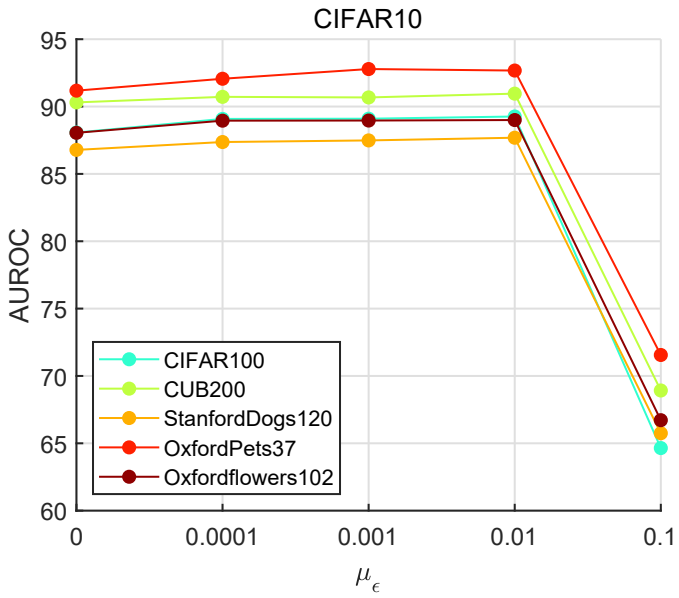


Fig. 6.4 DRL: Effect of the perturbation coefficient ϵ.
Each curve represents the detection performance on an OOD dataset, and each point indicates the AUROC for the corresponding perturbation coefficient expectation ϵ.

6.3.3 Parameter Analyses

6.3.4 The Effect of Perturbation Coefficient ϵ

Based on Eq. (6.11), we use a small perturbation coefficient ϵ to ensure positive weight $w(\mathbf{c})$. We also empirically guarantee that a small perturbation coefficient ϵ is more suitable for DRL. We select the value of the perturbation coefficient expectation ϵ in $\{0, 0.0001, 0.001, 0.01, 0.1\}$. Note that DRL becomes the deep ensemble method having two independent networks when $\epsilon = 0$.

The experimental results are summarized in Fig. 6.4. We observe that increasing ϵ can gradually improve the detection performance. Furthermore, the effect is drastically reduced when ϵ is sufficiently large ($\epsilon > 0.01$). DRL thus is more likely to use a perturbation coefficient that is small but larger than zero. According to the weight $w(\mathbf{c})$ measuring the dissimilarity between a label-discriminative representation $\mathbf{d}$ and an intermediate representation $\mathbf{z}$, ϵ is applied to ensure that $w(\mathbf{c})$ is positive. However, a large ϵ cannot satisfy the criteria. From another point of view, the expression of the distribution-discriminative representation $\mathbf{c}$ in Eq. (6.12) is similar to adversarial samples [36] where the coefficient ϵ affects the portion of the perturbation $\Sigma_{\mathcal{Z}}(\mathbf{c}) \times \mathbf{d} + \mu_{\mathcal{Z}}(\mathbf{c}) \times \mu_{\mathcal{Z}}(\mathbf{c}) \times \mathbf{d}$ on $\mu_{\mathcal{Z}}(\mathbf{c})$. Based on the idea of adversarial learning, a small coefficient is sufficient to change the predicted label of a test sample. Similarly, in DRL, a small ϵ is sufficient to obtain a distribution-discriminative representation $\mathbf{c}$ significantly differing from the corresponding label-discriminative representation, and a large ϵ can lead to an unstable distribution-discriminative representation which cannot capture the weakly label-related information from the training ID samples.

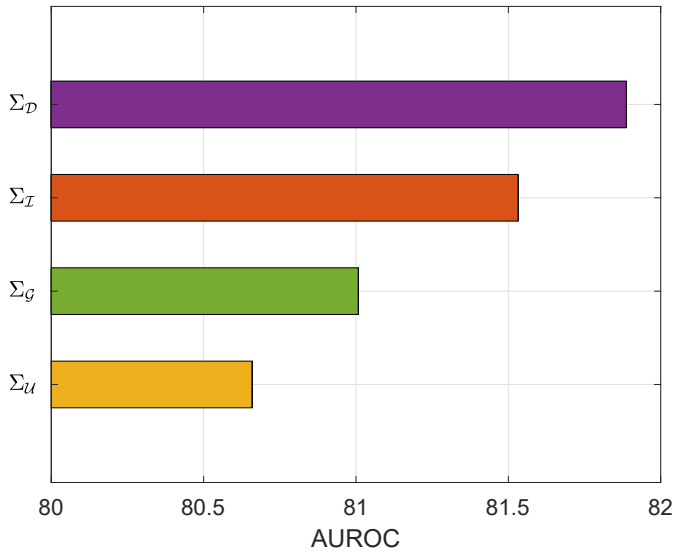


Fig. 6.5 DRL: Effect of the covariance matrix $\Sigma_{\mathcal{Z}}(\mathbf{c})$.
Each bar indicates the average AUROC across all eight OOD datasets.

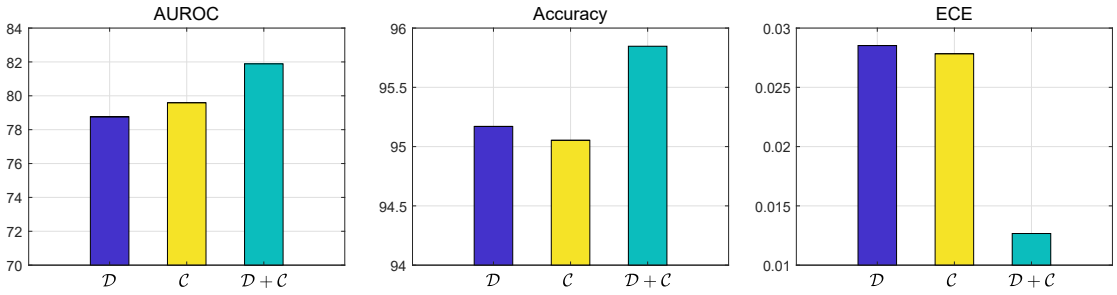


Fig. 6.6 DRL: Results of the ablation study.
The three subfigures report the results of AUROC, Accuracy and ECE, respectively. $\mathcal{D}$ indicates the label-discriminative representations from a pretrained network. $\mathcal{C}$ indicates the complementary distribution-discriminative representations from an auxiliary network. $\mathcal{D}+\mathcal{C}$ indicates the combination of label- and distribution-discriminative representations, i.e., the proposed DRL method.

6.3.5 The Effect of Covariance Matrix $\Sigma_{\mathcal{Z}}(\mathbf{c})$

We empirically verify that DRL is robust to the selection of covariance matrix $\Sigma_{\mathcal{Z}}(\mathbf{c})$. Furthermore, it is more suitable to use the covariance matrix $\Sigma_{\mathcal{D}}$ estimated from label-discriminative representations to construct the implicit constraint Eq. (6.14) for learning the corresponding distribution-discriminative representations, i.e., $\Sigma_{\mathcal{Z}}(\mathbf{c}) = \Sigma_{\mathcal{D}}$. We compare $\Sigma_{\mathcal{D}}$ with different covariance matrices $\Sigma_{\mathcal{Z}}(\mathbf{c})$, including the identity matrix $\Sigma_{\mathcal{I}}$, the Gaussian random matrix $\Sigma_{\mathcal{G}}$, and the uniform random matrix $\Sigma_{\mathcal{U}}$.

The experimental results are presented in Fig. 6.5. We observe that the performance gap ($[80.7, 81.9]$) among the four different covariance matrices is not significant. This represents that the covariance matrix $\Sigma_{\mathcal{Z}}(\mathbf{c})$ does not play an important role in learning distribution-discriminative representations. Besides, DRL is not sensitive to the choice of $\Sigma_{\mathcal{Z}}(\mathbf{c})$. Specifically, the two random matrices $\Sigma_{\mathcal{G}}$ and $\Sigma_{\mathcal{U}}$ obtain relatively poor

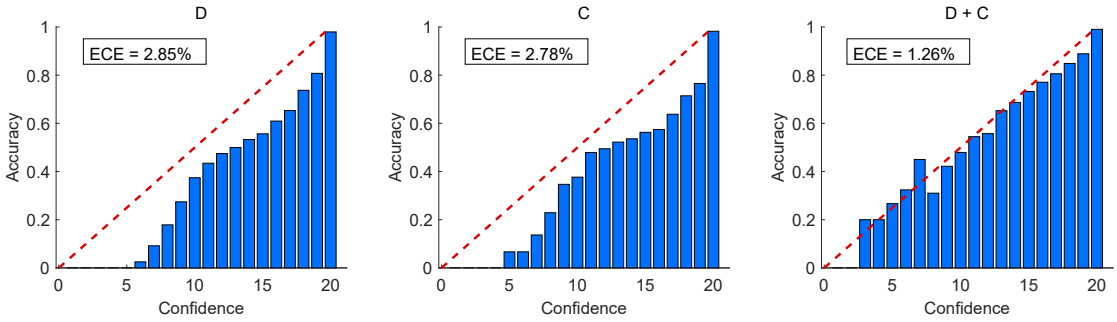


Fig. 6.7 DRL: Calibration results.
The confidence is equally divided into 20 intervals, and each bar represents the expected accuracy of samples whose confidence values are in the same interval. The red dotted diagonal indicates the perfect calibration.

performance, while the identity matrix achieves relatively better performance. It is because networks are more likely to decouple the features within an output representation. Therefore, the identity matrix, assuming all features are independent, is more appropriate than the two random matrices assuming random feature correlations. Adopting $\Sigma_{\mathcal{D}}$ achieves slightly better performance than $\Sigma_{\mathcal{I}}$. The reason is $\Sigma_{\mathcal{D}}$ is obtained from the pretrained network, and the pretrained and auxiliary networks own the same network structure and are trained on the same ID dataset. Therefore, $\Sigma_{\mathcal{D}}$ is the most appropriate estimation for $\Sigma_{\mathcal{Z}}(\mathbf{c})$. Note that the covariance matrix only represents the dispersion degree of representations. Therefore, assuming the pretrained and auxiliary networks have the same covariance matrix does not indicate their output representations are drawn from the same unknown distribution.

6.3.6 Ablation Study

We run a set of ablation study experiments to guarantee that label- and distribution-discriminative representations are indispensable for improving OOD detection performance. Note that DRL contains a pretrained network and an auxiliary network where the two networks learn label-discriminative representations ($\mathcal{D}$) and distribution-discriminative representations ($\mathcal{C}$), respectively. Furthermore, DRL applies the combination ($\mathcal{D}+\mathcal{C}$) of the two representations by Eq. (6.16) to detect OOD samples. Therefore, we compare the combination with the two different representations in terms of AUROC, Accuracy and ECE.

The results are shown in Fig. 6.6 and Fig. 6.7. They represent that solely exploiting the label-discriminative representations has a similar performance to solely exploiting the distribution-discriminative representations. Specifically, the distribution-discriminative representations obtain slightly better performance in detecting OOD samples but slightly worse performance in classifying ID samples than the label-discriminative representations. This is because the distribution-discriminative representation of an ID

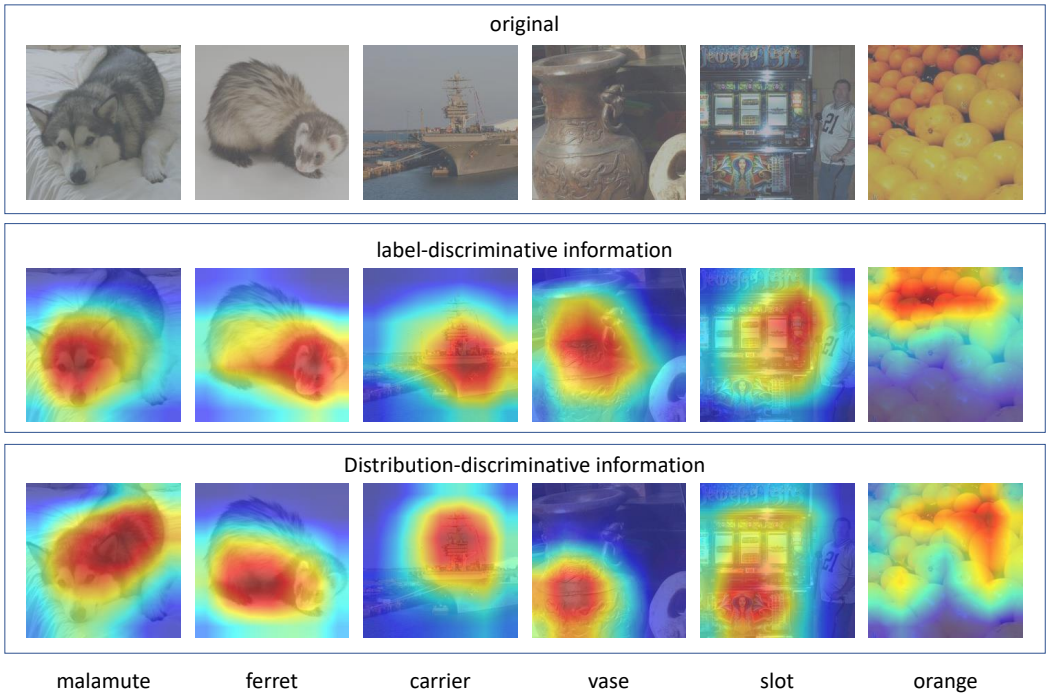


Fig. 6.8 DRL: Heat maps of Grad-CAM for label- and distribution-discriminative representations.
Red regions correspond to high scores for class, while blue regions correspond to low scores. The figure is best viewed in color.

sample has information that is weakly related to its labeling, and the weakly label-related information is more sensitive to OOD samples than the strongly-related information in the label-discriminative representations. From Fig. 6.7, we know that the pretrained and auxiliary networks are poorly-calibrated generating highly over-confident predictions. However, DRL is nearly perfectly calibrated. Therefore, considering both label- and distribution-discriminative representations, DRL achieves better performance on all metrics than any of the two components. DRL thus can take advantage of both components, which indicates that label- and distribution-discriminative representations are complementary to each other.

6.3.7 Visualization

We use Grad-Cam [102] to generate coarse localization maps that highlight the essential regions for learning label- and distribution-discriminative representations. Fig. 6.8 visually presents the heat maps of different classes of input samples on pretrained and auxiliary networks. We observe that the two networks focus on different regions for the same input sample. For instance, for the malamute, the pretrained network focuses on the head, but the auxiliary network focuses on the body. Based on the design principles of the two networks, the pretrained network is trained without any restriction on label-discriminative representations. However, the auxiliary network is trained with

an implicit constraint to ensure that the learned distribution-discriminative representations differ from the corresponding label-discriminative representations. Accordingly, we know that the pretrained network is more likely to extract the strongly label-related information from an ID sample to learn a label-discriminative representation (the head information in the malamute). Therefore, the auxiliary network tends to extract the weakly label-related information to learn a distribution-discriminative representation (the body information in the malamute). Based on the presented experimental results and the properties of the two networks, the pretrained and auxiliary networks extract strongly and weakly label-related information from inputs to learn representations, respectively. For an ID sample, the label- and distribution-discriminative representations that correspond to the same label are complementary.

6.4 Summary of This Chapter

In this chapter, we propose a Dual Representation Learning (DRL) method which combines both label- and distribution-discriminative representations to improve the OOD sensitivity. From the modeling perspective, a pretrained network learns label-discriminative representations that are strongly related to labeling, while an auxiliary network learns complementary distribution-discriminative representations that are weakly related to labeling. From the data perspective, the label- and distribution-discriminative representations are complementary in labeling for an ID sample and correspond to different labels for an OOD sample. According to the different informativeness properties of ID and OOD samples, DRL distinguishes ID and OOD samples according to the OOD scores estimated by integrating the two representations. We empirically demonstrate that DRL more effectively detects OOD samples than the state-of-the-art pretrained, retraining and ensemble methods across different datasets.

CHAPTER 7

Conclusion and Future Work

In this chapter, we first conclude the entire thesis, and then show several interesting future directions.

7.1 Conclusion

Limited training ID samples and unavailable OOD samples in the training process cause that networks could provided high-confidence predictions for OOD samples. Therefore, it is essential to consider the OOD detection performance of a pretrained network for adapting to real-world applications and avoid serious issues. OOD detection suffers from the following four important problems to address the over-confidence issue, and this thesis has addressed these questions with experimental and theoretical guarantees.

- **If OOD samples are unavailable, how to find the specific OOD samples that owning high-confidence prediction and are with semantic shift for a pretrained network?** In Chapter 3, we propose *fine-tuning discriminators by implicit generators* (FIG) to improve OOD sensitivity of a given pretrained discriminator which contains a network backbone and a linear classifier. FIG reveals the distributional vulnerability by the corresponding implicit generator inferred from a pretrained discriminator without extra training costs, draws OOD samples from the generator by a Langevin dynamic sampler, and patches the distributional vulnerability by penalizing the prediction confidence of these generated samples.

- **If OOD samples are generated or obtained from real-world datasets, how to involve them in the retraining process to balance ID classification and OOD detection?** In Chapter 4, we propose *supervision adaptation* (SA) approach to define the supervision information for OOD samples to adapt OOD to ID samples. We reveal the form of the supervision information by measuring the data relationships between ID samples and their labels in terms of the mixed space mutual information. Also, we estimate this supervision information in terms of multiple binary regression problems by considering the data correlations between the two kinds of samples.

- **If drawing OOD samples from generators is expensive, how to finetune a pretrained network with augmented ID samples to improve OOD sensitivity?** In Chapter 5, we propose *learning from cross-class vicinity distribution* (LCVD) method which makes the first attempt to generate OOD samples by augmenting ID samples. The cross-class vicinity distribution of ID samples explores OOD samples by considering the vicinity relations between samples of different classes. An OOD input is generated by linearly combining multiple ID inputs, which corresponds to a complementary label different from those labels of the constituent ID samples. Given a pretrained network, we can then finetune it to reject such OOD samples drawn from the cross-class vicinity distribution.

- **If retraining or fine-tuning are not allowed, how to learn an auxiliary network to capture OOD sensitive information discarded by a pretrained network?** In Chapter 6, we propose *Dual Representation Learning* (DRL) method combining both label- and distribution-discriminative representations to improve the OOD sensitivity. From the modeling perspective, a pretrained network learns label-discriminative representations that are strongly related to labeling, while an auxiliary network learns complementary distribution-discriminative representations that are weakly related to labeling. From the data perspective, the label- and distribution-discriminative representations are complementary in labeling for an ID sample and correspond to different labels for an OOD sample. Based on the different informativeness properties of ID and OOD samples, DRL distinguishes ID and OOD samples according to the OOD scores estimated by integrating the two representations.

7.2 Future Work

Although the methods proposed in the thesis have addressed some valuable questions in OOD detection, some issues are still open and should be further investigated.

- **Distinguishing the OOD samples with semantic shift and covariate shift**: The considered OOD detection task in this thesis focuses on detecting OOD samples with semantic shift from training ID samples (e.g., OOD samples are drawn from different classes) and, another related task OOD generalization task focuses on predicting classes for OOD samples with covariate shift (e.g., OOD samples are drawn from different domains with same classes). A reliable network should provide predicted labels for OOD samples with covariate shift and reject OOD samples with semantic shift. However, the gap between the OOD samples with semantic shift and covariate shift is still unknown.

- **Improving OOD sensitivity from data perspective:** The existing methods for improving OOD sensitivity of a pretrained network focus on involving OOD prior knowledge in the retraining or fine-tuning processes. The existing methods require modifying the training procedure and objective, which indicates that they improve OOD sensitivity from a model perspective. Recall that the main causes of the over-confidence issue include limited training ID samples and unavailable OOD samples. Therefore, improving OOD sensitivity from a data perspective is more straightforward and effective because the data characteristics cause the over-confidence issue. This motivates the future task of re-sampling and re-weighting the training ID samples to amend the training distribution.

- **Detecting OOD samples for tabular data:** This thesis focuses on detecting OOD samples for image data. When the OOD detection task comes to tabular data, the existing methods may not be applied. This is because tabular data differing from image data own explicit and implicit feature correlations. For example, a tabular instance with all values in normal ranges could be OOD due to the rare feature correlation patterns. For adapting OOD detection methods to tabular data, we should explore the feature correlations to capture the OOD-sensitive information.

REFERENCES

[1] 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.

[2] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," in *5th International Conference on Learning Representations*, 2017, pp. 1–15.

[3] Z. Allen-Zhu, Y. Li, and Y. Liang, "Learning and generalization in overparameterized neural networks, going beyond two layers," in *Advances in Neural Information Processing Systems 32*, 2019, pp. 6155–6166.

[4] A. M. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 427–436.

[5] J. Yang, K. Zhou, Y. Li, and Z. Liu, "Generalized out-of-distribution detection: A survey," *CoRR*, pp. 1–20, 2021.

[6] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," in *5th International Conference on Learning Representations*, 2017, pp. 1–12.

[7] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2242–2251.

[8] D. Amodei, C. Olah, J. Steinhardt, P. F. Christiano, J. Schulman, and D. Mané, "Concrete problems in AI safety," *CoRR*, pp. 1–29, 2016.

[9] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," *J. Mach. Learn. Res.*, vol. 9, no. 86, pp. 2579–2605, 2008.

[10] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *Advances in Neural Information Processing Systems 30*, 2017, pp. 5574–5584.

[11] H. Touvron, A. Vedaldi, M. Douze, and H. Jégou, "Fixing the train-test resolution discrepancy: Fixefficientnet," in *CoRR*, vol. abs/2003.08237, 2020, pp. 1–5.

[12] Y. Bengio, A. C. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, 2013.

[13] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *6th International Conference on Learning Representations*, 2018, pp. 1–13.

[14] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation strategies from data," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 113–123.

[15] A. Bendale and T. E. Boult, "Towards open set deep networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1563–1572.

[16] A. Malinin and M. J. F. Gales, "Predictive uncertainty estimation via prior networks," in *Advances in Neural Information Processing Systems 31*, 2018, pp. 7047–7058.

[17] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning From Theory to Algorithms*. Cambridge University Press, 2014.

[18] V. N. Vapnik, "Statistical learning theory," 1998.

[19] P. Y. Simard, Y. LeCun, J. S. Denker, and B. Victorri, "Transformation invariance in pattern recognition - tangent distance and tangent propagation," *Neural Networks: Tricks of the Trade*, pp. 235–269, 2012.

[20] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, "Deep variational information bottleneck," in *5th International Conference on Learning Representations*, 2017, pp. 1–19.

[21] 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 *6th International Conference on Learning Representations*, 2018, pp. 1–19.

[22] H. Zhang, A. Li, J. Guo, and Y. Guo, "Hybrid models for open set recognition," in *Proceedings of the European Conference on Computer Vision*, 2020, pp. 102–117.

[23] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *CoRR*, vol. abs/1503.02531, 2015.

[24] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," in *6th International Conference on Learning Representations*, 2018, pp. 1–27.

[25] J. van Amersfoort, L. Smith, A. Jesson, O. Key, and Y. Gal, "Improving deterministic uncertainty estimation in deep learning for classification and regression," in *CoRR*, 2021, pp. 1–16.

[26] K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," in *Advances in Neural Information Processing Systems 31*, 2018, pp. 7167–7177.

[27] E. Zisselman and A. Tamar, "Deep residual flow for out of distribution detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 991–14 000.

[28] C. S. Sastry and S. Oore, "Detecting out-of-distribution examples with gram matrices," in *Proceedings of the 37th International Conference on Machine Learning*, 2020, pp. 8491–8501.

[29] W. Liu, X. Wang, J. D. Owens, and Y. Li, "Energy-based out-of-distribution detection," in *Advances in Neural Information Processing Systems 33*, no. 1–12, 2020.

[30] Y. Hsu, Y. Shen, H. Jin, and Z. Kira, "Generalized ODIN detecting out-of-distribution image without learning from out-of-distribution data," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 948–10 957.

[31] P. Bevandic, I. Kreso, M. Orsic, and S. Segvic, "Simultaneous semantic segmentation and outlier detection in presence of domain shift," in *Pattern Recognition - 41st DAGM German Conference*, 2019, pp. 33–47.

[32] H. Blum, P. Sarlin, J. I. Nieto, R. Siegwart, and C. Cadena, "Fishyscapes: A benchmark for safe semantic segmentation in autonomous driving," in *IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 2403–2412.

[33] Z. Liu, Z. Wang, P. P. Liang, R. Salakhutdinov, L. Morency, and M. Ueda, "Deep Gamblers: Learning to abstain with portfolio theory," in *Advances in Neural Information Processing Systems 32*, 2019, pp. 10 622–10 632.

[34] R. Huang and Y. Li, "MOS: towards scaling out-of-distribution detection for large semantic space," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8710–8719.

[35] D. Hendrycks, M. Mazeika, and T. G. Dietterich, "Deep anomaly detection with outlier exposure," in *7th International Conference on Learning Representations*, 2019, pp. 1–18.

[36] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *3rd International Conference on Learning Representations*, 2015, pp. 1–11.

[37] K. Lee, H. Lee, K. Lee, and J. Shin, "Training confidence-calibrated classifiers for detecting out-of-distribution samples," in *6th International Conference on Learning Representations*, 2018, pp. 1–16.

[38] D. Hendrycks, M. Mazeika, S. Kadavath, and D. Song, "Using self-supervised learning can improve model robustness and uncertainty," in *Advances in Neural Information Processing Systems 32*, 2019, pp. 15 637–15 648.

[39] J. Tack, S. Mo, J. Jeong, and J. Shin, "CSI: Novelty detection via contrastive learning on distributionally shifted instances," no. 1–14, 2020.

[40] M. M. Christiansen and K. R. Duffy, "Guesswork, large deviations, and shannon entropy," *IEEE Trans. Inf. Theory*, vol. 59, no. 2, pp. 796–802, 2013.

[41] B. Poole, S. Ozair, A. van den Oord, A. Alemi, and G. Tucker, "On variational bounds of mutual information," in *Proceedings of the 36th International Conference on Machine Learning*, 2019, pp. 5171–5180.

[42] A. M. Saxe, Y. Bansal, J. Dapello, M. Advani, A. Kolchinsky, B. D. Tracey, and D. D. Cox, "On the information bottleneck theory of deep learning," in *6th International Conference on Learning Representations*, 2018, pp. 1–27.

[43] G. Shalev, Y. Adi, and J. Keshet, "Out-of-distribution detection using multiple semantic label representations," in *Advances in Neural Information Processing Systems 31*, 2018, pp. 7386–7396.

[44] 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*, 2017, pp. 1321–1330.

[45] T. Salimans, A. Karpathy, X. Chen, and D. P. Kingma, "Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications," in *5th International Conference on Learning Representations*, 2017, pp. 1–10.

[46] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," in *Advances in Neural Information Processing Systems 31*, 2018, pp. 10 236–10 245.

[47] Y. Bengio, E. Laufer, G. Alain, and J. Yosinski, "Deep generative stochastic networks trainable by backprop," in *Proceedings of the 31nd International Conference on Machine Learning*, 2014, pp. 226–234.

[48] M. I. Belghazi, A. Baratin, S. Rajeswar, S. Ozair, Y. Bengio, R. D. Hjelm, and A. C. Courville, "Mutual information neural estimation," in *Proceedings of the 35th International Conference on Machine Learning*, 2018, pp. 530–539.

[49] "A tutorial on energy-based learning," *Predicting structured data*, vol. 1, no. 0, pp. 1–59, 2006.

[50] M. Welling and Y. W. Teh, "Bayesian learning via stochastic gradient langevin dynamics," in *Proceedings of the 28nd International Conference on Machine Learning*, 2011, pp. 681–688.

[51] W. Grathwohl, K. Wang, J. Jacobsen, D. Duvenaud, M. Norouzi, and K. Swersky, "Your classifier is secretly an energy based model and you should treat it like one," in *8th International Conference on Learning Representations*, 2020, pp. 1–23.

[52] R. Bardenet, A. Doucet, and C. C. Holmes, "On Markov Chain Monte Carlo methods for tall data," *J. Mach. Learn. Res.*, vol. 18, pp. 1–47, 2017.

[53] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Comput.*, vol. 14, no. 8, pp. 1771–1800, 2002.

[54] Y. Du and I. Mordatch, "Implicit generation and modeling with energy based models," in *Advances in Neural Information Processing Systems 32*, 2019, pp. 3603–3613.

[55] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *6th International Conference on Learning Representations*, 2018, pp. 1–23.

[56] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, 2014, pp. 2672–2680.

[57] G. Pereyra, G. Tucker, J. Chorowski, L. Kaiser, and G. E. Hinton, "Regularizing neural networks by penalizing confident output distributions," in *5th International Conference on Learning Representations*, 2017, pp. 1–11.

[58] T. Wu and D. F. Gleich, "Multiway Monte Carlo method for linear systems," *SIAM J. Sci. Comput.*, vol. 41, no. 6, pp. 3449–3475, 2019.

[59] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[60] A. Krizhevsky, "Learning multiple layers of features from tiny images," Tech. Rep., 2009.

[61] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li, "Imagenet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.

[62] F. Yu, Y. Zhang, S. Song, A. Seff, and J. Xiao, "LSUN: construction of a large-scale image dataset using deep learning with humans in the loop," *CoRR*, vol. abs/1506.03365, 2015.

[63] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[64] G. Griffin, A. Holub, and P. Perona, "The caltech 256," Tech. Rep., 2006.

[65] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proceedings of the European Conference on Computer Vision*, vol. 8693, 2014, pp. 740–755.

[66] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[67] N. Ma, X. Zhang, H. Zheng, and J. Sun, "Shufflenet V2: practical guidelines for efficient CNN architecture design," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 122–138.

[68] 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. 2261–2269.

[69] K. Vladimir, D. Panchenko, and F. Lozano, "Bounding the generalization error of convex combinations of classifiers: balancing the dimensionality and the margins," *The Annals of Applied Probability*, vol. 13, no. 1, pp. 213–252, 2003.

[70] D. Barber and F. V. Agakov, "The IM algorithm: A variational approach to information maximization," in *Advances in Neural Information Processing Systems 16*, 2003, pp. 201–208.

[71] P. L. Bartlett and S. Mendelson, "Rademacher and gaussian complexities: Risk bounds and structural results," *J. Mach. Learn. Res.*, vol. 3, pp. 463–482, 2002.

[72] M. Sugiyama, T. Suzuki, and T. Kanamori, "Density-ratio matching under the bregman divergence: a unified framework of density-ratio estimation," *Annals of the Institute of Statistical Mathematics*, vol. 64, no. 5, pp. 1009–1044, 2012.

[73] P. Becker, O. Arenz, and G. Neumann, "Expected information maximization: Using the i-projection for mixture density estimation," in *8th International Conference on Learning Representations*, 2020, pp. 1–16.

[74] C. M. Bishop, *Pattern Recognition and Machine Learning*. springer, 2006.

[75] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.

[76] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36nd International Conference on Machine Learning*, 2019, pp. 6105–6114.

[77] P. Xu, K. A. Ehinger, Y. Zhang, A. Finkelstein, S. R. Kulkarni, and J. Xiao, "Turkergaze: Crowdsourcing saliency with webcam based eye tracking," *CoRR*, vol. abs/1504.06755, 2015.

[78] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3730–3738.

[79] X. Peng, B. Usman, N. Kaushik, J. Hoffman, D. Wang, and K. Saenko, "Visda: The visual domain adaptation challenge," *CoRR*, vol. abs/1710.06924, 2017.

[80] D. Li, Y. Yang, Y. Song, and T. M. Hospedales, "Deeper, broader and artier domain generalization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5543–5551.

[81] L. Yuan, F. E. H. Tay, G. Li, T. Wang, and J. Feng, "Revisiting knowledge distillation via label smoothing regularization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3902–3910.

[82] O. Chapelle, J. Weston, L. Bottou, and V. Vapnik, "Vicinal risk minimization," in *Advances in Neural Information Processing Systems 13*, 2000, pp. 416–422.

[83] P. Goyal, P. Dollár, R. B. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, "Accurate, large minibatch SGD: training imagenet in 1 hour," *CoRR*, 2017.

[84] T. Ishida, G. Niu, W. Hu, and M. Sugiyama, "Learning from complementary labels," in *Advances in Neural Information Processing Systems 30*, 2017, pp. 5639–5649.

[85] S. M. Kakade, K. Sridharan, and A. Tewari, "On the complexity of linear prediction: Risk bounds, margin bounds, and regularization," in *Advances in Neural Information Processing Systems 21*, 2008, pp. 793–800.

[86] S. Wang and Z. Xu, "New approximation algorithms for weighted maximin dispersion problem with box or ball constraints," *J. Optim. Theory Appl.*, vol. 190, no. 2, pp. 524–539, 2021.

[87] M. Nägele and R. Zenklusen, "A new dynamic programming approach for spanning trees with chain constraints and beyond," in *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, 2019, pp. 1550–1569.

[88] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD birds-200-2011 dataset," Tech. Rep., 2011.

[89] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei, "Novel dataset for fine-grained image categorization," in *Proc. CVPR Workshop on Fine-Grained Visual Categorization*, no. 1–2, 2011.

[90] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V. Jawahar, "Cats and dogs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, no. 3498–3505, 2012.

[91] M. Nilsback and A. Zisserman, "A visual vocabulary for flower classification," in *IEEE Computer Society Conference on Computer Vision and Pattern*, no. 1447–1454, 2006.

[92] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi, "Describing textures in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, no. 3606–3613, 2014.

[93] D. Hendrycks, K. Lee, and M. Mazeika, "Using pre-training can improve model robustness and uncertainty," in *Proceedings of the 36th International Conference on Machine Learning*, no. 2712–2721, 2019.

[94] J. Gonzalez-Lopez, S. Ventura, and A. Cano, "Distributed selection of continuous features in multilabel classification using mutual information," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 31, no. 7, pp. 2280–2293, 2020.

[95] M. Federici, A. Dutta, P. Forré, N. Kushman, and Z. Akata, "Learning robust representations via multi-view information bottleneck," in *8th International Conference on Learning Representations*, 2020, pp. 1–26.

[96] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *Proceedings of the 33nd International Conference on Machine Learning*, 2016, pp. 1050–1059.

[97] A. Kulesza and B. Taskar, "Determinantal point processes for machine learning," *Found. Trends Mach. Learn.*, vol. 5, no. 2-3, pp. 123–286, 2012.

[98] N. Anari, S. O. Gharan, and A. Rezaei, "Monte carlo markov chain algorithms for sampling strongly rayleigh distributions and determinantal point processes," in *Proceedings of the 29th Conference on Learning Theory*, no. 103–115, 2016.

[99] A. Kulesza and B. Taskar, "Learning determinantal point processes," in *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, 2011, pp. 419–427.

[100] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30*, 2017, pp. 5998–6008.

[101] I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in Neural Information Processing Systems 30*, 2017, pp. 6402–6413.

[102] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 336–359, 2020.

LIST OF RESEARCH RESULTS

1. Z.-L. Zhao, L.-B. Cao and K.-Y. Lin. Revealing the Vulnerabilities of Discriminators by Implicit Generators. TPAMI, 2022.

2. Z.-L. Zhao, L.-B. Cao and K.-Y. Lin. Out-of-distribution Detection by Cross-class Vicinity Distribution of In-distribution Data. TNNLS, 2023.

3. Z.-L. Zhao, L.-B. Cao. Supervision Adaptation Balances In-Distribution Generalization and Out-of-Distribution Detection. submitted to TPAMI, 2021, TPAMI-2021-06-0908, major revision.

4. Z.-L. Zhao, L.-B. Cao and K.-Y. Lin. Dual Representation Learning for Out-of-Distribution Detection. submitted to TMLR, 2023.

5. Z.-L. Zhao, L.-B. Cao and Y.-Y. Wan. Coupling Online-Offline Learning for Multi-Distribution Data Streams. submitted to Information Science, 2023, INS-D-23-682, major revision.

6. Z.-L. Zhao, L.-B. Cao and C.-D. Wang. Gray Learning from Non-IID Data with Out-of-distribution Samples. submitted to TNNLS, 2022, TNNLS-2023-P-26217.