RCL: Reliable Continual Learning for Unified Failure Detection

Fei Zhu[1], Zhen Cheng[2,3], Xu-Yao Zhang[2,3], Cheng-Lin Liu[2,3], Zhaoxiang Zhang[1,2,3,4]*
[1]Centre for Artificial Intelligence and Robotics, HKISI-CAS
[2]State Key Laboratory of Multimodal Artificial Intelligence Systems, CASIA
[3]School of Artificial Intelligence, UCAS
[4]Shanghai Artificial Intelligence Laboratory
{zhufei2018, chengzhen2019, zhaoxiang.zhang}@ia.ac.cn, {xyz, liucl}@nlpr.ia.ac.cn

Abstract

Deep neural networks are known to be overconfident for what they don't know in the wild, which is undesirable for decision-making in high-stakes applications. Despite quantities of existing works, most of them focus on detecting out-of-distribution (OOD) samples from unseen classes, while ignoring large parts of relevant failure sources like misclassified samples from known classes. In particular, recent studies reveal that prevalent OOD detection methods are actually harmful for misclassification detection (MisD), indicating that there seems to be a tradeoff between those two tasks. In this paper, we study the critical yet under-explored problem of unified failure detection, which aims to detect both misclassified and OOD examples. Concretely, we identify the failure of simply integrating learning objectives of misclassification and OOD detection, and show the potential of sequence learning. Inspired by this, we propose a reliable continual learning paradigm, whose spirit is to equip the model with MisD ability first, and then improve the OOD detection ability without degrading the already adequate MisD performance. Extensive experiments demonstrate that our method achieves strong unified failure detection performance. The code is available at `https://github.com/Impression2805/RCL`.

1. Introduction

Modern deep neural networks have made rapid progress in many fields [31, 35, 43, 44, 49, 52, 66]. Nevertheless, prediction errors are still inevitable due to the imperfect generalization ability in complex and open environments. Therefore, the model is expected to identify "what it does not know", i.e., being aware of when it is likely to be wrong and avoiding catastrophic decisions, especially in safety-critical scenarios such as medical diagnostics [38], autonomous driving [65] and robot vision [39]. Existing work has tackled this problem from two perspectives: semantic (new-class) out-of-distribution (OOD) detection and misclassification detection (MisD). Specifically, ***OOD detection*** determines whether an input is from known classes or unknown class [4, 5, 18–20, 27, 29, 30, 34, 40, 42, 45], while ***MisD*** aims to reject commonly existed misclassified data from known classes [8, 18, 36, 61–64].

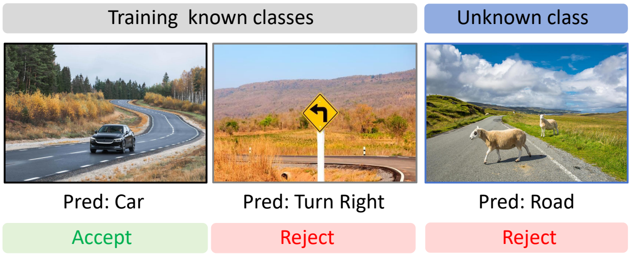

Figure 1. Illustration of typical failure cases arise when deploying models in real-world applications: misclassified example from known classes (e.g., recognize `Turn Left` as `Turn Right`), and semantic-shifted OOD example from unknown classes (e.g., `sheep`). A unified failure detector should accept correctly recognized examples and reject both misclassified and OOD examples.

Remarkably, OOD detection and MisD aim to achieve the same goal of detecting wrong predictions of a classifier. However, they are studied and evaluated individually nowadays, which is not consistent with the requirement in real-world applications where both misclassified examples and unknown class data should be detected and rejected [22]. As an example, as shown in Fig. 1, an autonomous driving car can not classify one type of traffic sign as another, or recognize an unknown animal as the road. Actually, when going back to the last century, the problems of OOD detection and MisD have been formulated and studied together [10, 12] as ambiguity rejection [6] and distance rejection [12], respectively. Besides, the baseline for reliable prediction in deep learning era also discussed those two tasks

*Corresponding author.

together [18]. Therefore, more recently, Jaeger et al. called for a unified evaluation, and formulated the unified ***failure detection*** problem [22] that deals with OOD detection and MisD simultaneously.

Although well reasoned, developing effective failure detection methods is more difficult than expected, as recent works [2, 3, 22–24] reveal that the simple maximum softmax probability (MSP) baseline [18] still performs the best when detecting OOD and misclassified examples jointly. In this paper, we aim to propose an approach that performs well under the realistic and challenging failure detection setup. To achieve this goal, a straightforward solution is to combine methods from OOD detection and MisD fields, e.g., optimizing the model with both CRL [36] and LogitNorm [46] losses. However, empirical results show that this simple strategy is useless. We identify that misclassified and OOD examples might have different levels of sensitivity, leading to conflict during training. Furthermore, we find that such conflicts can be mitigated by decoupling the two learning objectives and learning them in sequence. Inspired by those observations, we discard the joint learning strategy and creatively propose the ***Reliable Continual Learning*** (***RCL***) paradigm for failure detection. That is, we first equip the model with MisD ability and then continually tune it to further incorporate OOD detection ability. Particularly, in most cases, we might already have a pre-trained classifier with good reliability on some failure cases, and the RCL paradigm allows us to enhance the reliability of a newly encountered failure case without re-training from scratch.

Technically, the proposed RCL involves fine-tuning a pre-trained model e.g., a classifier with good MisD ability. The major challenge is how to retain the original equipped reliability knowledge while adapting model weights to enhance another requirement, e.g., OOD detection. On the one hand, a substantial change in the model state would considerably diminish the reliability knowledge already have. On the other hand, a minor update leads to inferior ability when dealing with new failure cases. To address this challenge, we view the knowledge stored in the given classifier as a well-defined region in the parameter space [16], and emphasize the preservation of important parameters during the tuning process. Specifically, Fisher Information [33] is adopted to estimate the importance of each parameter. In addition, we also ensemble parameter spaces in the tuning trajectory to further mitigate the forgetting of reliability knowledge. Moreover, we show that tuning can be performed on some selective informative layers in a deep neural network (DNN), which is much more efficient.

Our contributions are summarized as follows:

- We study the challenging unified failure detection problem, and propose a reliable continual learning paradigm to address the tradeoff between OOD detection and MisD.
- To improve the OOD detection performance while best preserving the existing MisD knowledge, we selectively regularize the parameter changes during the tuning process and gradually ensemble the yielded parameter space.
- Extensive experiments demonstrate that the proposed method can significantly and consistently help to detect both misclassified and OOD examples.

2. Related Work

Out-of-distribution detection. Score function based methods focus on designing proper confidence scores given a pre-trained classifier. Hendrycks et al. [18] established the baseline that directly leverages MSP score for detecting OOD examples. Later, many other score functions have been proposed, such as ODIN [29], Mahalanobis distance [27], Energy [30], ViM [45], MaxLogit [20], ReAct [41] and KNN [42]. Training regularization based methods [5, 11, 19, 34, 40, 46] directly learn a classifier that can separate OOD data from in-distribution (InD) samples. For example, LogitNorm [46] keeps a constant norm of logits during training to improve the OOD detection ability.

Misclassification detection. MisD [14, 18] focuses on distinguishing misclassified examples from correctly classified ones in training classes. The MSP score [14, 18] is also the baseline method for MisD in deep learning era. ConfidNet [8] and SS [32] learn the true class probability via an auxiliary model trained on the misclassified samples. CRL [36] learns the correctness ranking based on the historical correct rate during training. A recent work [61, 64] analyzes relations among confidence calibration, OOD detection and MisD from the perspective of proper scoring rules [15] and Bayes optimal reject rules [6], and demonstrates that seeking flat minima is helpful for rejecting misclassified examples. Besides, the effectiveness of outlier data has been verified for improving MisD [63].

3. Preliminary and Analysis

Notations. Considering a K-class classification task, let $(X_{\text{in}}, Y_{\text{in}}) \in \mathcal{X} \times \mathcal{Y}$ be jointly distributed random variables, where $\mathcal{X} \subset \mathbb{R}^d$ denotes the input space and $\mathcal{Y}$ is the label space. A labeled dataset $\mathcal{D}_{\text{train}} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ contains samples drawn *i.i.d.* from $(X_{\text{in}}, Y_{\text{in}})$. Assuming that $f_k(\boldsymbol{x})$ is the logits output of a DNN classifier f with respect to class k, the predicted class of an input $\boldsymbol{x}$ is $\underset{k=1,\dots,K}{\arg\max}\, p_k(\boldsymbol{x})$, in which $p_k(\boldsymbol{x}) = \exp(f_k(\boldsymbol{x}))/\sum_{k'=1}^{K}\exp(f_{k'}(\boldsymbol{x}))$ is the probability of $\boldsymbol{x}$ belonging to class k. The common reliability metrics [14, 18, 64] are the area under the risk-coverage curve (AURC), the probability that a negative example is predicted as a positive one when the true positive rate is as high as 95% (FPR95) and the area under the receiver operating characteristic curve (AUROC).

3.1. Classical Problem Definition

OOD detection. Formally, we have a joint InD $\mathcal{D}_{X_{\text{in}}Y_{\text{in}}}$ and an joint OOD $\mathcal{D}_{X_{\text{out}}Y_{\text{out}}}$, where $X_{\text{out}} \in \mathcal{X}$, but $Y_{\text{out}} \notin \mathcal{Y}$ (i.e., unknown new classes). At inference stage, we encounter a mixture of InD and OOD joint distributions $\mathcal{D}_{XY} = \pi_{\text{in}}\mathcal{D}_{X_{\text{in}}Y_{\text{in}}} + (1-\pi_{\text{in}})\mathcal{D}_{X_{\text{out}}Y_{\text{out}}}$, and can only observe the marginal distribution $\mathcal{D}_X = \pi_{\text{in}}\mathcal{D}_{X_{\text{in}}} + (1-\pi_{\text{in}})\mathcal{D}_{X_{\text{out}}}$, where $\pi_{\text{in}} \in (0,1)$ is an unknown prior probability [13, 53]. For a classifier f trained on $\mathcal{D}_{\text{train}}$, given a score function s and a predefined threshold δ, OOD detection can be performed based on a decision function $g : \mathcal{X} \to \{0,1\}$ such that for any test data $\boldsymbol{x}$ drawn from $\mathcal{D}_X$, we have: $g(\boldsymbol{x}) = 1$ (outlier, $\boldsymbol{x} \in \mathcal{D}_{X_{\text{out}}}$) if $s(\boldsymbol{x}) \geq \delta$ and $g(\boldsymbol{x}) = 0$ (inlier, $\boldsymbol{x} \in \mathcal{D}_{X_{\text{in}}}$) otherwise.

Misclassification detection. This task focuses on distinguishing incorrect ($\mathcal{D}^{\times}_{X_{\text{in}}}$) from correct ($\mathcal{D}^{\checkmark}_{X_{\text{in}}}$) predictions based on their confidence ranking [18, 36, 64]. For a classifier f trained on $\mathcal{D}_{\text{train}}$, given the score function s and a predefined threshold δ, MisD can be performed based on the following decision function $g : \mathcal{X} \to \{0,1\}$ such that for any test data $\boldsymbol{x}$ drawn from $\mathcal{D}_{X_{\text{in}}}$, we have: $g(\boldsymbol{x}) = 1$ (misclassified, $\boldsymbol{x} \in \mathcal{D}^{\times}_{X_{\text{in}}}$) if $s(\boldsymbol{x}) \geq \delta$ and $g(\boldsymbol{x}) = 0$ (correctly classified, $\boldsymbol{x} \in \mathcal{D}^{\checkmark}_{X_{\text{in}}}$) otherwise.

3.2. Unified Failure Detection

Problem statement of failure detection. Unified failure detection distinguishes both misclassified examples ($\mathcal{D}^{\times}_{X_{\text{in}}}$) and OOD examples ($\mathcal{D}_{X_{\text{out}}}$) from correctly classified examples ($\mathcal{D}^{\checkmark}_{X_{\text{in}}}$). For a classifier f trained on $\mathcal{D}_{\text{train}}$, given the score function s and a predefined threshold δ, failure detection can be performed based on the following decision function $g : \mathcal{X} \to \{0,1\}$ such that for any test data $\boldsymbol{x}$ drawn from the mixed marginal distribution $\mathcal{D}_X = \pi_{\text{in}}\mathcal{D}_{X_{\text{in}}} + (1-\pi_{\text{in}})\mathcal{D}_{X_{\text{out}}}$, we have:

$$g(\boldsymbol{x}) = \begin{cases} 1 \ (\boldsymbol{x} \in \mathcal{D}^{\times}_{X_{\text{in}}} \cup \mathcal{D}_{X_{\text{out}}}) & \text{if} \ \ s(\boldsymbol{x}) \geq \delta \\ 0 \ (\boldsymbol{x} \in \mathcal{D}^{\checkmark}_{X_{\text{in}}}) & \text{if} \ \ s(\boldsymbol{x}) < \delta \end{cases}. \quad (1)$$

3.3. Joint Learning or Sequence Learning?

Recent studies [22, 24, 63, 64] have verified that existing OOD detection methods are often harmful for detecting misclassified examples from InD, and MSP baseline performs the best for unified failure detection. Besides, we find that some MisD methods such as CRL [36] and FMFP [61, 64] can improve OOD detection on simple datasets like CIFAR-10, the improvement is marginal on more challenging datasets like CIFAR-100. Therefore, unified failure detection still remains challenging. We ask a natural but unexplored question: *Can unified failure detection be achieved by jointly learning objectives of MisD and OOD detection?*

To answer the above question, we conduct experiments with representative training-time MisD method CRL [36] and OOD detection method LogitNorm (LN) [46]. Specifically, CRL regularizes the relationship between training examples with different difficulties as follows:

$$\mathcal{L}_{\text{CRL}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \max(0, -r(c_i, c_j)(\kappa_i - \kappa_j) + |c_i - c_j|), \quad (2)$$

where c_i is the proportion of correct prediction events of $\boldsymbol{x}_i$, κ_i denotes a confidence function, and $r(c_i, c_j) = 1$ if $c_i > c_j$, $r(c_i, c_j) = 0$ if $c_i = c_j$ and $r(c_i, c_j) = -1$ otherwise. LogitNorm enforces a constant vector norm on the logits:

$$\mathcal{L}_{\text{LN}} = -\log \frac{\exp(f_y/(\tau\|\boldsymbol{f}\|))}{\sum_{i=1}^{k}\exp(f_i/(\tau\|\boldsymbol{f}\|))}, \quad (3)$$

where τ denotes the temperature parameter that modulates the magnitude of the logits. Existing works have verified that CRL and LogitNorm are quite effective for improving MisD and OOD detection performance, respectively. The learning objectives of them can be easily integrated, and we expect that the ***joint learning*** schema can be effective for unified failure detection. Experiments are conducted on CIFAR benchmark [26] and the model is ResNet110 [17] trained with SGD using standard learning schedule and data augmentation following [61]. The hyperparameter $\tau = 0.04$ following [46].

Table 1. Jointly learning MisD and OOD detection objectives is useless for unified failure detection. The network is ResNet110.

Dataset	Method	MisD			OOD Detection	
		AURC↓	FPR95↓	AUROC↑	FPR95↓	AUROC↑
CIFAR-10	CE	8.96	45.72	90.43	39.54	89.86
	CRL	**6.56**	**21.86**	**93.61**	**29.74**	91.13
	CRL+LN	8.50	26.21	91.53	32.43	**91.86**
CIFAR-100	CE	90.38	52.07	84.80	70.14	73.29
	CRL	**76.93**	**41.40**	**86.92**	73.05	71.97
	CRL+LN	80.42	44.62	86.00	**69.60**	**73.97**

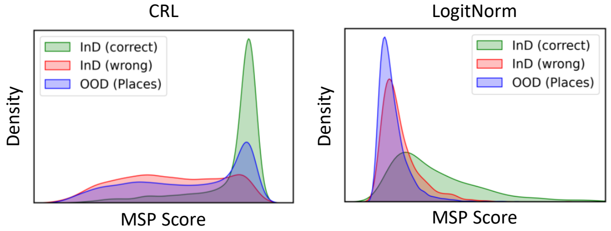

Figure 2. Distribution of InD (correct), InD (wrong) and OOD examples. The dataset is CIFAR-100 and the model is ResNet110.

Joint learning is ineffective for unified failure detection. As shown in Table 1, when combined with LogitNorm, the OOD detection ability of CRL is improved. However, the MisD performance decreases observably, even worse than the CE baseline. That is, the joint learning schema fails to improve both MisD and OOD detection ability, indicating the existing conflicts between those two learning objectives. Besides, we observe that for the CE baseline and CRL, the

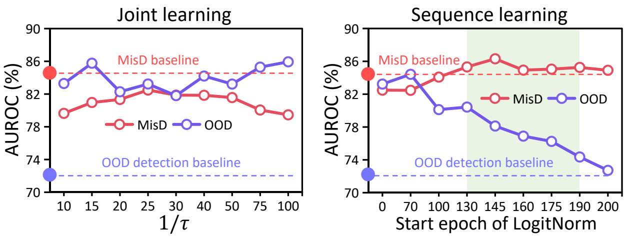

Figure 3. Comparison between joint and sequence learning of CRL and LogitNorm on CIFAR-100 with ResNet110. Improvements (over baseline) on both MisD and OOD detection performance can be observed in sequence learning (green region).

average confidence of OOD examples is higher than that of misclassified InD examples; while the relationship is on the contrary for LogitNorm. This can also be observed in Fig. 2. In conclusion, those results indicate that misclassified and OOD examples have different levels of sensitivity, which might also explain the lack of interoperability between existing MisD and OOD detection methods.

Sequence learning for unified failure detection. Based on the above observations and analysis, we argue that the learning objective of MisD and OOD detection should be decoupled during the training process. To this end, we propose a ***sequence learning*** schema as follows:

- Step 1: Train the model with CRL for T epochs;
- Step 2: Switch the learning objective from CRL to LogitNorm, and train the model for remaining epochs.

Fig. 3 compares those two learning paradigms. Specifically, for joint learning (CRL+LogitNorm), we conduct experiments with different values of temperature τ, which represents the strength of LogitNorm. For sequence learning (CRL⇒LogitNorm), we plot the performance with different start epoch T of LogitNorm. As shown in Fig. 3, the OOD detection performance of joint learning is remarkable, but the MisD performance is always worse than the CE baseline. Interestingly, both the MisD and OOD detection can be improved (over baseline) with sequence learning (the green region), which verifies our hypothesis about decoupling the learning objective of MisD and OOD detection when training a unified failure detector.

Limitations of sequence learning. Despite the positive effectiveness, sequence learning still has limitations. First, the performance is affected by the start epoch T, which is hard to choose beforehand, hindering its flexibility in real-world applications. Second, the unified failure detection performance is still less than satisfactory. For example, with sequence learning, the MisD is comparable to that of CRL, but the OOD detection performance is remarkably worse than LogitNorm. Inspired by the good and the bad of sequence learning, we propose a new paradigm to tackle the unified failure detection problem in Section 4.

4. Reliable Continual Learning

Overview of reliable continual learning paradigm. The analysis in Section 3.3 demonstrates that sequence learning is a promising direction for developing unified failure detectors. Moreover, real-world applications may require a classifier to be evolved to detect newly emerged failure sources without training from scratch. Motivated by this, we propose a ***reliable continual learning*** paradigm, which achieves the goal of unified failure detection by equipping a classifier with the ability to detect one failure case and then tuning it to further incorporate other reliable knowledge. In Fig. 4, we provide an illustration of the proposed paradigm and the details of our approach are presented below.

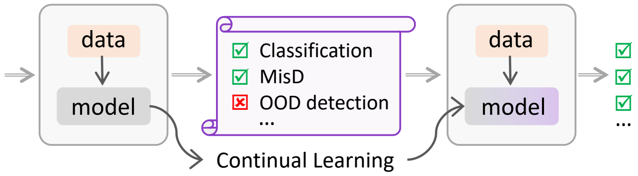

Figure 4. Illustration of reliable continual learning paradigm. During deployment, a classifier already has good performance on some aspects such as classification and MisD. When facing new failure cases like OOD examples, we continually update the model via reliable fine-tuning, without training from scratch.

4.1. Reliable Weight Consolidation

Fine-tuning is a commonly used schema to further learn new knowledge. Nonetheless, without any regularization, the classifier tends to forget already acquired reliability knowledge catastrophically when learning to detect new failures. For example, direct tuning with LogitNorm could enlarge the separation between InD and OOD data, however, this also ruins the separation between correctly classified and misclassified examples. Therefore, the remaining question is how to effectively tune the classifier while also keeping previous reliability knowledge. To ameliorate this issue, one can regularize the parameter updates so that they do not deviate too much from the original parameter space.

Intuitively, parameters in different layers should be regularized differently during the tuning process. Therefore, to retain existing knowledge, a proper way is to regularize the more influential parameters while updating those less influential ones to incorporate new knowledge. Inspired by continual learning [1, 25, 56, 57, 59, 60], we estimate the model parameters via Fisher Information [33], which describes the model's expected sensitivity to a change in parameters. Specifically, given a pretrained model, based on training data $\mathcal{D}_{\text{train}} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ or its subset, the Fisher information F can be computed as:

$$F = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_{\text{train}}}[\nabla^2 \log p(y|\boldsymbol{x};\theta^*)], \tag{4}$$

where θ^* is the weights of the pretrained model and $\ell =$

$-\log p(y|\boldsymbol{x};\theta^*)$ is the loss function. In practice, the Fisher information is only computed once before tuning the model, and we use approximation by only computing the diagonal elements. As for the loss function, it is better to reflect the already equipped reliability (e.g., the CRL loss in Eq. (2) if the correctness history has been recorded). Generally, we can just use the cross-entropy loss, since models trained by CE loss achieve a fairly strong baseline in MisD [3, 22, 64]. Then, a regularization is added to penalize the weight change during tuning as follows:

$$\mathcal{L}_{\text{RWC}} = \mathcal{L}_{\text{target}} + \lambda \sum\nolimits_i F_i(\theta_i - \theta_i^*), \tag{5}$$

where i is the index of each parameter of the model, and λ is the regularization weight to balance different losses. $\mathcal{L}_{\text{target}}$ is the learning objective of detecting new failures, e.g., the LogitNorm loss to detect OOD examples. We call the above method ***reliable weight consolidation*** (***RWC***).

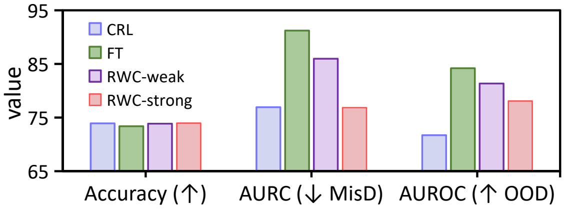

Figure 5. Given a model pre-trained with CRL, the classification accuracy is maintained during fine-tuning with LogitNorm loss. However, the MisD ability is ruined e.g., AURC (↓) dramatically increased. RWC-weak and RWC-strong preserve the MisD ability with different strengths. We use CIFAR-100 with ResNet110.

Discussion. The regularization term in Eq. (5) was first proposed in [25] and named as elastic weight consolidation (EWC), which aims to avoid a significant reduction of classification performance when learning different datasets continually. In this work, we demonstrate its effectiveness in failure detection. As shown in Fig. 5, the MisD performance is ruined remarkably, but the classification performance is still maintained without the regularization term. Therefore, different from [25] that aims to maintain classification accuracy, we focus on maintaining the knowledge of reliability. Without this regularization, the tuning process with target objective $\mathcal{L}_{\text{target}}$ will quickly result in catastrophic forgetting of already acquired reliability like MisD. In conclusion, the work of [25] learns different datasets or classes with the same optimizing objective, while we learn different objectives with the same dataset or classes.

4.2. Weight Space Interpolation

Tuning with the regularization loss in Section 4.1 leads to a series of models residing in generally good parameter regions for both MisD and OOD detection, as shown in Fig. 6. However, for a specific point in the tuning trajectory, it may prefer to detect one failure than the other, which indicates the diversity of knowledge encoded in the model at different tuning steps. Besides, the model would encounter unconstrained, various misclassified and OOD data.

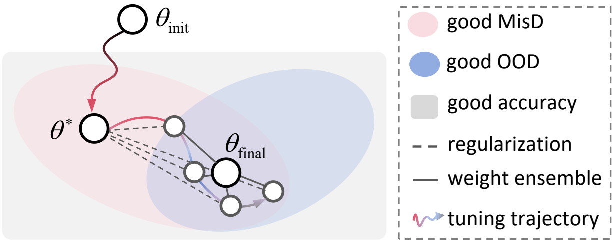

Figure 6. Illustration of the proposed simple yet effective reliable tuning procedure. A model with good MisD ability is tuned towards good OOD ability with weight regularization. Finally, multiple weight spaces in the tuning trajectory are interpolated.

Motivated by this, we propose a ***weight space interpolation*** (***WSI***) strategy to best leverage the diverse and rich uncertainty knowledge during the tuning trajectory. Formally, consider a tuning procedure with T training epochs, we can get a trajectory of models $P = \{\theta_t\}_{t=0}^T$, where θ_0 is the pretrained model and θ_t is the model after tuning the t-th epoch. We ensemble those intermediate models as follows:

$$\theta_t^{\text{WSI}} = \frac{\sum_{i=0}^{t-1}\alpha_i}{\sum_{i=0}^{t}\alpha_i} \cdot \theta_{t-1}^{\text{WSI}} + \frac{\alpha_t}{\sum_{i=0}^{t}\alpha_i} \cdot \theta_t, \tag{6}$$

where θ_0 is the same as θ^* and α_i denotes the contribution of each model θ_t. In this paper, we simply set $\alpha_i = 1$ for $i \in \{0, 1, ..., T\}$. Fig. 6 illustrates the ensemble process.

Discussion. Weight interpolation has been explored by other works for improving classification accuracy [21, 48]. For example, Ilharco et al. [21, 48] performed linearly interpolation between θ_0 and θ_{ft} to produce $\theta_{\text{final}} = (1-\alpha) \cdot \theta_0 + \alpha \cdot \theta_{\text{ft}}$, where the mixing coefficient is determined via held-out validation sets. Wortsman et al. [47] fine-tuned a pre-trained model multiple times and then averaged them to get the final model $\theta_{\text{final}} = (1/K) \cdot \sum_{k=0}^{K} \theta_{\text{ft}}^k$. Our approach differs from theirs in the following aspects. First, different from [21, 48] where only the beginning and final model are used for weight interpolation, our method ensembles the models at each tuning epoch. We argue this is more robust and the experiments in Section 5.2 will verify this point. Second, in [21, 47, 48], there is no regularization during fine-tuning process. In our approach, the existing regularization ensures that parameter spaces in the tuning trajectory reside in generally good regions. Third, our focus is confidence reliability rather than classification accuracy, and the failure detection performance is remarkably enhanced though the accuracy is marginally improved.

4.3. The Overall Procedure

Based on reliable weight consolidation and weight space interpolation, we introduce the three-step procedure for producing a unified failure detector. Specifically, the first step is to access a pretrained model f_{θ^*} with good MisD ability. In practice, we may already have one; otherwise, we can train a randomly initialized model with existing MisD method such as CRL [36] and flat minima based method FMFP [61, 64], which are simple and can be implemented in a few lines of code. As for the OOD detection learning objective, we chose the training regularization based method LogitNorm [46] due to its simplicity. Moreover, those methods do not need auxiliary dataset, making them more flexible and practical in real-world applications. The overall procedure can be summarized as follows:

- Step 1: Prepare a pretrained model f_{θ^*}, or learn one from scratch with CRL or FMFP;
- Step 2: Fine-tune f_{θ^*} for total T epochs with $\mathcal{D}_{\text{train}}$ by minimizing $\mathcal{L}_{\text{RWC}}$ on Eq. (5);
- Step 3: Perform weight space interpolation following Eq. (6) and get the final model $f_{\theta_{\text{final}}}$.

Note that we do not introduce any additional parameter when fine-tuning. Besides, we do not use any auxiliary outlier or misclassified data. In general, our approach is simple, making it more practical for real-world applications.

5. Experiments

Datasets and implementation. We mainly conduct experiments on CIFAR-10 and CIFAR-100 [26] using ResNet110 [17] and WideResNet (WRN-28-10) [55] as the backbone. Particularly, CIFAR-100 is a quite challenging benchmark for both MisD and OOD detection. In **Step 1** of our method, we get a pre-trained classifier with standard training protocol following [36, 61], e.g., train 200 epochs using SGD with momentum 0.9, weight decay 5e-4 and batch-size 128. The start learning rate is 0.1 and decays by a factor of 10 at epochs 100, 150 respectively. In **Step 2**, we fine-tune the model for 10 epochs using the same optimizer setup except that the start learning rate is 0.01 and decays by a factor of 10 at epoch 7. We also verify the effectiveness of our method on ImageNet-subset [9], and leave the results in Appendix due to space limitation.

Evaluation metrics. In MisD, we report AURC (‰), FPR95 (%) and AUROC (%). In OOD detection, we report the average performance of FPR95 and AUROC on six standard OOD test datasets including Textures [7], SVHN [37], Places365 [58], LSUN-C [54], LSUN-R [54] and iSUN [50]. For unified failure detection, we keep the same number of misclassified and OOD examples by randomly sampling subsets of OOD data. We average measures across five runs and report AURC, FPR95 and AUROC. It is worth noting that the misclassified and OOD examples are viewed as positive examples by the failure detector, which is consistent with the problem formulation in Section 3.

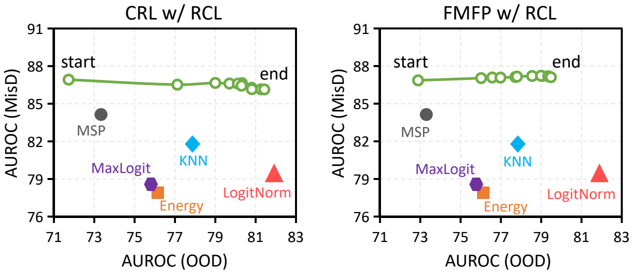

Figure 7. The green lines show the performance evolution during the reliable continual learning process (CIFAR-100 with ResNet110). Compared with other approaches, ours can achieve both strong MisD and OOD detection performance.

5.1. Results and Analysis

RCL achieves superior failure detection performance. We compare the proposed RCL with representative OOD detection and MisD approaches. Specifically, the compared OOD detection methods including MSP [18], Energy [30], MaxLogit [20], KNN [42] and LogitNorm [46]; the compared MisD methods including MSP [18], CRL [36] and FMFP [61]. For training regularization methods, we adopt MSP score by default. As shown in Table 2, failure detection performance can be significantly improved with RCL, and we highlight two groups of comparisons:

• RCL *v.s.* OOD detection methods. In Table 2 and Fig. 7, we also confirm the observation in recent studies [2, 22, 24, 63, 64] that those representative OOD detection methods often have much worse MisD performance than MSP. Consequently, although effective for OOD detection, those methods struggle to outperform MSP when evaluated under the unified failure detection setting. More specifically, KNN and LogitNorm yield comparable (sometimes slightly better or worse) performance with MSP, while Energy and MaxLogit typically perform worse than MSP. Our RCL can consistently achieve better failure detection performance across different datasets and architectures.

• RCL *v.s.* MisD methods. As shown in Table 2 and Fig. 7, although CRL and FMFP are good at MisD, their OOD detection performance is undesirable, which further limits their unified failure detection ability. For instance, CRL consistently performs worse than MSP on challenging datasets like CIFAR-100. The proposed RCL successfully addresses this problem, e.g., it improves AUROC of OOD detection by 8.18% and 6.21% for CRL and FMFP, respectively. Finally, the failure detection performance can be remarkably enhanced.

The above comparison suggests that RCL can equip the model with both good MisD and OOD detection ability,

Table 2. MisD, OOD detection and unified failure detection Performance on CIFAR-10 and CIFAR-100 with different network architectures. The highest score on each column is shown in bold, and we use darker color to represent higher performance.

Architecture	Method	MisD			OOD Detection		Failure Detection			ID-ACC
		AURC↓	FPR95↓	AUROC↑	FPR95↓	AUROC↑	AURC↓	FPR95↓	AUROC↑	
		CIFAR-10								
ResNet110	MSP	8.96	45.72	90.43	39.54	89.86	18.81	40.64	91.07	94.31
	Energy	14.25	67.13	85.35	43.57	91.31	23.85	57.63	89.01	94.31
	MaxLogit	14.02	67.14	85.65	45.45	90.97	24.02	57.03	88.89	94.31
	KNN	8.06	38.14	90.73	33.26	91.29	16.50	32.69	91.98	94.31
	LogitNorm	14.06	42.61	87.88	22.77	94.23	23.95	35.63	91.83	92.50
	CRL	6.56	21.86	93.61	29.74	91.13	15.98	24.15	93.57	93.66
	w/ RCL	6.56	22.59	93.59	21.65	94.33	14.26	20.55	94.82	93.63
	FMFP	5.26	20.29	94.01	20.03	94.13	11.60	18.37	94.96	94.42
	w/ RCL	**4.60**	**18.77**	**94.34**	**18.59**	**94.81**	**9.98**	**17.60**	**95.39**	**94.81**
WRN-28-10	MSP	4.52	29.77	93.29	32.94	92.26	10.52	29.87	93.55	95.86
	Energy	7.29	63.21	90.17	35.44	93.42	13.21	45.97	92.35	95.86
	MaxLogit	7.24	63.21	90.28	34.91	93.41	13.10	45.48	92.40	95.86
	KNN	3.90	25.50	93.95	22.84	94.05	8.24	23.09	94.69	95.86
	LogitNorm	6.48	33.89	90.54	**11.41**	**97.40**	10.22	27.30	94.29	95.34
	CRL	3.94	24.17	94.51	24.79	93.16	9.39	21.72	94.68	95.37
	w/ RCL	3.42	19.37	94.84	11.99	97.01	6.72	14.89	96.39	95.75
	FMFP	**2.22**	15.14	**95.90**	12.01	96.51	4.75	11.92	96.74	96.51
	w/ RCL	2.27	**13.68**	95.71	**11.32**	96.80	**4.62**	**11.69**	**96.76**	**96.58**
		CIFAR-100								
ResNet110	MSP	90.38	52.07	84.80	70.14	73.29	143.68	56.82	83.60	73.04
	Energy	122.30	73.43	77.92	69.73	76.22	165.91	70.77	80.22	73.04
	MaxLogit	119.51	73.21	78.74	69.66	75.95	164.08	70.50	80.68	73.04
	KNN	101.22	59.72	81.97	67.26	77.83	145.32	59.75	83.56	73.04
	LogitNorm	121.61	66.30	79.37	**52.72**	**82.19**	148.26	59.88	83.68	73.13
	CRL	76.93	41.40	86.92	73.05	71.97	139.17	56.25	84.16	73.92
	w/ RCL	80.56	46.68	86.14	56.82	80.95	123.90	**46.98**	**87.73**	74.04
	FMFP	67.91	**40.86**	86.86	68.46	72.91	132.87	52.94	84.18	**75.87**
	w/ RCL	**67.25**	43.09	**87.11**	59.63	79.12	**122.69**	47.46	86.75	75.71
WRN-28-10	MSP	46.49	41.10	88.47	64.13	77.49	112.03	49.30	86.66	80.83
	Energy	57.44	51.96	84.77	64.37	79.14	121.15	55.80	84.95	80.83
	MaxLogit	56.10	51.71	85.37	64.52	78.90	120.02	55.66	85.28	80.84
	KNN	49.52	45.73	87.15	60.12	80.33	109.66	48.81	86.97	80.84
	LogitNorm	73.19	62.23	81.57	**43.75**	**85.11**	122.90	53.26	85.80	79.25
	CRL	45.38	38.47	88.63	63.45	76.70	111.84	49.80	86.38	80.67
	w/ RCL	47.46	41.18	88.46	50.89	82.24	104.33	41.69	88.79	80.26
	FMFP	**37.12**	**33.61**	**90.09**	56.18	79.85	**92.64**	42.69	88.38	**82.05**
	w/ RCL	40.98	36.87	89.35	49.87	83.11	93.65	**39.78**	**89.34**	81.48

building on existing MisD methods such as CRL and FMFP. As a result, it establishes a strong unified failure detector, consistently and significantly outperforming the MSP baseline (which is considered as the existing state-of-the-art failure detection method [3, 22]). In addition, RCL is also simple to use and implement, without relying on auxiliary misclassified or outlier data. Due to space constraints, we provide more experimental results (e.g., ViT experiments, order of learning MisD and OOD detection) in Appendix.

Table 3. Results of ablation study on CIFAR-100 with ResNet110.

Method	MisD			OOD detection	
	AURC↓	FPR95↓	AUROC↑	FPR95↓	AUROC↑
CRL [36]	76.93	41.40	86.92	73.05	71.97
w/ RWC	87.14	50.69	85.38	54.49	82.24
w/ WSI	82.59	48.56	85.68	52.52	82.73
w/ RCL	80.56	46.68	86.14	56.82	80.95
w/ FT	90.73	51.26	84.50	51.61	84.02
w/ BI	77.76	44.69	86.52	61.13	78.90

5.2. Ablation and Extra Investigation

Ablation study. Table 3 reports the results of how weight regularization and interpolation affect the MisD and OOD detection performance, respectively. (1) Both the two strategies can improve OOD detection remarkably, their integration can further reduce the forgetting of previously equipped MisD performance. (2) We also compare with direct fine-tuning (FT) and the interpolation technique in [21, 48] which only performs binary interpolation (BI) between the beginning and final model. As expected, FT has good OOD detection performance but forgets previous MisD knowledge, while BI goes the opposite. Besides, the parameter λ in Eq. (5) can flexibly control the power of regularization

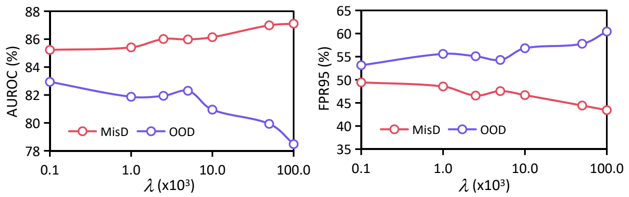

Figure 8. MisD and OOD detection performance on CIFAR-100 (ResNet110) of different regularization strength λ in Eq. (5).

term added during fine-tuning, and Fig. 8 shows its effect on individual performance of MisD and OOD detection. When increasing the value of λ, the freedom on changes of weight is reduced, which hinders the OOD detection performance but preserves more MisD ability. Nevertheless, they are consistently remarkably better than baseline. Actually, the unified failure detection performance is quite stable (see Appendix). In all experiments, we empirically set $\lambda = 1e4$ for the balance of MisD and OOD detection ability without tuning for specific dataset or network.

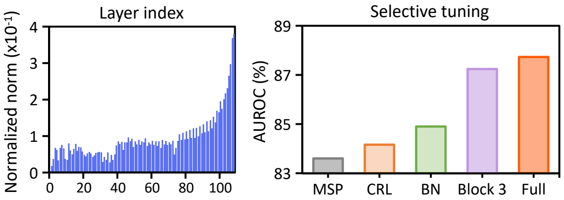

Figure 9. Selective layer tuning. Left: The ℓ_2 norms of "rate of changes" on weights at different layers before and after continual learning. Right: Comparison of the unified failure detection performance used when tuning different parts of the network. The results are obtained on ResNet110 trained on CIFAR-100.

Selective layer tuning. In our main experiments, RCL simply fine-tunes all layers with weight regularization. Here we explore selective tuning, i.e., fixing some layers and tuning others to improve the efficiency. To this end, we analyze the difference between a given model and that after tuning. Formally, given a pretrained model f_{θ^*} and the tuned $f_{\theta_{\text{final}}}$, we compute and visualize the norms of "rate of changes" ($\Delta = \frac{1}{M}\sum_i \frac{|\theta_{\text{final},i} - \theta_i^*|}{|\theta_i^*|}$, where M denotes the number of parameters) on weights at each convolution layer. Fig. 9 (Left) shows that changes in deeper layers are significantly larger, indicating that our method mainly encodes more reliable knowledge in deeper layers. Inspired by this, we propose and compare different patching strategies: (1) only fine-tune deeper layers (e.g., layers in the last block 3 of ResNet) or (2) even fix all convolutional layers and fine-tune parameters in Batch-Normalization (BN) layers. The failure detection performance in Fig. 9 (Right) shows that only tuning BN layers is less effective, while tuning the last block is both promising and efficient. Besides, our method is also data-efficient, e.g., good performance can be achieved when using only 10% of the original training data.

Table 4. Ablation study of continual learning strategies.

Metric	MSP	EWC	SI	MAS	LwF	DER
AURC↓	143.68	123.90	127.14	124.27	125.66	**114.87**
FPR95↓	56.82	46.98	48.71	46.13	46.69	**39.70**
AUROC↑	83.60	87.73	87.07	87.31	87.41	**89.71**

Experiments with more continual learning methods. Table 4 reports results using other continual learning methods on CIFAR-100 / ResNet110, e.g., MAS [1], SI [56], LwF [28] and DER [51]. They are all effective within our framework. Among them, structure-based DER performs the best but introduces additional parameters, while regularization-based methods (MAS and SI) are effective and efficient.

The choice of OOD detection method in RCL. Many OOD methods are post-hoc [18, 20, 27, 29, 30, 45], whose negative effects on MisD have been revealed and well demonstrated by recent studies. As shown in Table 2, KNN [42] unstably performs slightly better or worse than MSP, similar phenomenon exists when combing with MisD methods. Besides, post-hoc methods leave us little room to modify and explore the possibility of unified failure detection. In addition to LogitNorm, we further show that when using outlier-based method OE [19], our RCL framework yields stronger failure detection performance and the results are provided in Appendix.

6. Conclusive Remarks

Our work is dedicated to investigating the challenging and gradually concerned unified failure detection problem. We introduce "Reliable Continual Learning" which fine-tunes a given model with reliable weight consolidation and weight space interpolation, leading to a powerful framework to easily achieve on-demand reliability towards both misclassified and OOD examples. We conduct extensive experiments to verify the effectiveness of our approach, and provide extra investigation to characterize and understand the framework deeply. Our study also demonstrates that it is possible to expand the reliability of a model on newly emerged failure cases without re-training it from scratch. We hope our work inspires future research on exploiting novel learning paradigms for unified failure detection.

Acknowledgement. This work was supported in part by the National Key R&D Program of China (2022ZD0160102), the National Science and Technology Major Project (2022ZD0116500), the National Natural Science Foundation of China (U21B2042, 62320106010, 62072457, U20A20223, 62222609, 62076236), the 2035 Innovation Program of CAS, the CAS Project for Young Scientists in Basic Research (YSBR-083), the Key Research Program of Frontier Sciences of CAS (ZDBS-LY-7004), and the InnoHK program.

References

[1] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European conference on computer vision (ECCV)*, pages 139–154, 2018. 4, 8

[2] Reza Averly and Wei-Lun Chao. Unified out-of-distribution detection: A model-specific perspective. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1453–1463, 2023. 2, 6

[3] Mélanie Bernhardt, Fabio De Sousa Ribeiro, and Ben Glocker. Failure detection in medical image classification: A reality check and benchmarking testbed. *Transactions on Machine Learning Research*, 2022. 2, 5, 7

[4] Zhen Cheng, Xu-Yao Zhang, and Cheng-Lin Liu. Unified classification and rejection: A one-versus-all framework. *arXiv preprint arXiv:2311.13355*, 2023. 1

[5] Zhen Cheng, Fei Zhu, Xu-Yao Zhang, and Cheng-Lin Liu. Average of pruning: Improving performance and stability of out-of-distribution detection. *arXiv preprint arXiv:2303.01201*, 2023. 1, 2

[6] C Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 16(1):41–46, 1970. 1, 2

[7] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3606–3613, 2014. 6

[8] Charles Corbière, Nicolas Thome, Avner Bar-Hen, Matthieu Cord, and Patrick Pérez. Addressing failure prediction by learning model confidence. In *Advances in Neural Information Processing Systems*, pages 2898–2909, 2019. 1, 2

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 6

[10] Thierry Denoeux. Analysis of evidence-theoretic decision rules for pattern classification. *Pattern recognition*, 30(7): 1095–1107, 1997. 1

[11] Xuefeng Du, Zhaoning Wang, Mu Cai, and Yixuan Li. Vos: Learning what you don't know by virtual outlier synthesis. In *International Conference on Learning Representations*, 2021. 2

[12] Bernard Dubuisson and Mylene Masson. A statistical decision rule with incomplete knowledge about classes. *Pattern recognition*, 26(1):155–165, 1993. 1

[13] Zhen Fang, Yixuan Li, Jie Lu, Jiahua Dong, Bo Han, and Feng Liu. Is out-of-distribution detection learnable? *Advances in Neural Information Processing Systems*, 35: 37199–37213, 2022. 3

[14] Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. *Advances in neural information processing systems*, 30, 2017. 2

[15] Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007. 2

[16] Almog Gueta, Elad Venezian, Colin Raffel, Noam Slonim, Yoav Katz, and Leshem Choshen. Knowledge is a region in weight space for fine-tuned language models. *arXiv preprint arXiv:2302.04863*, 2023. 2

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Proceedings of the European Conference on Computer Vision*, pages 630–645, 2016. 3, 6

[18] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2017. 1, 2, 3, 6, 8

[19] Dan Hendrycks, Mantas Mazeika, and Thomas G Dietterich. Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations*, 2019. 2, 8

[20] Dan Hendrycks, Steven Basart, Mantas Mazeika, Andy Zou, Joe Kwon, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. Scaling out-of-distribution detection for real-world settings. In *International Conference on Machine Learning*, 2022. 1, 2, 6, 8

[21] Gabriel Ilharco, Mitchell Wortsman, Samir Yitzhak Gadre, Shuran Song, Hannaneh Hajishirzi, Simon Kornblith, Ali Farhadi, and Ludwig Schmidt. Patching open-vocabulary models by interpolating weights. *Advances in Neural Information Processing Systems*, 35:29262–29277, 2022. 5, 7

[22] Paul F Jaeger, Carsten Tim Lüth, Lukas Klein, and Till J Bungert. A call to reflect on evaluation practices for failure detection in image classification. In *The Eleventh International Conference on Learning Representations*, 2022. 1, 2, 3, 5, 6, 7

[23] CEN Jun, Di Luan, Shiwei Zhang, Yixuan Pei, Yingya Zhang, Deli Zhao, Shaojie Shen, and Qifeng Chen. The devil is in the wrongly-classified samples: Towards unified open-set recognition. In *The Eleventh International Conference on Learning Representations*, 2022.

[24] Jihyo Kim, Jiin Koo, and Sangheum Hwang. A unified benchmark for the unknown detection capability of deep neural networks. *Expert Systems with Applications*, 229: 120461, 2023. 2, 3, 6

[25] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017. 4, 5

[26] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009. 3, 6

[27] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, pages 7167–7177, 2018. 1, 2, 8

[28] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017. 8

[29] Shiyu Liang, Yixuan Li, and R. Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *International Conference on Learning Representations*, 2018. 1, 2, 8

[30] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. In *Advances in Neural Information Processing Systems*, pages 21464–21475, 2020. 1, 2, 6, 8

[31] Yijia Lu, Han Tian, Jia Cheng, Fei Zhu, Bin Liu, Shanshan Wei, Linhong Ji, and Zhong Lin Wang. Decoding lip language using triboelectric sensors with deep learning. *Nature communications*, 13(1):1401, 2022. 1

[32] Yan Luo, Yongkang Wong, Mohan S Kankanhalli, and Qi Zhao. Learning to predict trustworthiness with steep slope loss. *Advances in Neural Information Processing Systems*, 2021. 2

[33] Alexander Ly, Maarten Marsman, Josine Verhagen, Raoul PPP Grasman, and Eric-Jan Wagenmakers. A tutorial on fisher information. *Journal of Mathematical Psychology*, 80:40–55, 2017. 2, 4

[34] Shijie Ma, Fei Zhu, Zhen Cheng, and Xu-Yao Zhang. Towards trustworthy dataset distillation. *arXiv preprint arXiv:2307.09165*, 2023. 1, 2

[35] Shijie Ma, Fei Zhu, Zhun Zhong, Xu-Yao Zhang, and Cheng-Lin Liu. Active generalized category discovery. *arXiv preprint arXiv:2403.04272*, 2024. 1

[36] Jooyoung Moon, Jihyo Kim, Younghak Shin, and Sangheum Hwang. Confidence-aware learning for deep neural networks. In *International Conference on Machine Learning*, pages 7034–7044, 2020. 1, 2, 3, 6, 7

[37] Yuval Netzer, Tao Wang, Adam Coates, A. Bissacco, Bo Wu, and A. Ng. Reading digits in natural images with unsupervised feature learning. 2011. 6

[38] Pranav Rajpurkar, Emma Chen, Oishi Banerjee, and Eric J Topol. Ai in health and medicine. *Nature medicine*, 28(1): 31–38, 2022. 1

[39] Harald Rueß. Systems challenges for trustworthy embodied systems. *arXiv preprint arXiv:2201.03413*, 2022. 1

[40] Vikash Sehwag, Mung Chiang, and Prateek Mittal. Ssd: A unified framework for self-supervised outlier detection. In *International Conference on Learning Representations*, 2020. 1, 2

[41] Yiyou Sun, Chuan Guo, and Yixuan Li. React: Out-of-distribution detection with rectified activations. In *Advances in Neural Information Processing Systems*, 2021. 2

[42] Yiyou Sun, Yifei Ming, Xiaojin Zhu, and Yixuan Li. Out-of-distribution detection with deep nearest neighbors. In *International Conference on Machine Learning*, pages 20827–20840. PMLR, 2022. 1, 2, 6, 8

[43] Haoru Tan, Sitong Wu, and Jimin Pi. Semantic diffusion network for semantic segmentation. *Advances in Neural Information Processing Systems*, 35:8702–8716, 2022. 1

[44] Haoru Tan, Sitong Wu, Fei Du, Yukang Chen, Zhibin Wang, Fan Wang, and Xiaojuan Qi. Data pruning via moving-one-sample-out. *Advances in Neural Information Processing Systems*, 36, 2024. 1

[45] Haoqi Wang, Zhizhong Li, Litong Feng, and Wayne Zhang. Vim: Out-of-distribution with virtual-logit matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4921–4930, 2022. 1, 2, 8

[46] Hongxin Wei, Renchunzi Xie, Hao Cheng, Lei Feng, Bo An, and Yixuan Li. Mitigating neural network overconfidence with logit normalization. In *International Conference on Machine Learning*, pages 23631–23644, 2022. 2, 3, 6

[47] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, pages 23965–23998. PMLR, 2022. 5

[48] Mitchell Wortsman, Gabriel Ilharco, et al. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7959–7971, 2022. 5, 7

[49] Sitong Wu, Tianyi Wu, Haoru Tan, and Guodong Guo. Pale transformer: A general vision transformer backbone with pale-shaped attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2731–2739, 2022. 1

[50] Pingmei Xu, Krista A Ehinger, Yinda Zhang, Adam Finkelstein, Sanjeev R Kulkarni, and Jianxiong Xiao. Turkergaze: Crowdsourcing saliency with webcam based eye tracking. *arXiv preprint arXiv:1504.06755*, 2015. 6

[51] Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3014–3023, 2021. 8

[52] Chun Yang, Chang Liu, and Xu-Cheng Yin. Weakly correlated knowledge integration for few-shot image classification. *Machine Intelligence Research*, 19(1):24–37, 2022. 1

[53] Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *arXiv preprint arXiv:2110.11334*, 2021. 3

[54] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 6

[55] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference*, 2016. 6

[56] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International conference on machine learning*, pages 3987–3995. PMLR, 2017. 4, 8

[57] Hongbo Zhao, Bolin Ni, Haochen Wang, Junsong Fan, Fei Zhu, Yuxi Wang, Yuntao Chen, Gaofeng Meng, and Zhaoxiang Zhang. Continual forgetting for pre-trained vision models. *arXiv preprint arXiv:2403.11530*, 2024. 4

[58] Bolei Zhou, Àgata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2018. 6

[59] Fei Zhu, Zhen Cheng, Xu-yao Zhang, and Cheng-lin Liu. Class-incremental learning via dual augmentation. *Advances in Neural Information Processing Systems*, 34:14306–14318, 2021. 4

[60] Fei Zhu, Xu-Yao Zhang, Chuang Wang, Fei Yin, and Cheng-Lin Liu. Prototype augmentation and self-supervision for

incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5871–5880, 2021. 4

[61] Fei Zhu, Zhen Cheng, Xu-Yao Zhang, and Cheng-Lin Liu. Rethinking confidence calibration for failure prediction. In *European Conference on Computer Vision*, pages 518–536. Springer, 2022. 1, 2, 3, 6

[62] Fei Zhu, Xu-Yao Zhang, Rui-Qi Wang, and Cheng-Lin Liu. Learning by seeing more classes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(6):7477–7493, 2022.

[63] Fei Zhu, Zhen Cheng, Xu-Yao Zhang, and Cheng-Lin Liu. Openmix: Exploring outlier samples for misclassification detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12074–12083, 2023. 2, 3, 6

[64] Fei Zhu, Xu-Yao Zhang, Zhen Cheng, and Cheng-Lin Liu. Revisiting confidence estimation: Towards reliable failure prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 1, 2, 3, 5, 6

[65] Fei Zhu, Shijie Ma, Zhen Cheng, Xu-Yao Zhang, Zhaoxiang Zhang, and Cheng-Lin Liu. Open-world machine learning: A review and new outlooks. *arXiv preprint arXiv:2403.01759*, 2024. 1

[66] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020. 1