2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) | 979-8-3503-5300-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/CVPR52733.2024.01657

SURE: SUrvey REcipes for building reliable and robust deep networks

Yuting Li[1,2], Yingyi Chen[3], Xuanlong Yu[4,5], Dexiong Chen[†6], and Xi Shen[†1]

[1]Intellindust, China
[2]China Three Gorges University, China
[3]ESAT-STADIUS, KU Leuven, Belgium
[4]SATIE, Paris-Saclay University, France
[5]U2IS, ENSTA Paris, Institut Polytechnique de Paris, France
[6]Max Planck Institute of Biochemistry, Germany

Abstract

In this paper, we revisit techniques for uncertainty estimation within deep neural networks and consolidate a suite of techniques to enhance their reliability. Our investigation reveals that an integrated application of diverse techniques–spanning model regularization, classifier and optimization–substantially improves the accuracy of uncertainty predictions in image classification tasks. The synergistic effect of these techniques culminates in our novel SURE approach. We rigorously evaluate SURE against the benchmark of failure prediction, a critical testbed for uncertainty estimation efficacy. Our results showcase a consistently better performance than models that individually deploy each technique, across various datasets and model architectures. When applied to real-world challenges, such as data corruption, label noise, and long-tailed class distribution, SURE exhibits remarkable robustness, delivering results that are superior or on par with current state-of-the-art specialized methods. Particularly on Animal-10N and Food-101N for learning with noisy labels, SURE achieves state-of-the-art performance without any task-specific adjustments. This work not only sets a new benchmark for robust uncertainty estimation but also paves the way for its application in diverse, real-world scenarios where reliability is paramount. Our code is available at https://yutingli0606.github.io/SURE/.

1. Introduction

Deep neural networks (DNNs) have established themselves as powerful and adaptable tools for prediction tasks on structured data. However, accurately assessing the reliability of their predictions continues to be a substantial challenge. In safety-critical areas such as medical diagnostics [2, 43, 56], robotics [29, 49], autonomous driving [9, 18, 49], and earth observation systems [24, 52], decisions based on overconfident predictions can result in severe

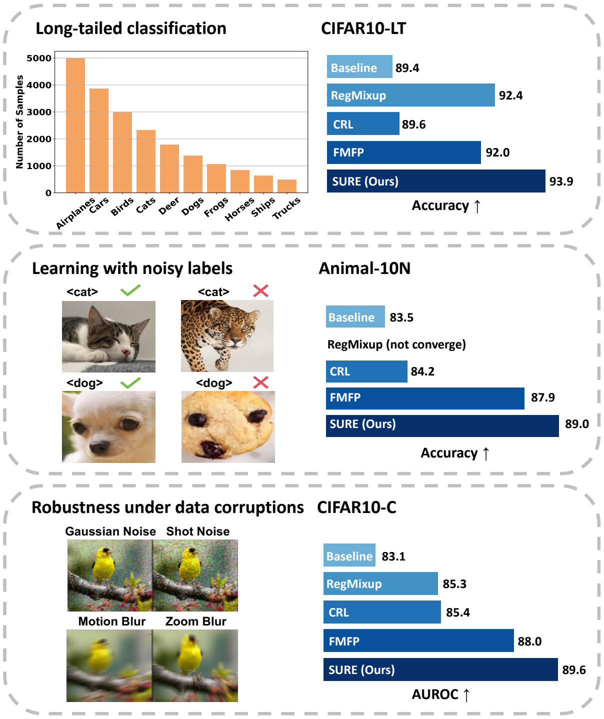


Figure 1. **SURE consistently performs better than previous approaches to uncertainty estimation under various scenarios.** Note that we did not manage to scale RegMixup [59] to the learning with noisy label task. Baseline refers to the MSP [31] method.

[†]Corresponding Author.

2575-7075/24/$31.00 ©2024 IEEE
DOI 10.1109/CVPR52733.2024.01657

consequences. Consequently, ensuring the robust dependability of artificial intelligence systems grounded in DNNs is of utmost importance.

Addressing the issue of overconfidence in deep learning has been a focal point of significant research efforts, such as [25, 32, 46, 48, 55, 66]. However, a key limitation of these methods is their restricted testing scenarios, typically confined to benchmark datasets for a single, predefined task like failure prediction or out-of-distribution (OOD) detection. The effectiveness of these methods in more complex, real-world situations involving issues like data corruption, label noise, or long-tailed class distributions remains largely under-explored. Our experiments reveal that no single approach excels uniformly across these diverse scenarios, as depicted in Figure 1. In this work, we propose a unified model designed to effectively address all these challenges.

In our pursuit to enhance uncertainty estimation, we start by examining the combined impact of several preexisting methods, leading to the discovery of an integrated approach that significantly refines this estimation. We classify these methods based on their function in the model training process: regularization, classifier and optimization. For regularization, we utilize techniques such as RegMixup regularization [59], correctness ranking loss (CRL) [54] and cosine similarity classifier (CSC) [23, 33], which can help in increasing entropy for challenging samples. In the realm of optimization, we incorporate Sharpness-Aware Minimization (SAM) [19] and Stochastic Weight Averaging (SWA) [35], as recommended by FMFP [81], to ensure that the model can converge towards flatter minima. The synergistic integration of these diverse techniques culminates in our novel approach, which we name SURE. This method harnesses the strengths of each individual component, resulting in a more robust and reliable model.

In the evaluation of SURE, we first focus on failure prediction, a pivotal task for evaluating uncertainty estimation. Our evaluations reveal that SURE consistently outperforms models deploying individual technique. This superior performance is evident across various datasets such as CIFAR-10 [40], CIFAR-100 [40], Tiny-ImageNet [41] and also across various model architectures, namely ResNet [28], VGG [64], DenseNet [34], WideResNet [76] and DeiT [70]. Notably, SURE even surpasses OpenMix [82], a method that leverages additional OOD data. By applying SURE directly to real-world scenarios, without or with minimal task-specific adjustments, we further witness its effectiveness in bringing robustness to the models. Specifically, the real-world challenges include data corruption in CIFAR10-C [30], label noise in Animal-10N [65] and Food-101N [42], and skewed class distribution in CIFAR-LT [12]. In these contexts, SURE achieves results that are either superior to or on par with the latest specialized methods. A standout achievement is observed on Food-101N, where SURE attains an impressive accuracy of 88.0%, significantly surpassing the previous state-of-the-art method, Jigsaw-ViT [7], which achieved accuracy of 86.7% by using extra training data to pre-train the model. This demonstrates SURE's remarkable capability in handling complex real-world data challenges.

The main contributions of this paper are summarized as follows:

- We reveal that existing methods do not uniformly excel in various real-world challenges. This analysis underlines the need for more reliable and robust approaches to handle the complexities of real-world data.
- We propose a novel approach, named SURE, for robust uncertainty estimation, inspired by the synergistic effect achieved by combining multiple techniques, across model regularization, classifier and optimization. Models trained under our SURE approach consistently achieve better performance in failure prediction than models that deploy individual technique, across various datasets and model architectures.
- When applied directly to real-world scenarios, SURE consistently shows performance at least comparable to state-of-the-art specialized methods.

2. Related work

Uncertainty estimation Quantifying uncertainty for DNN outputs can improve the interpretability and trustworthiness of the predictions and serve various downstream tasks, such as model calibration [25], OOD detection [32, 46], failure prediction [10, 31], etc. MSP [31], Entropy [66], and Energy [48] provide uncertainty estimates for outputs using the information provided by the DNN itself. Modifying the architecture and optimization of the DNN can further improve the performance of these measures on downstream tasks, *i.e.*, attaining robust and reliable uncertainty estimates. To balance the sensitivity and smoothness of the DNN and achieve robust uncertainty estimation, DDU [55] applies spectral normalization layers [53] to encourage bi-Lipschitzness and LDU [20] introduces distinction maximization layer and an uncertainty estimation head to the DNN. Yet, they all lead to increased training parameters, and a predefined input image size is needed for the former, which also lacks scalability. A simpler adjustment to DNN architecture introduced by OVADM [58] improves OOD detection performance, which replaces the output layer with an ℓ_2 distance-based layer and uses a one-vs-all loss for training. In terms of optimization, in addition to FMFP [81] mentioned in the previous section, Qu et al. [61] use meta-learning to achieve flat minima yet apply to the auxiliary uncertainty estimators. Methods based on data augmentation, such as Mixup [77], RegMixup [59] and OpenMix [82], apply regularization when training the model, resulting in dependable uncertainty estimates, while ensuring classification accuracy. This work selects and integrates these methods and obtains a scalable solution to improve classification accuracy with more reliable uncertainty estimates.

Learning with noisy labels This task aims to perform learning while noisy annotated data is presented in the training set. Mainstream solutions include: *i)* label correction, which aims at revising possibly wrong labels with more consistent substitutes [65, 68, 74, 78]; *ii)* semi-supervised learning, which trains networks in a semi-supervised manner with only the clean labels used [3, 15, 39, 44]; *iii)* sample re-weighting, which assigns more weights to possibly clean samples [5, 6, 17, 26, 36, 51, 60, 72, 75]; *iv)* over-fitting prevention, which prevents networks from overfitting on noisy training data so as to have better generalization on clean test set [6, 37, 47, 50, 57, 79]. Specifically, WarPI [67] and [37] are based on meta-learning framework, which propose adaptively rectifying the training procedure for the classification network. SSR+ [17] designs a sample selection and relabelling based on a non-parametric KNN classifier and a parametric classifier.

Long-tailed classification In addressing the long-tailed classification challenge, various strategies have been proposed. BBN [80] utilizes a dual-branch network to balance learning between different class frequencies, while SSP [73] leverages self-supervised learning and semi-supervised learning for contrastive learning in long-tailed distribution. LDAM-DRW [4] introduces logit compensation to handle class frequency imbalance. Hybrid-SC [71] proposes a two-branch network for supervised contrastive learning and reducing classifier bias. BCL [83] develops a balanced contrastive loss, ensuring that all classes are optimized for a regular simplex configuration that yields a balanced feature space. Recently, GLMC [16] proposes a new paradigm that contains a global and local mixture consistency loss to improve the robustness of the feature extractor, and a cumulative head-tail soft label re-weighted loss to mitigate the head class bias problem.

In this work, we show that by simply applying the uncertainty score provided by DNNs trained using SURE to the re-weighting training strategy, which is commonly used in the community of long-tailed classification [1, 4, 38, 69, 80], the classification performance on imbalanced data can on par with the previous SOTAs.

3. Methods

As illustrated in Figure 2, our proposed approach SURE aims to train reliable and robust DNNs through two aspects: *i)* increasing entropy for hard samples; *ii)* enforcing flat minima during optimization. In the following, we denote the dataset by $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$ where $\mathbf{x}_i$ is the input image, $\mathbf{y}_i$ is its ground-truth label and N is the number of samples.

The recipes in SURE for increasing entropy for hard samples consist of three components: the RegMixup regularization [59] denoted as $\mathcal{L}_{\text{mix}}$, the correctness ranking loss $\mathcal{L}_{\text{crl}}$ which serves to regularize the class probabilities by aligning the confidence with the ordinal ranking of correctness, and the cosine similarity classifier (CSC). These recipes are employed collectively to optimize the objective, which includes a task-specific loss, *e.g.*, the cross-entropy loss for classification, denoted as $\mathcal{L}_{\text{ce}}$, in addition to the RegMixup regularization $\mathcal{L}_{\text{mix}}$, and the confidence-aware regularization $\mathcal{L}_{\text{crl}}$ based on the historical correctness information gathered during training. The recipes for enforcing flat minima lie in leveraging Sharpness-Aware Minimization (SAM) [19] and Stochastic Weight Averaging (SWA) [35] during optimization.

This section is organized as follows: Section 3.1 illustrates our objective function and CSC to increase entropy for hard samples. Section 3.2 introduces the flat minima-enforced techniques. Implementation details are provided in Section 3.3.

3.1. Increasing entropy for hard samples

Total loss As described above, the objective function of SURE is composed of three components, which is expressed as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{ce}} + \lambda_{\text{mix}}\mathcal{L}_{\text{mix}} + \lambda_{\text{crl}}\mathcal{L}_{\text{crl}}, \tag{1}$$

where λ_{mix} and λ_{crl} denote hyper-parameters to balance the contribution of each loss component to the total loss. The impact of λ_{mix} and λ_{crl} is studied in the Supplementary Material.

RegMixup regularization $\mathcal{L}_{\text{mix}}$ Mixup [77] is a widely used data augmentation. Given two input-target pairs $(\mathbf{x}_i,\ \mathbf{y}_i)$ and $(\mathbf{x}_j,\ \mathbf{y}_j)$, we obtain an augmented sample $(\tilde{\mathbf{x}}_i, \tilde{\mathbf{y}}_i)$ by linearly interpolating between them:

$$\tilde{\mathbf{x}}_i = m\mathbf{x}_i + (1-m)\mathbf{x}_j, \quad \tilde{\mathbf{y}}_i = m\mathbf{y}_i + (1-m)\mathbf{y}_j, \tag{2}$$

where m denotes the mixing coefficient, following a Beta distribution:

$$m \sim \text{Beta}(\beta, \beta), \quad \beta \in (0, \infty). \tag{3}$$

The RegMixup regularization $\mathcal{L}_{\text{mix}}$ consists of fitting the model additionally on the augmented samples $(\tilde{\mathbf{x}}_i, \tilde{\mathbf{y}}_i)$:

$$\mathcal{L}_{\text{mix}}(\tilde{\mathbf{x}}_i, \tilde{\mathbf{y}}_i) = \mathcal{L}_{\text{ce}}(\tilde{\mathbf{x}}_i, \tilde{\mathbf{y}}_i), \tag{4}$$

with $\beta = 10$ leading to a heavy mixing of two samples with high probability.

Similar to RegMixup [59], we incorporate $\mathcal{L}_{\text{mix}}$ as an additional regularizer alongside the original cross-entropy loss

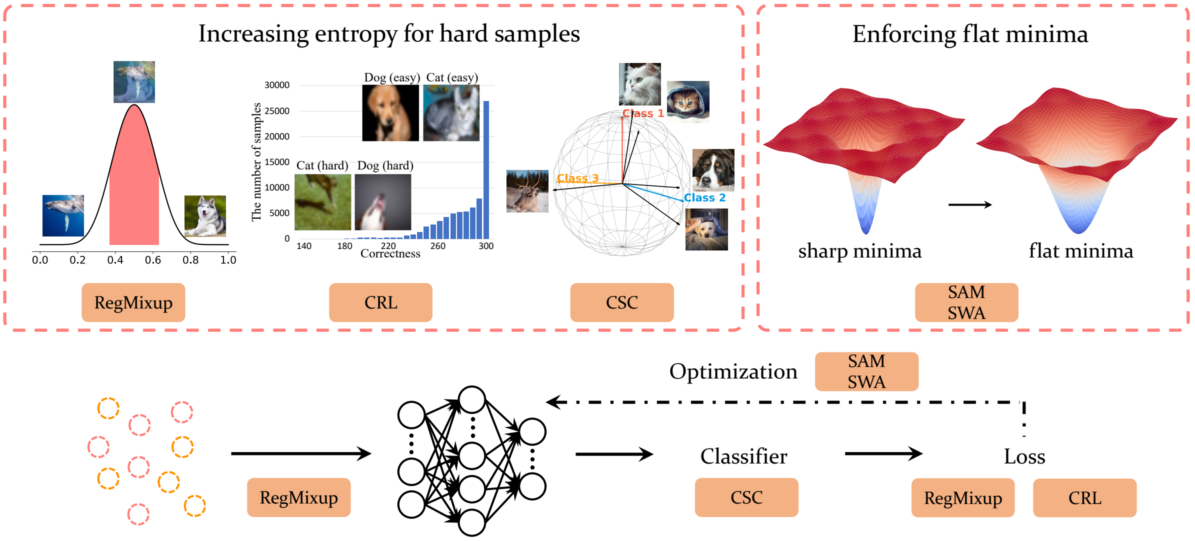


Figure 2. **Overview of recipes.** Our proposed approach SURE contains two aspects: increasing entropy for hard samples and enforcing flat minima during optimization. We incorporate RegMixup [59] loss and correctness ranking loss (CRL) [54] as our loss function and employ cosine similarity classifier (CSC) [23, 33] as our classifier to increase entropy for hard samples. As in optimization, we leverage Sharpness-Aware Minimization (SAM) [19] and Stochastic Weight Averaging (SWA) [35] to find flat minima.

on $(\mathbf{x}_i, \mathbf{y}_i)$, *i.e.*, $\mathcal{L}_{\text{ce}}$ in (1). A high value of β results in a heavy mixing of samples, prompting the model to exhibit high entropy on heavily interpolated samples, which can be regarded as challenging examples.

Correctness ranking loss $\mathcal{L}_{\text{crl}}$ The correctness ranking loss [54] encourages the DNN to align the model's confidence with the ordinal ranking of historical correctness information gathered during training. Specifically, for two input images $\mathbf{x}_i$ and $\mathbf{x}_j$, $\mathcal{L}_{\text{crl}}$ is defined as:

$$\mathcal{L}_{\text{crl}}(\mathbf{x}_i, \mathbf{x}_j) = \max(0, |c_i - c_j| - \text{sign}(c_i - c_j)(\mathbf{s}_i - \mathbf{s}_j)), \tag{5}$$

where c_i and c_j represent the proportions of correct prediction events for $\mathbf{x}_i$ and $\mathbf{x}_j$ during the training, $\mathbf{s}_i$ and $\mathbf{s}_j$ denote the confidence score for $\mathbf{x}_i$ and $\mathbf{x}_j$, which are the softmax scores in this work, sign denotes the sign function, $\mathcal{L}_{\text{crl}}$ aims to align the confidence score to the correctness statistics. Hard samples, which are less likely to be correctly predicted during training, are encouraged to have lower confidence and thus, higher entropy.

Cosine Similarity Classifier (CSC) CSC has shown to be effective on few-shot classification [23, 33] by simply replacing the last linear layer with a cosine classifier. For the image $\mathbf{x}_i$, we denote the classification logits for $\mathbf{x}_i$ belonging to class k as $\mathbf{s}_i^k$, which is defined as follows:

$$\mathbf{s}_i^k = \tau \cdot \cos(f_\theta(\mathbf{x}_i), w^k) = \tau \cdot \frac{f_\theta(\mathbf{x}_i)}{\left\| f_\theta(\mathbf{x}_i) \right\|_2} \cdot \frac{w^k}{\|w^k\|_2}, \tag{6}$$

where τ is the temperature hyper-parameter, f_θ is a DNN parameterized with θ, used to extract features of input images, w^k representing the k-th class prototype, denotes the weight of the k-th class.

CSC encourages the classifier to focus on the directional alignment between the feature vector extracted from the input image and the class prototype vector, rather than the dot product. This makes it conceptually distinct from the traditional linear classifier, where magnitude plays a significant role. A key benefit of using CSC is its ability to handle hard samples better. CSC views hard samples as equidistant in angle to several class prototypes, leading to more effective interpretation and potentially higher entropy than the traditional linear classifier that uses the dot product.

3.2. Flat minima-enforced optimization

We jointly employ Sharpness-Aware Minimization (SAM) [19] and Stochastic Weight Averaging (SWA) [35] to enhance flat minima. Note that these two techniques are also jointly used in [81] to improve uncertainty estimation.

Sharpness-Aware Minimization (SAM) SAM [8, 19] is an optimization method that enhances model generalization by seeking parameters lying in flat neighborhoods such that the DNN has a uniformly small loss. For our objective function $\mathcal{L}_{\text{total}}$ and DNN parameters θ, the SAM optimizer seeks θ satisfying:

$$\min_{\theta} \max_{\|\epsilon\|_2 \leq \rho} \mathcal{L}_{\text{total}}(\theta + \epsilon), \tag{7}$$

where ϵ is a perturbation vector and ρ is the neighborhood size within which we seek to minimize the sharpness of the loss. The SAM algorithm proceeds by alternating between finding the worst-case perturbation ϵ that maximizes the loss within the ℓ_2-norm ball of radius ρ, and updating the model parameters θ to minimize this perturbed loss.

Stochastic Weight Averaging (SWA) SWA is introduced in [35], which improves the generalization of DNNs by averaging model weights over the course of training. The process begins with a standard training phase, after which SWA starts by averaging the weights at each subsequent epoch. The mathematical representation of the SWA weight update is given by:

$$\theta_{\text{SWA}} = \frac{1}{T} \sum\nolimits_{t=1}^{T} \theta_t, \tag{8}$$

where θ_t represents the model weights at epoch t , and T is the total number of epochs during which SWA is applied.

3.3. Implementation details

Following [81], our models are trained using SAM [19] with stochastic gradient descent (SGD) as the base optimizer with a momentum of 0.9, starting with an initial learning rate of 0.1 and a weight decay of 5e-4, over 200 epochs with a batch size of 128. We employ a cosine annealing learning rate schedule and set the SWA [35] start epoch to 120 and a SWA-specific learning rate of 0.05, to enhance the training effectiveness and model robustness. We set β = 10 in (3) for the Mixup data augmentation, which is following [59]. All hyper-parameters, including λ_{mix}, λ_{crl}, and τ, are selected on the validation set. An ablation study of λ_{mix}in (1), λ_{crl} in (1), and τ in (6) are provided in the Supplementary Material. In terms of finetuning DeiT-Base [70] with the ImageNet [14] pre-trained model, we set the learning rate at 0.01 with a weight decay of 5e-5 over 50 epochs and start SWA start epoch to 1 and a SWA-specific learning rate of 0.004.

4. Experiments

In this section, we evaluate the performance of SURE in failure prediction and further explore SURE's ability in tackling real-world challenges, including long-tailed classification, learning with noisy labels, and generalization in corrupted image scenarios. We first introduce the datasets used in our experiments and outline the key metrics in Section 4.1. Then, we present results on failure prediction in Section 4.2. Results on long-tailed classification are presented in Sections 4.3. In Section 4.4, we present results for learning with noisy labels. Performances on corrupted images are provided in Section 4.5. Finally, we present analysis in Section 4.6.

4.1. Datasets and evaluation metrics

CIFAR10, CIFAR100 and Tiny-ImageNet We use CIFAR [40] and Tiny-ImageNet [41] to evaluate failure prediction. CIFAR datasets are commonly used in the community [59, 81, 82] and we use Tiny-ImageNet [41] as a larger dataset to evaluate the effectiveness and robustness of our proposed method. The CIFAR10 dataset contains 60,000 color images with a resolution of 32×32, divided into 10 classes, each holding 5,000 training images and 1,000 testing images. The CIFAR100 dataset follows a similar structure, but with 100 classes. Each class contains 500 training samples and 100 testing samples. Tiny-ImageNet [41] contains 100,000 images of 200 classes downsized to 64×64 colored images which are a subset of the ImageNet dataset [14]. Each class has 500 training images. 50 images are collected for testing. Note that for all our experiments, we keep 10% of the training set as our validation set. We report the means and standard deviations over *three* runs.

Long-Tailed CIFAR: CIFAR10-LT and CIFAR100-LT We use CIFAR10-LT and CIFAR100-LT [12] to evaluate long-tailed classification. Note that these datasets are widely used as evaluation datasets in the community [1, 16, 83]. Following previous works [1, 16, 83], the datasets are created by only keeping the number of training samples per class according to an exponential function $\tilde{N}_i = N_i \mu^i$ where i is the class index, N_i is the number of training images in the i-th class and $\mu \in (0, 1)$. The imbalanced factor IF quantifies the level of distribution imbalance and determines μ, which is defined by the ratio between the maximum and the minimum number of samples in a category. The test set remains unchanged.

Animal-10N and Food-101N Animal-10N [65] and Food-101N [42] are two real-world datasets where noisy labels are present in the training set. Animal-10N is a benchmark that contains 10 animal classes with confusing appearance. The training set size is 50,000, and the test set is 5,000. The estimated label noise ratio of the training set is 8%. No data augmentation is applied so as to follow the settings in [65]. Food-101N contains 310,009 training images of different food recipes collected online and are classified into 101 classes. The training set is with an approximate noise ratio of 20%. Following [42], the learned models should be evaluated on the test set of Food-101 with 25,250 clean labeled images.

CIFAR10-C To evaluate the model's robustness, we use CIFAR10-C dataset [30], which applies 15 common image corruptions, *e.g.*, Gaussian noise, impulse noise, motion blur, frost, etc., to CIFAR10 [40] test set. Each type of corruption is characterized by five severity levels, as these corruptions can occur at different intensities.

Evaluation Metrics We report metrics that are commonly used in the community of failure prediction to assess the

Backbones	Methods	CIFAR-10 [40]				CIFAR-100 [40]				Tiny-ImageNet [41]			
		Acc. ↑	AURC ↓	AUROC ↑	FPR95 ↓	Acc. ↑	AURC ↓	AUROC ↑	FPR95 ↓	Acc. ↑	AURC ↓	AUROC ↑	FPR95 ↓
ResNet-18 [28]	MSP [31]	94.89±0.20	6.78±0.33	92.20±0.55	38.73±2.89	75.87±0.31	69.44±2.11	87.00±0.21	60.73±1.16	63.39±0.59	136.50±1.08	85.62±0.35	63.99±0.64
	RegMixup [59]	95.69±0.13	4.74±0.27	92.96±0.29	34.26±1.98	77.90±0.37	59.23±1.65	87.61±0.13	58.65±0.43	66.36±0.43	115.08±1.98	86.53±0.27	62.54±0.43
	CRL [54]	94.85±0.10	5.09±0.28	93.64±0.48	35.33±1.73	76.42±0.21	62.78±0.21	88.07±0.17	59.02±0.39	65.50±0.03	117.46±0.56	87.01±0.13	61.15±0.07
	SAM [19]	95.30±0.25	3.97±0.33	94.53±0.31	31.13±3.62	76.60±0.21	62.97±1.02	87.72±0.10	59.35±0.87	64.95±0.21	120.04±2.11	87.19±0.57	**59.98±0.55**
	SWA [35]	95.38±0.09	4.00±0.21	94.40±0.50	35.70±1.44	77.65±0.19	55.87±0.32	88.55±0.25	60.43±1.90	68.09±0.19	102.11±0.51	87.27±0.15	60.63±1.38
	FMFP [81]	95.60±0.09	3.56±0.06	94.74±0.10	33.49±0.33	77.82±0.08	55.03±0.52	88.59±0.07	59.79±0.31	68.18±0.42	100.93±2.12	87.45±0.05	60.18±1.26
	SURE	**96.14±0.16**	**2.97±0.13**	**95.08±0.04**	**28.64±0.66**	**80.49±0.18**	**45.81±0.15**	**88.73±0.24**	**58.91±0.58**	**69.55±0.10**	**93.46±0.82**	**87.67±0.12**	60.13±0.32
VGG [64]	MSP [31]	93.30±0.21	10.41±0.33	90.71±0.04	44.66±1.81	72.43±0.42	91.40±1.95	85.69±0.90	64.41±1.66	59.52±0.62	156.45±2.51	86.33±0.63	63.79±0.95
	RegMixup [59]	94.11±0.28	9.89±0.81	89.90±0.26	39.93±1.58	73.51±0.18	85.98±1.05	86.35±0.32	61.70±1.83	63.04±0.57	146.72±2.59	85.60±0.39	**59.00±1.27**
	CRL [54]	93.42±0.09	7.61±0.44	92.88±0.56	39.66±2.83	72.63±0.27	80.94±0.47	87.37±0.28	61.96±0.77	60.20±0.36	146.76±1.42	**87.42±0.28**	59.26±1.44
	SAM [19]	94.11±0.06	5.97±0.08	93.68±0.13	37.21±2.92	73.33±0.36	77.44±0.75	87.42±0.33	63.19±0.58	61.24±0.07	142.54±1.04	86.82±0.25	62.93±1.12
	SWA [35]	93.76±0.25	6.64±0.24	93.43±0.16	40.44±1.27	73.98±0.16	74.23±0.58	87.30±0.14	62.89±1.80	62.48±0.19	137.01±0.71	86.29±0.16	62.15±1.64
	FMFP [81]	94.26±0.23	5.89±0.16	93.46±0.26	40.67±3.14	74.77±0.31	70.07±1.26	87.58±0.19	60.98±1.16	62.95±0.16	134.04±1.42	86.36±0.12	61.71±1.08
	SURE	**95.00±0.11**	**4.98±0.24**	**93.79±0.62**	**35.92±2.95**	**76.51±0.07**	**65.25±0.17**	**87.59±0.07**	**60.27±0.60**	**63.75±0.11**	**131.40±0.28**	86.12±0.19	63.04±1.05
DenseNet [34]	MSP [31]	94.72±0.23	5.94±0.23	93.00±0.45	37.00±0.31	75.14±0.07	74.68±0.32	86.22±0.22	62.79±0.80	57.90±0.25	180.08±2.52	83.65±0.29	68.61±0.37
	RegMixup [59]	95.13±0.22	6.03±0.50	92.20±0.80	38.63±1.63	77.29±0.16	63.96±1.15	86.57±0.07	63.76±1.10	61.96±0.09	147.22±1.57	**84.91±0.17**	65.92±0.40
	CRL [54]	94.79±0.02	5.58±0.42	93.22±0.61	37.34±2.73	76.09±0.06	65.96±0.62	87.41±0.11	60.67±0.72	58.80±0.56	169.44±3.74	84.49±0.04	66.05±0.60
	SAM [19]	95.31±0.10	4.25±0.17	94.15±0.46	33.33±1.27	78.17±0.26	57.20±0.73	86.99±0.23	61.42±0.74	60.49±0.31	158.94±3.86	84.39±0.57	66.51±1.85
	SWA [35]	94.86±0.09	4.65±0.18	94.27±0.27	35.78±4.61	78.17±0.26	57.20±0.73	87.23±0.22	63.33±0.63	60.74±0.46	159.68±3.12	83.83±0.07	68.03±0.75
	FMFP [81]	95.07±0.15	4.11±0.19	94.74±0.06	34.67±0.48	78.33±0.40	54.88±1.62	87.92±0.46	60.52±1.12	61.18±0.72	154.98±3.72	84.29±0.26	66.66±1.21
	OpenMix [82][§]	95.51±0.23	4.68±0.72	93.57±0.81	33.57±3.70	78.97±0.31	53.83±0.93	87.45±0.18	62.22±1.15	-	-	-	-
	SURE	**95.57±0.06**	**3.51±0.09**	**94.91±0.25**	**29.52±0.56**	**80.02±0.13**	**46.69±0.59**	**88.78±0.26**	**58.37±0.39**	**62.61±0.18**	**142.59±2.16**	84.31±0.42	**65.39±2.12**
WRNet [76]	MSP [31]	95.71±0.17	5.90±0.89	92.19±0.82	35.95±3.75	79.15±0.19	53.02±0.89	88.21±0.06	59.46±1.23	67.52±0.18	107.97±0.80	86.78±0.20	61.68±0.99
	RegMixup [59]	97.03±0.04	3.47±0.26	93.10±0.56	26.16±1.17	82.14±0.47	47.01±2.12	87.70±0.17	55.24±1.19	69.63±0.09	95.96±0.21	87.38±0.21	59.09±0.75
	CRL [54]	95.87±0.08	3.85±0.20	94.10±0.06	32.73±1.22	80.10±0.28	47.99±1.08	88.43±0.34	59.44±1.45	69.00±0.22	97.46±0.90	87.42±0.23	61.02±1.71
	SAM [19]	96.47±0.11	2.91±0.38	94.79±0.29	28.05±1.56	80.67±0.31	44.93±0.87	89.01±0.31	56.60±1.30	69.86±0.37	93.66±2.03	87.49±0.30	60.44±1.19
	SWA [35]	94.86±0.09	4.65±0.18	94.27±0.27	35.78±4.61	81.31±0.33	41.15±0.89	89.39±0.16	57.57±1.97	71.27±0.16	84.97±0.12	87.71±0.26	60.00±2.42
	FMFP [81]	96.47±0.12	2.33±0.08	95.73±0.01	26.68±2.62	81.66±0.12	39.60±0.15	89.51±0.10	56.41±1.44	71.62±0.04	83.04±0.16	87.78±0.03	60.09±0.83
	OpenMix [82][§]	**97.16±0.10**	2.32±0.15	94.81±0.34	22.08±1.86	82.63±0.06	39.61±0.54	89.06±0.11	55.00±1.29	-	-	-	-
	SURE	97.02±0.20	**1.79±0.16**	**96.18±0.01**	**19.53±1.23**	**83.71±0.10**	**32.10±0.28**	**90.33±0.18**	**54.34±0.29**	**73.34±0.36**	**74.11±0.97**	**88.23±0.31**	**58.17±1.50**
DeiT-B ⋆ [70]	MSP [31]	98.28±0.08	0.97±0.02	95.76±0.28	20.47±5.38	89.71±0.03	17.66±0.56	90.40±0.25	50.99±0.61	-	-	-	-
	RegMixup [59]	98.90±0.04	0.89±0.05	94.30±0.25	24.98±3.87	90.79±0.11	15.38±0.51	90.34±0.33	52.01±1.76	-	-	-	-
	CRL [54]	98.27±0.04	0.99±0.11	95.85±0.44	19.65±2.51	89.74±0.16	17.61±0.71	90.30±0.18	51.58±0.23	-	-	-	-
	SAM [19]	98.62±0.10	0.58±0.09	96.89±0.34	**15.74±1.71**	90.43±0.17	15.29±0.19	90.75±0.15	50.02±1.52	-	-	-	-
	SWA [35]	98.44±0.07	0.82±0.03	96.11±0.20	17.78±3.23	90.17±0.34	15.37±0.44	90.86±0.38	50.64±3.37	-	-	-	-
	FMFP [81]	98.76±0.02	**0.46±0.02**	**97.15±0.16**	16.17±0.55	90.53±0.13	14.30±0.18	**91.15±0.32**	51.90±1.50	-	-	-	-
	SURE	**98.92±0.07**	0.86±0.08	94.37±0.69	27.52±3.11	**91.18±0.01**	**13.79±0.29**	90.85±0.05	**48.81±0.39**	-	-	-	-

[§] reports the results given by models training on extra outliers and all the training data on CIFAR10 [40] CIFAR100 [40]

[⋆] reports the results given by finetuning ImageNet [14] pre-trained DeiT-B [70] for 50 epochs

Table 1. **Comparison of the performance of failure prediction on CIFAR10 [40], CIFAR100 [40] and Tiny-ImageNet [41].** We keep 10% training data as the validation set to select the best model. The means and standard deviations over *three* runs are reported. ↓ and ↑ indicate that lower and higher values are better respectively. AURC [22] values are multiplied by 10^3, and all remaining values are in percentage.

performance of our model, including Accuracy (Acc.), Area Under the Risk-Coverage Curve (AURC) [22], Area Under the Receiver Operating Characteristic Curve (AUROC) [13], False Positive Rate at 95% True Positive Rate (FPR95). Specifically, we leverage AURC, which is complementary to Accuracy to measure the uncertainty of the model. AURC measures the area under the curve drawn by plotting the risk according to coverage. Given a confidence threshold, the coverage indicates the ratio of samples whose confidence estimates are higher than the confidence threshold, and the risk, also known as the selective risk [21], is an error rate computed by using those samples. A lower value of AURC implies a higher accuracy, and correct and erroneous predictions can be well-separable by a confidence threshold. The definitions of AUROC [13] and FPR95 are detailed in the Supplementary Material.

4.2. Failure prediction

We present results on failure prediction on CIFAR10 [40], CIFAR100 [40] and Tiny-ImageNet [41] in Table 1. Experiments are conducted with different backbones: ResNet18 [28], VGG16-BN [64], DenseNetBC [34], WRNet28 [76] and DeiT [70]. The architectures and datasets are commonly used in the community [59, 81, 82]. Note that to ensure the reliability of our model and maintain the rigor and fairness of our experiments, we split 10% of the training data as a validation set for the selection of hyper-parameters and report the performance on the test set. All the experiments are repeated *three* times and we report the mean and the standard deviation in the table. From Table 1, we can see that our SURE achieves significantly better performance on almost all the metrics than all the competitive approaches across different datasets and diverse architectures, which demonstrates the effectiveness and robustness of our proposed approaches. Note that even though the latest approach OpenMix [82] trains on all the training sets as well as additional outlier data, our SURE still maintains a significant performance gain without using any additional data.

4.3. Long-tailed classification

Uncertainty-aware re-weighting When the training data distribution is imbalanced, we find that the second stage uncertainty-aware re-weighting can consistently improve the performance. Note that the two-stage training strategy is commonly used in the community of long-tailed classification [1, 4, 38, 69, 80]. The key difference is that

Methods	CIFAR10-LT [12]			CIFAR100-LT [12]		
	IF=100	IF=50	IF=10	IF=100	IF=50	IF=10
CE	70.40	74.80	86.40	38.30	43.90	55.70
Mixup [77]	73.06	77.82	87.1	39.54	54.99	58.02
CB-Focal [12]	74.57	79.27	87.10	39.60	45.17	57.99
LDAM-DRW [4]	77.03	81.03	88.16	42.04	46.62	58.71
SSP [73]	77.83	82.13	88.53	43.43	47.11	58.91
BBN [80]	79.82	81.18	88.32	42.56	47.02	59.12
Casual model [69]	80.60	83.60	88.50	44.10	50.30	59.60
MetaSAug-LDAM [45]	80.66	84.34	89.68	48.01	52.27	61.28
Hybrid-SC [71]	81.40	85.36	91.12	46.72	51.87	63.05
ResLT [11]	82.40	85.17	89.70	48.21	52.71	62.01
Dynamic Loss [37]	82.95	88.30	91.24	50.14	54.51	63.99
BCL [83]	84.32	87.24	91.12	51.93	56.59	64.87
GLMC [16]	**87.75**	90.18	94.04	55.88	61.08	70.74
SURE	83.28	87.72	93.73	51.60	58.57	71.13
GLMC + MaxNorm [1]	87.57	**90.22**	94.03	57.11	62.32	72.33
SURE + re-weighting	86.93	**90.22**	**94.96**	**57.34**	**63.13**	**73.24**

Table 2. **Top-1 accuracy (%) of ResNet32 [28] on CIFAR10-LT and CIFAR100-LT [12] with different imbalance factors [100, 50, 10].** SURE, enhanced with re-weighting, achieves comparable top-1 accuracy to the SOTA method GLMC [16] + MaxNorm [1].

Methods	CE [78]	SELFIE [65]	PLC [78]	NCT [6]	Dynamic Loss [37]	SSR+ [17]	Jigsaw-ViT * [7]	**SURE**
Acc. (%)	79.4	81.8	83.4	84.1	86.5	88.5	89.0	**89.0**

* is with DeiT-S [70] and an extra self-supervised loss. The others are with VGG19-BN [64].

Table 3. **Comparison of SOTA approaches on learning with noisy labels task on Animal-10N [65] (noise ratio ∼8%).** Top-1 test accuracy (%) is reported.

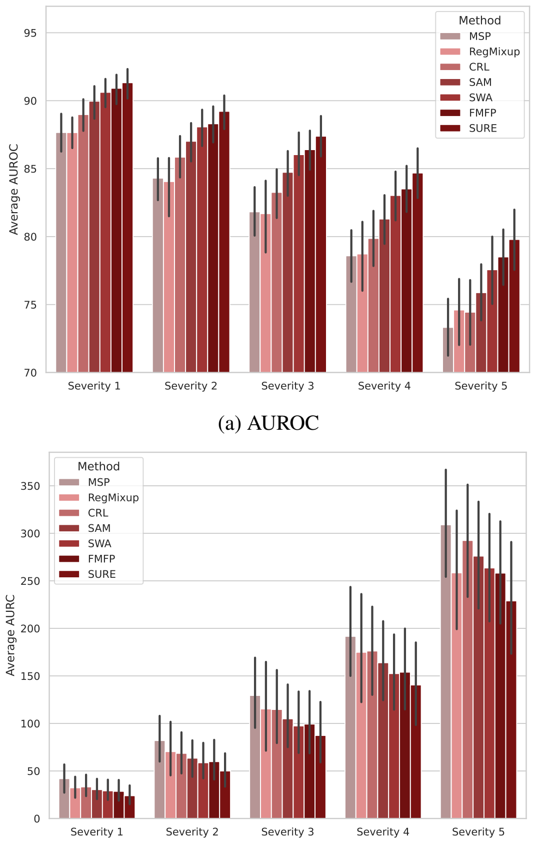
Methods	CE [78]	CleanNet [42]	MWNet [63]	SMP [27]	NRank [62]	PLC [78]	WarPI [67]	Jigsaw-ViT * [7]	**SURE**
Acc. (%)	81.7	83.5	84.7	85.1	85.2	85.3	85.9	86.7	**88.0**

* is with DeiT-S [70] and an extra self-supervised loss. The others are with ResNet-50 [28].

Table 4. **Comparison of SOTA approaches on learning with noisy labels task on Food-101N [42] (noise ratio ∼20%).** Top-1 test accuracy (%) is reported.

we use the uncertainty scores obtained from our first-stage training for re-weighting. Precisely, during the first epoch of re-weighting, we save the maximum softmax score for each sample in the training set to serve as the uncertainty score. By applying an exponential mapping to the uncertainty score. We re-weight the cross-entropy loss for each sample using e^{-s_i} and normalize the weights of all samples in a training batch such that they sum up to one. This re-weighting process is carried out over 50 epochs with a learning rate of 5e-3. Further ablation studies about variations of our re-weighting mapping are in the Supplementary Material.

Comparison to state-of-the-art approaches We also conduct fair comparison to state-of-the-art approaches on CIFAR10-LT [12] and CIFAR100-LT [12] with different imbalance factors. To make a fair comparison in this task, we train our SURE with ResNet32 [28], the most commonly used backbone in the community. The results are presented in Table 2. Although our proposed SURE is not originally designed for long-tailed classification, it achieves competitive results by equipping with the second stage uncertainty-aware re-weighting compared to task-specific solutions. The results suggest that leveraging uncertainty estimation for downstream applications is promising, especially using SURE to train the DNNs.



(a) AUROC

(b) AURC

Figure 3. **Comparison of the average AUROC [13] (higher is better) and AURC [22] (lower is better) on CIFAR10-C [30].** We use DenseNet [34] as the backbone and train on the standard CIFAR10 training set. The evaluation results are averaged across the images with 15 types of corruption under 5 severity levels.

4.4. Learning with noisy labels

For learning with noisy labels, we report top-1 test accuracy on benchmark Animal-10N [65] and Food-101N [42] in Tables 3 and 4 , respectively. On Animal-10N, our SURE outperforms the baseline trained with cross-entropy loss by 9.6%. Compared with NCT [6], which uses two backbones for training, SURE trained with only one backbone improves performance by 4.9%. Moreover, SURE achieves higher accuracy than SSR+ [17], which is designed for noisy labels employing techniques such as sample selection and relabelling. In Table 4 on Food-101N, although SURE is not designed for learning with noisy labels, with the default settings, it significantly outperforms all current

Method	Loss		Optimzation		Classifer	CIFAR100 [40]			
	λ_{crl}	λ_{mix}	SAM	SWA	CSC	Acc. ↑	AURC ↓	AUROC ↑	FPR95 ↓
Baseline(MSP)	0	0	✗	✗	✗	75.87±0.31	69.44±2.11	87.00±0.21	60.73±1.16
SAM	0	0	✓	✗	✗	76.60±0.21	62.97±1.02	87.72±0.10	59.35±0.87
SWA	0	0	✗	✓	✗	77.65±0.19	55.87±0.32	88.55±0.25	60.43±1.90
CSC	0	0	✗	✗	✓	74.05±0.18	78.14±0.26	86.82±0.24	63.56±1.20
FMFP	0	0	✓	✓	✗	77.82±0.08	55.03±0.52	88.59±0.07	59.79±0.31
SAM + CSC	0	0	✓	✗	✓	75.97±0.39	64.20±1.55	88.06±0.19	59.36±1.21
SWA + CSC	0	0	✓	✓	✗	78.46±0.33	55.68±0.41	87.74±0.44	61.22±2.54
FMFP + CSC	0	0	✓	✓	✓	78.45±0.13	54.18±0.47	88.23±0.20	60.05±1.03
CRL	1	0	✗	✗	✗	76.42±0.21	62.78±0.21	88.07±0.17	59.02±0.39
CRL + SAM	1	0	✓	✗	✗	76.98±0.32	59.71±1.39	88.26±0.07	59.52±1.92
CRL + SWA	1	0	✗	✓	✗	77.56±0.20	56.88±0.28	88.24±0.45	61.73±1.77
CRL + CSC	1	0	✗	✗	✓	75.61±0.46	67.83±1.98	87.84±0.11	59.80±2.16
CRL + FMFP	1	0	✓	✓	✗	77.71±0.54	56.24±0.89	88.21±0.44	61.75±1.74
CRL+ SAM + CSC	1	0	✓	✗	✓	78.21±0.53	53.55±3.28	88.86±0.45	56.37±1.71
CRL+ SWA + CSC	1	0	✓	✓	✗	78.09±0.10	56.61±0.91	87.78±0.21	61.37±1.56
CRL+ FMFP + CSC	1	0	✓	✓	✓	78.24±0.18	55.01±0.44	88.14±0.11	60.48±0.27
Reg	0	1	✗	✗	✗	76.99±1.19	63.09±4.22	87.71±0.13	58.78±0.50
Reg + SAM	0	1	✓	✗	✗	77.45±0.55	60.68±3.75	87.70±0.39	58.72±1.42
Reg + SWA	0	1	✗	✓	✗	78.55±0.62	52.31±2.10	88.71±0.22	58.99±2.07
Reg + CSC	0	1	✗	✗	✓	78.32±0.28	62.40±0.58	86.57±0.34	58.77±2.27
Reg + FMFP	0	1	✓	✓	✗	79.04±0.50	50.09±1.00	88.89±0.20	58.47±0.88
Reg + SAM + CSC	0	1	✓	✗	✓	78.91±0.34	57.43±2.25	87.16±0.23	58.35±0.22
Reg + SWA + CSC	0	1	✓	✓	✗	80.17±0.52	49.87±1.86	87.89±0.10	61.08±1.06
Reg + FMFP + CSC	0	1	✓	✓	✓	79.88±0.07	48.58±0.34	88.50±0.20	58.52±0.75
CRL + Reg	1	1	✗	✗	✗	78.38±0.17	52.93±1.19	**88.97±0.38**	**56.12±1.33**
CRL + Reg + SAM	1	1	✓	✗	✗	78.21±0.53	53.55±3.28	88.86±0.45	56.37±1.71
CRL + Reg + SWA	1	1	✗	✓	✗	78.64±0.16	50.96±1.01	88.96±0.31	59.27±1.47
CRL + Reg + CSC	1	1	✗	✗	✓	79.42±0.11	54.35±0.91	87.59±0.20	59.67±0.53
CRL + Reg + FMFP	1	1	✓	✓	✗	79.17±0.30	49.96±1.63	88.70±0.20	59.85±2.07
CRL + Reg + SAM + CSC	1	1	✓	✗	✓	79.10±0.34	56.39±1.25	87.44±0.16	56.98±0.31
CRL + Reg + SWA + CSC	1	1	✓	✓	✗	79.63±0.27	49.14±0.22	88.51±0.34	59.28±2.14
SURE	1	1	✓	✓	✓	**80.49±0.18**	**45.81±0.15**	88.73±0.24	58.91±0.58

Table 5. **Ablation study** of **different components** used in SURE and their **combinations** on CIFAR100 [40].

SOTAs by at least 1.3%. Results on both benchmarks verify SURE's robustness towards datasets with label noise.

4.5. Failure prediction under distribution shift

In real-world applications, environmental conditions are prone to change frequently, such as shifts in weather from sunny to cloudy and then to rainy. It's crucial for models to maintain reliable decision-making capabilities under such distribution or domain shifts. To emulate these scenarios, we evaluate our model trained with the clean training set of CIFAR10 (the same training set presents in Section 4.2) on corruption datasets CIFAR10-C [30]. We present the average AUROC and AURC of 15 corruptions for different approaches in Figure 3. Our SURE significantly enhances the failure prediction performance across a spectrum of corruptions. When compared to our baseline model, the SURE-based model demonstrates a notable improvement: the average AURC is reduced from 309 to 229. These results highlight SURE's robustness and adaptability in dynamically changing environments. Note that the performances of each corruption are presented in the Supplementary Material.

4.6. Analysis

Ablation study To further analyze SURE, we analyze the contribution of each component to our model's performance on CIFAR100 in Table 5. We report the means and standard deviations over *three* runs in our ablations with ResNet18 [28]. Starting from our baseline model, MSP, we observe the incremental impact of adding techniques like RegMixup, CRL, SAM, SWA, and the CSC to the SURE framework. Each addition to the SURE approach appears to improve accuracy and AURC, with the complete SURE method achieving the highest scores reported in the study. Among them, RegMixup and SWA contribute the most to performance, the combination of RegMixup and FMFP holds importance. This comprehensive analysis highlights the synergistic effect of our model's components, underscoring their collective importance in achieving optimal performance. Note that more analysis, such as the effect of RegMixup regularization weight λ_{mix} and CRL weight λ_{crl} are provided in the Supplementary Material.

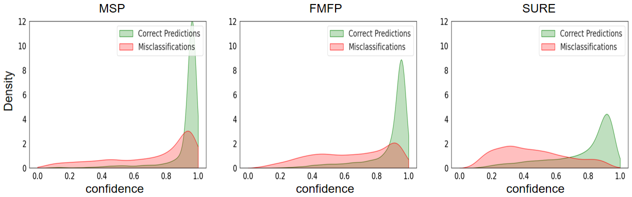


Figure 4. **The visual results of confidence separation given by different methods on CIFAR100-LT [12] IF=10.** SURE leads to better confidence separation than MSP [31] and FMFP [81].

Visualization We provide visualization of confidence distribution on CIFAR100-LT [12] IF=10 in Figure 4. From which, one can recognize SURE leads to clearly better confidence separation than MSP and FMFP. From the competitive approaches to SURE, the proposed method can increase the uncertainty of misclassified samples while improving accuracy.

5. Conclusion

In this paper, we introduce SURE, a novel framework that integrates multiple techniques for model regularization, classifier and optimization, aiming to enhance the reliability and robustness of DNNs. Our work highlights the shortcomings of existing methods when dealing with the complex nature of real-world data. This insight underlines the imperative need for approaches like SURE. Through rigorous evaluation, SURE has consistently outperformed individual methods across various datasets and model architectures in failure prediction. Moreover, its application in addressing real-world challenges, such as long-tailed classification, learning with noisy labels and data corruption, has not only yielded results comparable to state-of-the-art methods in long-tailed distribution datasets but also excelled in scenarios with label noise. This work paves the way for the application of uncertainty estimation methods in various intricate real-world situations.

Acknowledgement We thank Caizhi Zhu, Yuming Du and Yinqiang Zheng for inspiring discussions and valuable feedback.

References

[1] Shaden Alshammari, Yu-Xiong Wang, Deva Ramanan, and Shu Kong. Long-tailed recognition via weight balancing. In *CVPR*, 2022. 3, 5, 6, 7

[2] Murat Seçkin Ayhan, Laura Kühlewein, Gulnar Aliyeva, Werner Inhoffen, Focke Ziemssen, and Philipp Berens. Expert-validated estimation of diagnostic uncertainty for deep neural networks in diabetic retinopathy detection. *Medical image analysis*, 2020. 1

[3] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *NeurIPS*, 32, 2019. 3

[4] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *NeurIPS*, 2019. 3, 6, 7

[5] Haw-Shiuan Chang, Erik Learned-Miller, and Andrew McCallum. Active bias: Training more accurate neural networks by emphasizing high variance samples. In *NeurIPS*, 2017. 3

[6] Yingyi Chen, Shell Xu Hu, Xi Shen, Chunrong Ai, and Johan A. K. Suykens. Compressing features for learning with noisy labels. *TNNLS*, 2022. 3, 7

[7] Yingyi Chen, Xi Shen, Yahui Liu, Qinghua Tao, and Johan AK Suykens. Jigsaw-vit: Learning jigsaw puzzles in vision transformer. *Pattern Recognition Letters*, 2023. 2, 7

[8] Zixiang Chen, Junkai Zhang, Yiwen Kou, Xiangning Chen, Cho-Jui Hsieh, and Quanquan Gu. Why does sharpness-aware minimization generalize better than sgd? In *NeurIPS*, 2023. 4

[9] Jiwoong Choi, Dayoung Chun, Hyun Kim, and Hyuk-Jae Lee. Gaussian yolov3: An accurate and fast object detector using localization uncertainty for autonomous driving. In *ICCV*, 2019. 1

[10] Charles Corbiere, Nicolas Thome, Antoine Saporta, Tuan-Hung Vu, Matthieu Cord, and Patrick Perez. Confidence estimation via auxiliary models. *IEEE TPAMI*, 2021. 2

[11] Jiequan Cui, Shu Liu, Zhuotao Tian, Zhisheng Zhong, and Jiaya Jia. Reslt: Residual learning for long-tailed recognition. *IEEE TPAMI*, 2022. 7

[12] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *CVPR*, 2019. 2, 5, 7, 8

[13] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *ICML*, 2006. 6, 7

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 5, 6

[15] Yifan Ding, Liqiang Wang, Deliang Fan, and Boqing Gong. A semi-supervised two-stage approach to learning from noisy labels. In *WACV*, 2018. 3

[16] Fei Du, Peng Yang, Qi Jia, Fengtao Nan, Xiaoting Chen, and Yun Yang. Global and local mixture consistency cumulative learning for long-tailed visual recognitions. In *CVPR*, 2023. 3, 5, 7

[17] Chen Feng, Georgios Tzimiropoulos, and Ioannis Patras. Ssr: An efficient and robust framework for learning with unknown label noise. In *BMVC*, 2022. 3, 7

[18] Di Feng, Lars Rosenbaum, and Klaus Dietmayer. Towards safe autonomous driving: Capture uncertainty in the deep neural network for lidar 3d vehicle detection. In *2018 21st international conference on intelligent transportation systems (ITSC)*, 2018. 1

[19] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *ICLR*, 2020. 2, 3, 4, 5, 6

[20] Gianni Franchi, Xuanlong Yu, Andrei Bursuc, Emanuel Aldea, Severine Dubuisson, and David Filliat. Latent discriminant deterministic uncertainty. In *ECCV*, 2022. 2

[21] Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *NeurIPS*, 2017. 6

[22] Yonatan Geifman, Guy Uziel, and Ran El-Yaniv. Bias-reduced uncertainty estimation for deep neural classifiers. In *ICLR*, 2018. 6, 7

[23] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *CVPR*, 2018. 2, 4

[24] Gianluca Giuffrida, Luca Fanucci, Gabriele Meoni, Matej Batič, Léonie Buckley, Aubrey Dunne, Chris van Dijk, Marco Esposito, John Hefele, Nathan Vercruyssen, et al. The ϕ-sat-1 mission: The first on-board deep neural network demonstrator for satellite earth observation. *IEEE Transactions on Geoscience and Remote Sensing*, 2021. 1

[25] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *ICML*, 2017. 2

[26] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, 2018. 3

[27] Jiangfan Han, Ping Luo, and Xiaogang Wang. Deep self-learning from noisy labels. In *ICCV*, 2019. 7

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2, 6, 7, 8

[29] Wei He, Yuhao Chen, and Zhao Yin. Adaptive neural network control of an uncertain robot with full-state constraints. *IEEE transactions on cybernetics*, 2015. 1

[30] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *ICLR*, 2019. 2, 5, 7, 8

[31] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *ICLR*, 2017. 1, 2, 6, 8

[32] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. *ICLR*, 2018. 2

[33] Shell Xu Hu, Pablo G Moreno, Yang Xiao, Xi Shen, Guillaume Obozinski, Neil D Lawrence, and Andreas Damianou. Empirical bayes transductive meta-learning with synthetic gradients. In *ICLR*, 2020. 2, 4

[34] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017. 2, 6, 7

[35] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights

leads to wider optima and better generalization. *arXiv*, 2018. 2, 3, 4, 5, 6

[36] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. MentorNet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *ICML*, 2018. 3

[37] Shenwang Jiang, Jianan Li, Jizhou Zhang, Ying Wang, and Tingfa Xu. Dynamic loss for robust learning. *IEEE TPAMI*, 2023. 3, 7

[38] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *ICLR*, 2020. 3, 6

[39] Kyeongbo Kong, Junggi Lee, Youngchul Kwak, Minsung Kang, Seong Gyun Kim, and Woo-Jin Song. Recycling: Semi-supervised learning with noisy labels in deep neural networks. *IEEE Access*, 7:66998–67005, 2019. 3

[40] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Toronto, ON, Canada*, 2009. 2, 5, 6, 8

[41] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 2, 5, 6

[42] Kuang-Huei Lee, Xiaodong He, Lei Zhang, and Linjun Yang. Cleannet: Transfer learning for scalable image classifier training with label noise. In *CVPR*, 2018. 2, 5, 7

[43] Christian Leibig, Vaneeda Allken, Murat Seçkin Ayhan, Philipp Berens, and Siegfried Wahl. Leveraging uncertainty information from deep neural networks for disease detection. *Scientific reports*, 2017. 1

[44] Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *ICLR*, 2020. 3

[45] Shuang Li, Kaixiong Gong, Chi Harold Liu, Yulin Wang, Feng Qiao, and Xinjing Cheng. Metasaug: Meta semantic augmentation for long-tailed visual recognition. In *CVPR*, 2021. 7

[46] Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *ICLR*, 2017. 2

[47] Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. In *NeurIPS*, 2020. 3

[48] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *NeurIPS*, 2020. 2

[49] Antonio Loquercio, Mattia Segu, and Davide Scaramuzza. A general framework for uncertainty estimation in deep learning. *IEEE Robotics and Automation Letters*, 2020. 1

[50] Xingjun Ma, Yisen Wang, Michael E Houle, Shuo Zhou, Sarah Erfani, Shutao Xia, Sudanthi Wijewickrema, and James Bailey. Dimensionality-driven learning with noisy labels. In *ICML*, 2018. 3

[51] Eran Malach and Shai Shalev-Shwartz. Decoupling "when to update" from "how to update". In *NeurIPS*, 2017. 3

[52] Pablo Miralles, Kathiravan Thangavel, Antonio Fulvio Scannapieco, Nitya Jagadam, Prerna Baranwal, Bhavin Faldu, Ruchita Abhang, Sahil Bhatia, Sebastien Bonnart, Ishita Bhatnagar, et al. A critical review on the state-of-the-art and future prospects of machine learning for earth observation operations. *Advances in Space Research*, 2023. 1

[53] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *ICLR*, 2018. 2

[54] Jooyoung Moon, Jihyo Kim, Younghak Shin, and Sangheum Hwang. Confidence-aware learning for deep neural networks. In *ICML*, 2020. 2, 4, 6

[55] Jishnu Mukhoti, Andreas Kirsch, Joost van Amersfoort, Philip H.S. Torr, and Yarin Gal. Deep deterministic uncertainty: A new simple baseline. In *CVPR*, 2023. 2

[56] Tanya Nair, Doina Precup, Douglas L Arnold, and Tal Arbel. Exploring uncertainty measures in deep networks for multiple sclerosis lesion detection and segmentation. *Medical image analysis*, 2020. 1

[57] Kento Nishi, Yi Ding, Alex Rich, and Tobias Hollerer. Augmentation strategies for learning with noisy labels. In *CVPR*, 2021. 3

[58] Shreyas Padhy, Zachary Nado, Jie Ren, Jeremiah Liu, Jasper Snoek, and Balaji Lakshminarayanan. Revisiting one-vs-all classifiers for predictive uncertainty and out-of-distribution detection in neural networks. In *ICML Workshops*, 2020. 2

[59] Francesco Pinto, Harry Yang, Ser Nam Lim, Philip Torr, and Puneet Dokania. Using mixup as a regularizer can surprisingly improve accuracy & out-of-distribution robustness. In *NeurIPS*, 2022. 1, 2, 3, 4, 5, 6

[60] Geoff Pleiss, Tianyi Zhang, Ethan Elenberg, and Kilian Q Weinberger. Identifying mislabeled data using the area under the margin ranking. In *NeurIPS*, 2020. 3

[61] Haoxuan Qu, Lin Geng Foo, Yanchao Li, and Jun Liu. Towards more reliable confidence estimation. *IEEE TPAMI*, 2023. 2

[62] Karishma Sharma, Pinar Donmez, Enming Luo, Yan Liu, and I Zeki Yalniz. Noiserank: Unsupervised label noise reduction with dependence models. In *ECCV*, 2020. 7

[63] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. In *NeurIPS*, 2019. 7

[64] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *CVPR*, 2014. 2, 6, 7

[65] Hwanjun Song, Minseok Kim, and Jae-Gil Lee. Selfie: Refurbishing unclean samples for robust deep learning. In *ICML*, 2019. 2, 3, 5, 7

[66] Jacob Steinhardt and Percy S Liang. Unsupervised risk estimation using only conditional independence structure. *NeurIPS*, 2016. 2

[67] Haoliang Sun, Chenhui Guo, Qi Wei, Zhongyi Han, and Yilong Yin. Learning to rectify for robust learning with noisy labels. *Pattern Recognition*, 2022. 3, 7

[68] Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *CVPR*, 2018. 3

[69] Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. Long-tailed classification by keeping the good and removing the bad momentum causal effect. In *NeurIPS*, 2020. 3, 6, 7

[70] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021. 2, 5, 6, 7

[71] Peng Wang, Kai Han, Xiu-Shen Wei, Lei Zhang, and Lei Wang. Contrastive learning based hybrid networks for long-tailed image classification. In *CVPR*, 2021. 3, 7

[72] Hongxin Wei, Lei Feng, Xiangyu Chen, and Bo An. Combating noisy labels by agreement: A joint training method with co-regularization. In *CVPR*, 2020. 3

[73] Yuzhe Yang and Zhi Xu. Rethinking the value of labels for improving class-imbalanced learning. In *NeurIPS*, 2020. 3, 7

[74] Kun Yi and Jianxin Wu. Probabilistic end-to-end noise correction for learning with noisy labels. In *CVPR*, 2019. 3

[75] Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor Tsang, and Masashi Sugiyama. How does disagreement help generalization against label corruption? In *ICML*, 2019. 3

[76] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016. 2, 6

[77] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 2, 3, 7

[78] Yikai Zhang, Songzhu Zheng, Pengxiang Wu, Mayank Goswami, and Chao Chen. Learning with feature-dependent label noise: A progressive approach. In *ICLR*, 2021. 3, 7

[79] Evgenii Zheltonozhskii, Chaim Baskin, Avi Mendelson, Alex M Bronstein, and Or Litany. Contrast to divide: Self-supervised pre-training for learning with noisy labels. In *WACV*, 2022. 3

[80] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *CVPR*, 2020. 3, 6, 7

[81] Fei Zhu, Zhen Cheng, Xu-Yao Zhang, and Cheng-Lin Liu. Rethinking confidence calibration for failure prediction. In *ECCV*, 2022. 2, 4, 5, 6, 8

[82] Fei Zhu, Zhen Cheng, Xu-Yao Zhang, and Cheng-Lin Liu. Openmix: Exploring outlier samples for misclassification detection. In *CVPR*, 2023. 2, 5, 6

[83] Jianggang Zhu, Zheng Wang, Jingjing Chen, Yi-Ping Phoebe Chen, and Yu-Gang Jiang. Balanced contrastive learning for long-tailed visual recognition. In *CVPR*, 2022. 3, 5, 7