Boosting Class Representation via Semantically Related Instances for Robust Long-Tailed Learning with Noisy Labels

Anonymous ICCV submission

Paper ID 11573

Abstract

*The problem of learning from long-tailed noisy data, referred to as Long-Tailed Noisy Label Learning (LTNLL), presents significant challenges in deep learning. LTNLL datasets are typically affected by two primary issues: class imbalance and label noise. While previous methods have addressed these problems separately, the simultaneous presence of both in real-world applications remains underexplored. In this paper, we introduce a simple yet effective method, **I**nstances **B**enefitting **C**lasses (IBC). Our philosophy is to simultaneously overcome overfitting to noisy classes and transfer knowledge between semantically related classes. At the instance level, we propose selecting top-k semantically similar classes and use them to construct soft labels. Specifically, we soften noisy hard labels by reducing the probability of noisy classes and reallocating this probability to the semantically similar classes.* ***This reduces the model's overconfidence in noisy classes while enhancing its focus on tail classes.*** *We next propose a novel shot-specific multi-expert ensemble learning framework to make knowledge transfer more targeted, where we maintain multiple shot-specific soft labels for each instance, with each expert supervised by one of these labels. By integrating these experts, we demonstrate that IBC exhibits more separable representations, improving both overall and partition performance. Extensive experiments show that IBC outperforms existing state-of-the-art (SOTA) methods on a variety of benchmark and real-world datasets, achieving improvements ranging from **1.89%** to **4.99%** on the CIFAR-10 and CIFAR-100 datasets across all settings. The source code is provided in the supplementary material.*

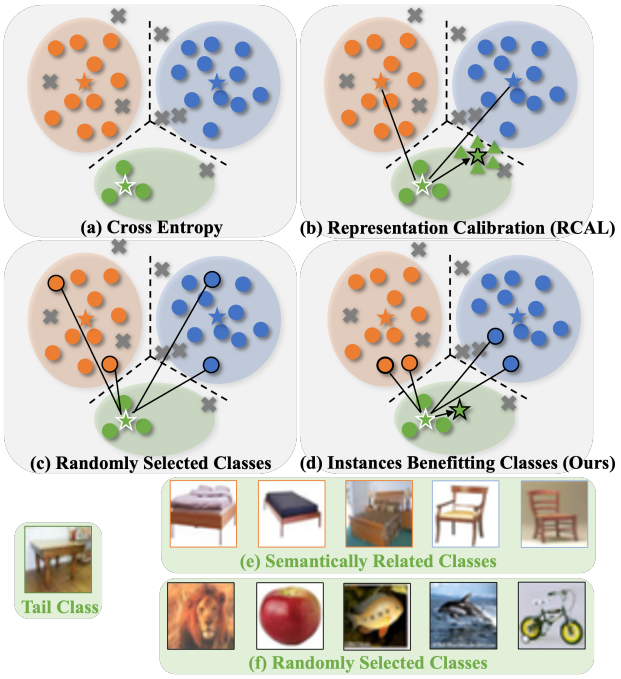

Figure 1. Overall comparison with previous methodologies. (a) Under-represented tail classes; (b) **RCAL** [37] leverages head classes for calibrating the macro-statistics (means and covariances) of tail classes, leading to over-calibration; (c)/(f) A simplification of **Label Smoothing**: Instances are leveraged to boost randomly selected classes, which provides minimal benefit to the representations. (d)/(e) Our method, **IBC**, boosts class representation through semantically related instances, leading to improved representation for tail classes.

1. Introduction

Deep learning has significantly advanced many areas of computer vision over the past few years, largely due to large-scale, well-annotated datasets [29]. However, the data required to train deep neural networks (DNNs) is often far from ideal. In particular, real-world datasets commonly face two key challenges: (1) *class imbalance*, or long-tailed distribution, where the majority classes (head classes) dominate the data, while minority classes (tail classes) contain only a limited number of training instances. This imbalance causes DNNs to focus too heavily on head classes, resulting in poor generalization on tail classes [6, 40]. (2) *Label noise*, often introduced through manual labeling methods

such as crowdsourcing, inevitably leads to labeling errors. Due to the large number of parameters in DNNs, they are prone to overfitting these noisy labels, which harms generalization [3, 17, 20, 25].

In this work, we focus on the scenario where both label noise and long-tailed distribution coexist, which we refer to as *Long-Tailed Noisy Label Learning* (LTNLL). Existing LTNLL methods can be broadly categorized into two groups: discriminative methods [7, 9, 18, 28] and representational methods [2, 36, 37]. Discriminative methods aim to distinguish clean data from mislabeled data in the presence of noisy labels. Several metrics and techniques designed specifically for long-tailed distribution have been proposed to improve sample selection. In contrast, representational methods focus on enhancing the quality of representations for long-tailed noisy data using techniques such as augmentation, calibration, and regularization. These approaches seek to mitigate the effects of label noise and class imbalance, promoting better representation learning in a unified framework.

Among the LTNLL methods, SFA [11] and RCAL [37] are the current SOTA. SFA utilizes a distance-based sample selection algorithm to identify clean instances that guide the training process. Since class prototypes, calculated using inaccurate supervision, are unreliable, SFA initially selects high-confidence instances to compute the instantaneous class prototypes and then updates them through a running average. However, this approach overlooks valuable information from low-confidence tail class instances, particularly when the imbalance ratio is high. Additionally, SFA does not apply targeted operations for tail classes, limiting its ability to improve representation learning. On the other hand, RCAL uses a representation calibration framework, adjusting the means and covariances of tail classes with a weighted average of their nearest head classes. This calibration mainly focuses on adjusting macro-statistics, but most head class instances share little similarity with tail classes. Therefore, directly using all head class instances to calibrate tail classes will inevitably leads to over-calibration, as shown in Fig. 1 (b). This introduces false correlations between head and tail classes, undermining its ability to generate high-quality representations. Although SFA and RCAL are prominent LTNLL methods, the issues outlined above persist, motivating our research to address these challenges.

Meanwhile, an interesting observation in multi-class image classification is that visual patterns can be shared across different classes [34]. For instance, the visual pattern "legged structures" is shared by "chair," "bed," and "table", as shown in Fig. 1 (e). **This observation inspires transfer learning, where instances from one class can aid in learning representations for other classes, provided they share similar semantic patterns.** Previous studies have explored this idea using two main approaches: (1) Raw data augmentation, such as CSA [24], which introduces a context shift augmentation module to generate diverse training images for tail classes by utilizing a context bank from head-class images, and (2) High-dimensional feature calibration, with RCAL [37] being a notable example. **However, there has been a lack of efforts focused on mining label space information, specifically the training labels, which we argue are just as crucial as the data itself. In this study, we aim to boost class representations by incorporating instances from other semantically related classes, moving beyond simply using head classes to support tail classes and extending the concept more broadly.**

In this paper, we introduce *Instances Benefitting Classes* (IBC), a simple yet effective method that addresses both label noise and long-tailed distribution. IBC softens noisy one-hot labels by reducing the probability of the noisy class and reallocating it to semantically related classes. Unlike RCAL [37], IBC strengthens each tail class using its nearest instances, which share similar visual patterns that represent the tail class effectively, as shown in Fig. 1 (d)/(e). While IBC and label smoothing (LS) [26] may appear similar, LS randomly selects classes that do not contribute to representation, as shown in Fig. 1 (c). We further compare the key distinction between IBC and LS in Section 3.5. A holistic comparison with several baselines is presented in Fig. 1.

IBC starts by using MoCo [5], a self-supervised contrastive learning method, to provide base representations for all classes. Unlike SFA's stochastic feature [11], MoCo's unsupervised nature makes its representations more robust. In each training epoch, IBC softens the original one-hot noisy labels as described above. To focus on specific shots and make knowledge transfer more targeted, we propose a shot-specific multi-expert framework that maintains multiple shot-specific soft labels for each instance, with each expert supervised by one of these labels. Additionally, a distance-based method is employed to distinguish clean instances from noisy ones, and MixMatch [1] is applied to handle noisy instances. Our key contributions are as follows:

- *A novel label softening method.* Focusing on label space rectification, IBC softens noisy hard labels by leveraging semantic similarity. This approach simultaneously addresses label noise and class imbalance in a unified manner, offering valuable insights for solving the LTNLL problem.
- *A novel shot-specific ensemble framework.* We introduce a shot-specific multi-expert ensemble learning framework to make knowledge transfer more targeted. By integrating these experts, we demonstrate that IBC exhibits more separable representations, improving both overall and partition performance.
- *SOTA performance.* Extensive experiments demonstrate IBC's superiority over existing *Noisy Label Learning*

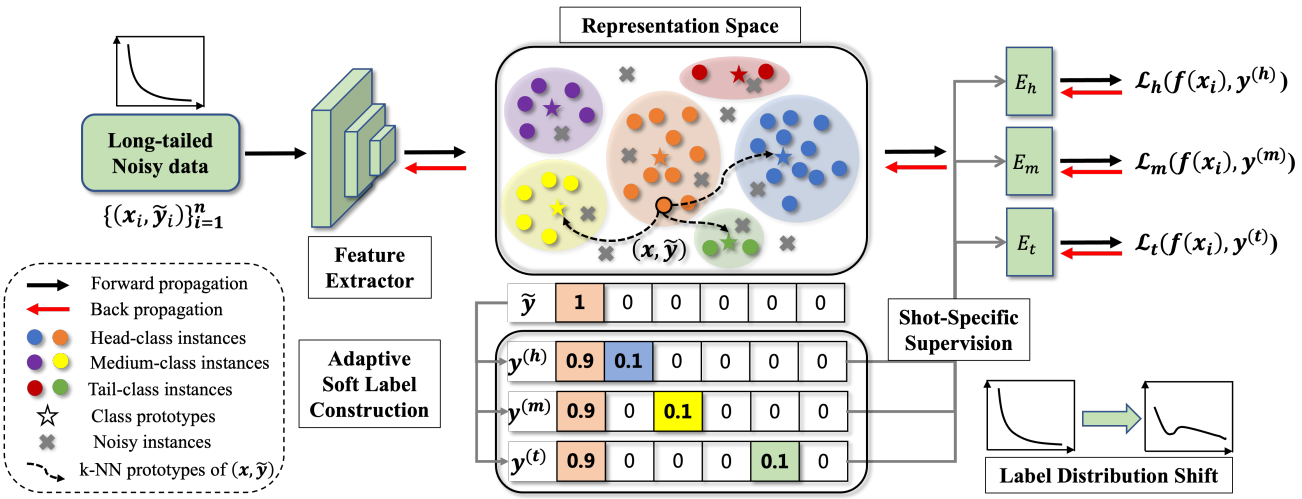

Figure 2. The overall framework of IBC. The primary operations occur in the representation and label spaces. By leveraging semantic information, we adaptively soften the one-hot noisy labels of each instance. To focus on each shot, IBC constructs three shot-specific classifiers, each supervised by shot-specific soft labels. This approach focuses on each shot individually, and the use of IBC soft labels facilitates a label distribution shift, leading to a more balanced label distribution in practice.

(NLL), *Long-tailed Learning* (LTL), and LTNLL methods, achieving improvements ranging from **1.89%** to **4.99%** on the CIFAR-10 and CIFAR-100 datasets across all settings. Additionally, we provide a comprehensive analysis and ablation studies, further validating IBC's effectiveness.

2. Methodology

2.1. Preliminaries

Problem Setup. Let $\mathcal{X} \subseteq \mathbb{R}^d$ denote the input space, and $\mathcal{Y} = \{1, \dots, K\}$ the label space, where $K \geq 2$ is the number of classes. We consider a training set $\tilde{\mathcal{D}} = \{(x_i, \tilde{y}_i)\}_{i=1}^n$, where $x_i \in \mathcal{X}$ denotes the i-th input instance, and $\tilde{y}_i \in \mathcal{Y}$ is its observed noisy label, while the true label y_i remains unobserved. Without loss of generality, we assume that the class distribution follows a long-tailed distribution which satisfies: $n_1 \geq n_2 \geq \cdots \geq n_K$, where $n_k = |\{i : \tilde{y}_i = k\}|$ represents the number of instances in class k. The classes are divided into three groups based on frequency: head ($\mathcal{G}_h = \{k \leq \tau_1\}$), medium ($\mathcal{G}_m = \{\tau_1 < k \leq \tau_2\}$), and tail ($\mathcal{G}_t = \{k > \tau_2\}$), with dataset-specific thresholds τ_1 and τ_2.

Our objective is to train a multi-class classifier $f : \mathcal{X} \to \mathcal{Y}$ that minimizes the generalization error across all classes while tackling both label noise and the under-representation of tail classes.

Overview. The proposed IBC framework is illustrated in Fig. 2 and the pseudo-code of IBC is presented in Algorithm 1. IBC consists of two main components: (1) **Adaptive Soft Labels Construction**, which redistributes probabilities based on instance-prototype similarity and class groupings, and (2) **Shot-Specific Multi-Expert Ensemble Learning**, which uses shot-specific soft label supervision and associated training losses.

2.2. MoCo-based Feature Extraction

To obtain noise-robust features, we pre-train the backbone using Momentum Contrast (MoCo) [5] on the noisy long-tailed dataset. Let $f_\theta : \mathcal{X} \to \mathbb{R}^d$ represent the feature encoder, and f_ξ its momentum counterpart, updated via:

$$\xi \leftarrow m \cdot \xi + (1 - m) \cdot \theta, \tag{1}$$

where $m \in [0, 1)$ is the momentum coefficient. Given an input x_i, we extract the normalized feature:

$$\mathbf{h}_i = \frac{f_\theta(x_i)}{|f_\theta(x_i)|_2} \in \mathbb{R}^d. \tag{2}$$

Since MoCo is trained with unsupervised contrastive learning, it generates features that are less sensitive to label noise and class imbalance.

2.3. Dynamic Class Prototype Computation

We compute the prototype $\mathbf{C}_k \in \mathbb{R}^d$ for each class $k \in \mathcal{Y}$ as the exponential moving average (EMA) of features assigned to class k:

$$\mathbf{C}_k^{(t)} = \mu \mathbf{C}_k^{(t-1)} + (1 - \mu) \frac{1}{|\mathcal{B}_k^{(t)}|} \sum_{i \in \mathcal{B}_k^{(t)}} \mathbf{h}_i^{(t)}, \tag{3}$$

where $\mu \in [0, 1)$ is the momentum coefficient, $\mathcal{B}_k^{(t)}$ is the set of instances in the current batch predicted as class k,

and $\mathbf{h}_i^{(t)}$ is the feature of x_i at iteration t. The initial prototypes $\mathbf{C}_k^{(0)}$ are computed from the MoCo features of instances with observed label $\tilde{y}_i = k$.

2.4. Distance-based Sample Selection

We propose applying targeted operations for noisy and clean instances separately. A distance-based sample selection strategy is used to effectively distinguish clean instances from noisy ones.

At the beginning of each epoch, we calculate the Euclidean distance between $\mathbf{C}_k^{(t)}$ and the feature vectors of class k instances:

$$\text{dist}(\mathbf{C}_k^{(t)}, x_i) = \|\mathbf{C}_k^{(t)} - \mathbf{h}_i\|_2^2. \tag{4}$$

These distances are then modeled using a two-component Gaussian mixture model (GMM) [22]:

$$\text{dist} \sim \sum_{j=1}^{2} \phi_j \mathcal{N}(\mu_j, \sigma_j^2), \tag{5}$$

where μ_j and σ_j^2 are the mean and variance of the j-th Gaussian component. Assuming $\mu_1 < \mu_2$, the probability that an instance x_i is clean is given by:

$$P(\text{clean}|x_i) = \frac{\phi_1 \mathcal{N}(\mu_1, \sigma_1^2)}{\sum_{j=1}^{2} \phi_j \mathcal{N}(\mu_j, \sigma_j^2)}. \tag{6}$$

Instances are classified as clean if $P(\text{clean}|x_i) > 0.5$; otherwise, they are considered noisy. The training dataset is then divided into clean and noisy subsets: $\tilde{\mathcal{D}}_{\text{clean}}$ and $\tilde{\mathcal{D}}_{\text{noisy}}$.

To exploit noisy instances, we treat them as unlabeled data and apply the semi-supervised learning (SSL) framework MixMatch [1].

For clean instances, we propose the following shot-specific soft label construction and multi-expert ensemble learning framework.

2.5. Instance-Prototype Similarity (IPS) Selection

For each instance $x_i \in \tilde{\mathcal{D}}_{\text{clean}}$, we compute its cosine similarity to all class prototypes:

$$S_{ik} = \frac{\mathbf{h}_i \cdot \mathbf{C}_k}{|\mathbf{h}_i||\mathbf{C}_k|}, \quad \forall k \in \mathcal{Y}. \tag{7}$$

The top-k nearest neighbors are then selected based on these similarities:

$$\mathcal{N}_i = \underset{k \in \mathcal{Y}}{\text{top-}k}\ S_{ik}. \tag{8}$$

These neighbors are divided into three shot groups: $\mathcal{N}_i^{(g)}$ for $g \in \{h, m, t\}$.

The goal of IPS is to identify semantically related classes for each instance in the high-dimensional feature space. Note that we compute instance-prototype similarity rather than instance-instance similarity because, in a well-constructed representation space, an instance and its k nearest neighbors are more likely to share the same ground-truth class, which offers limited benefits for knowledge transfer between classes. We ablate this component in Sec. 3.5.

2.6. Adaptive Shot-specific Soft Label Construction

The goal of constructing soft labels is to simultaneously overcome overfitting to noisy classes and facilitate knowledge transfer between semantically related classes. The first intuitive idea comes from LS [26], which reduces the probability of the noisy hard label and redistributes the probability evenly among the remaining labels. However, this strategy has limited impact on addressing long-tailed label distribution as it does not alter the distribution of the soft labels themselves. This motivates us to focus on specific shots of the labels and redistribute the probability more effectively.

We propose to redistribute the reduced probability to classes with top-k instance-prototype similarity (IPS), weighted by similarity. In other words, the more semantically similar a class is to the instance, the more probability it will receive in the constructed soft label.

The second idea comes from ensemble learning. Specifically, we train multiple experts, each focused on a specific shot $g \in \{h, m, t\}$. Previous methods [11] mainly use the same hard noisy label as supervision for training these experts, which inadvertently causes the classifier to bias towards inaccurate noisy label distributions. However, in our approach, the soft label provides more accurate and stronger supervision. It naturally enhances the representation of the redistributed classes, as demonstrated in Fig. 7. Therefore, we propose using shot-specific soft labels to supervise the learning of shot-specific experts. The shot-specific soft labels are constructed as follows.

We construct soft labels for each shot group $g \in \{h, m, t\}$ by transferring probability mass from the noisy label $\tilde{y}_i$ to semantically related classes in $\mathcal{N}_i^{(g)}$:

$$\mathbf{y}_i^{(g)} = (1 - \delta)\mathbf{e}_{\tilde{y}_i} + \delta \sum_{k \in \mathcal{N}_i^{(g)}} \frac{S_{ik}}{\sum_{j \in \mathcal{N}_i^{(g)}} S_{ij}} \mathbf{e}_k, \tag{9}$$

where $\delta \in [0, 1]$ controls the degree of probability redistribution, and $\mathbf{e}_k$ is the one-hot vector for class k. This produces three shot-specific soft labels: $\mathbf{y}_i^{(h)}$, $\mathbf{y}_i^{(m)}$, and $\mathbf{y}_i^{(t)}$ for each instance.

2.7. Shot-specific Multi-Expert Ensemble Learning

We train three expert classifiers E_h, E_m, E_t, all sharing the same feature encoder f_θ, but each specialized for different class groups. Each expert is supervised by the corresponding shot-specific soft labels, with the following loss functions:

Head-class expert E_h: Let n_c denote the size of the clean instances set $\tilde{\mathcal{D}}_{\text{clean}}$. The head-class expert is trained using standard cross-entropy loss:

$$\mathcal{L}_h = -\frac{1}{n}\sum_{i=1}^{n_c} \mathbf{y}_i^{(h)} \cdot \log \sigma \left(E_h(\mathbf{f}_i)\right), \tag{10}$$

where σ is the softmax function.

Medium-class expert E_m: For the medium-class expert, we use a balanced cross-entropy loss [23] that adjusts the logits based on label frequency:

$$\mathcal{L}_m = -\frac{1}{n}\sum_{i=1}^{n_c}\sum_{k=1}^{K} \mathbf{y}_{ik}^{(m)} \log \sigma \left(E_m^{(k)}(\mathbf{f}_i + \log n_k)\right), \tag{11}$$

where $\mathbf{y}_{ik}^{(m)}$ denotes the k-th element of $\mathbf{y}_i^{(m)}$ and $E_m^{(k)}(\cdot)$ represents the k-th element of the output vector from E_m.

Tail-class expert E_t: The tail-class expert is trained using a modified balanced cross-entropy loss that emphasizes tail-class instances by replacing n_k with n_k^2:

$$\mathcal{L}_t = -\frac{1}{n}\sum_{i=1}^{n_c}\sum_{k=1}^{K} \mathbf{y}_{ik}^{(t)} \log \sigma \left(E_t^{(k)}(\mathbf{f}_i + \log n_k^2)\right). \tag{12}$$

Dynamic Soft Label Frequency. In contrast to previous methods, we compute the label frequency using soft labels rather than noisy counts:

$$n_k^{(g)} = \sum_{i=1}^{n} \mathbf{y}_{ik}^{(g)}, \tag{13}$$

where $n_k^{(g)}$ represents the shot-specific label frequency derived from $\mathbf{y}_{ik}^{(g)}$. This approach provides a more accurate estimate of the class distribution, aligning better with the corresponding soft supervision.

The soft labels in IBC facilitate targeted knowledge transfer within each shot group, enhancing performance and generalization across varying class frequencies.

3. Experiments

3.1. Baselines

We compare the performance of IBC with various baseline methods with same backbone architecture, divided into three types: (1) Noisy label learning (NLL): DivideMix [13], DISC [14]; (2) Long-tailed learning (LTL): cRT [10], RIDE [30], SADE [39], DSCL [34]; (3) Long-tailed noisy label learning (LTNLL): HAR [2], RoLT+ [32], PCL [28], RCAL [37], RCAL+ [37], TABASCO [18], SFA [11], OT [15]. The technical details of the above baselines are provided in supplementary material. All experiments are run on NVIDIA RTX 4090 GPUs for fair comparisons.

Algorithm 1 Pseudo-code of IBC

Input: training dataset $\tilde{\mathcal{D}} = \{(x_i, y_i)\}_{i=1}^{n}$, encoder network f, classifiers E_h, E_m, E_t, holistic model parameter Θ, pre-training epochs T_p, total training epochs T_{max}, redistribution strength δ and k-nearest neighbor selection parameter k.
for $t = 1, \ldots, T_p$ **do**
  Pre-train the encoder network f with MoCo [5].
end for
for $t = 0, \ldots, T_{max}$ **do**
  for $k = 1, \ldots, K$ **do**
    Compute class prototypes $\mathbf{C}_k^{(t)}$ by Eq. (3).
    Compute Euclidean distance $dist(\mathbf{C}_k^{(t)}, x_i)$ by Eq. (4).
    Obtain $\mathcal{D}_{\text{clean}}^k$ and $\mathcal{D}_{\text{noisy}}^k$ by Eq. (6).
  end for
  $\mathcal{D}_{\text{clean}} = \bigcup_{k=1}^{K} \mathcal{D}_{\text{clean}}^k$, $\mathcal{D}_{\text{noisy}} = \bigcup_{k=1}^{K} \mathcal{D}_{\text{noisy}}^k$.
  Compute cosine similarity and k-NN prototypes for each instance by Eq. (7) and Eq. (8).
  Construct three shot-specific soft labels: $\mathbf{y}_i^{(h)}$, $\mathbf{y}_i^{(m)}$, $\mathbf{y}_i^{(t)}$ for each instance by Eq. (9).
  Forward clean instances losses $\mathcal{L}_{\text{clean}} = \mathcal{L}_h + \mathcal{L}_m + \mathcal{L}_t$ by Eq. (10) $\sim$ Eq. (12).
  Forward noisy instances losses $\mathcal{L}_{\text{noisy}} = \text{MixMatch}(\mathcal{D}_{\text{clean}}, \mathcal{D}_{\text{noisy}}, f)$
  Backward total loss $\mathcal{L} = \mathcal{L}_{\text{clean}} + \mathcal{L}_{\text{noisy}}$.
  Update parameters: $\Theta_t = \text{SGD}(\mathcal{L}, \Theta_{t-1})$.
end for
return parameters of Θ.

3.2. Datasets and Implementation Details

The data processing follows SFA [11] and RCAL [37]. Due to the page limit, we have placed this section, as well as **related literature**, in the supplementary material.

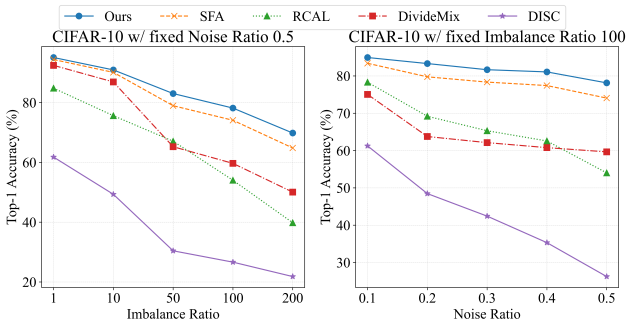

Figure 3. Performance degradation trends compared to other methods. The left illustrates cases with noise ratio fixed at 0.5. The right shows cases with imbalance ratio held constant at 100.

Table 1. Comparison of top-1 test accuracies (%) between different methods on simulated CIFAR-10/100 datasets under different noises and imbalance ratios. The best results are highlighted in bold and the second best underlined (The same applies hereinafter).

Dataset	CIFAR-10						CIFAR-100					
Noise Ratio	0.2			0.5			0.2			0.5		
Imbalance Ratio	50	100	200	50	100	200	50	100	200	50	100	200
Cross Entropy	65.13	56.64	49.87	33.80	29.36	25.85	38.01	33.08	30.92	20.08	18.32	15.92
DivideMix	73.48	63.76	53.65	65.22	59.65	50.03	48.37	42.59	33.21	35.72	31.05	21.33
DISC	62.69	51.83	36.46	49.48	41.12	31.02	43.83	37.93	26.93	32.74	27.22	16.63
cRT	67.94	58.67	46.33	41.58	33.86	26.25	30.43	23.97	16.89	19.32	14.82	10.89
RIDE	63.73	60.41	52.67	31.41	28.95	25.74	40.69	30.28	23.37	24.22	24.16	14.72
SADE	68.39	64.91	56.48	31.27	30.42	26.98	47.38	33.75	25.45	25.51	27.24	16.87
DSCL	55.89	48.46	41.39	30.41	26.62	24.83	29.21	25.86	23.28	17.65	16.16	14.11
PCL	83.71	78.34	69.15	71.44	64.69	58.93	51.46	47.12	36.25	42.51	38.36	27.94
TABASCO	82.31	74.38	66.25	71.05	62.56	57.67	47.38	41.62	33.76	39.28	33.42	27.58
RCAL	82.80	77.19	69.43	72.06	64.01	59.80	51.99	46.68	32.38	40.52	35.04	29.89
SFA	84.80	79.22	72.59	78.93	74.06	64.81	53.10	47.73	40.25	43.41	39.73	32.85
OT	–	79.76	–	–	72.86	–	–	44.80	–	–	39.11	–
IBC	**86.69**$_{\uparrow 1.89}$	**83.32**$_{\uparrow 3.56}$	**74.77**$_{\uparrow 2.18}$	**83.00**$_{\uparrow 4.07}$	**78.17**$_{\uparrow 4.11}$	**69.80**$_{\uparrow 4.99}$	**56.12**$_{\uparrow 3.02}$	**49.65**$_{\uparrow 1.92}$	**42.27**$_{\uparrow 2.02}$	**46.21**$_{\uparrow 2.80}$	**41.96**$_{\uparrow 2.23}$	**35.60**$_{\uparrow 2.75}$

3.3. Results on Simulated CIFAR-10/100

The top-1 test accuracies on the simulated CIFAR-10 and CIFAR-100 datasets are presented in Table 1. IBC achieves SOTA performance across all settings on both datasets, outperforming all baselines by significant margins, ranging from **+1.89%** to **+4.99%**. Notably, IBC demonstrates remarkable effectiveness in challenging conditions, with improvements of **+4.99%** on CIFAR-10 and **+2.75%** on CIFAR-100, when both the noise ratio and imbalance ratio are set to 0.5 and 100, respectively. Additionally, NLL methods typically perform poorly under high class-imbalance scenarios, while LTL methods struggle in the presence of high label noise. In contrast, IBC remains consistently robust across all configurations, even when compared to existing LTNLL methods, showing minimal performance degradation as noise and imbalance ratios increase. This advantage is clearly illustrated in Fig. 3.

3.4. Results on Real-world Noisy and Imbalanced Datasets

The top-1 and top-5 test accuracies on the WebVision datasets are presented in Table 2. We evaluate the performance of the model trained on the WebVision dataset using two validation sets: WebVision-50 and ILSVRC12. As shown in the table, under the original setting (**IR** $\approx$ 6.78), IBC outperforms the previous SOTA by an average of **+1.24%** and **+0.93%** on the WebVision-50 and ILSVRC12 validation sets, respectively. This further highlights the effectiveness of IBC in real-world scenarios. Notably, IBC results in a more substantial improvement in top-5 accuracy compared to top-1 accuracy. This can be attributed to the adaptive enhancement of labels, which identifies semantically similar classes—not necessarily the most similar class, but one of the most similar. While this step may not always retrieve the true label in LTNLL settings, it helps mitigate false correlations in the model. dditionally, we manually increased the imbalance ratio on the mini-WebVision training set. The results indicate that IBC continues to outperform previous SOTA methods, achieving an average improvement of **+2.51%** and **+2.22%** on the WebVision-50 and ILSVRC12 validation sets, respectively, when the imbalance ratio is set to 100.

Additionally, the comparison results on the Clothing1M dataset, shown in Table 3, demonstrate a **+1.18%** improvement, further confirming the superiority of IBC.

Table 2. Comparison of top-1 and top-5 test accuracies (%) between different methods on mini-WebVision datasets under different imbalance ratios. **IR** is short for imbalance ratio. WebVision-50 and ILSVRC12 denote different testing set.

IR	**Method**	**WebVision-50**		**ILSVRC12**	
		top-1	**top-5**	**top-1**	**top-5**
$\approx$ 6.78	DivideMix	77.32	91.68	75.20	90.84
	HAR	75.50	90.70	70.30	90.00
	RoLT+	77.64	92.44	74.64	92.48
	PCL	77.32	92.60	75.12	91.92
	RCAL	76.68	92.89	73.57	93.25
	SFA	78.96	93.00	76.16	92.68
	IBC	**79.76**	**94.68**	**77.25**	**94.02**
50	DivideMix	64.56	83.56	62.68	85.24
	PCL	68.00	88.44	65.00	86.32
	SFA	70.64	89.96	69.04	90.36
	IBC	**72.18**	**92.71**	**70.38**	**92.53**
100	DivideMix	55.76	73.48	53.92	74.00
	PCL	62.12	85.88	59.60	84.20
	SFA	65.68	88.52	65.08	88.92
	IBC	**68.35**	**90.87**	**67.47**	**90.96**

Table 3. Comparison of top-1 test accuracies (%) between various NLL methods on Clothing1M datasets.

Method	top-1	Method	top-1
MentorNet [8]	67.25	CDR [33]	68.25
Forward [21]	69.84	D2L [19]	69.74
Joint [27]	72.23	GCE [41]	69.75
Pencil [35]	73.49	LRT [42]	71.74
SL [31]	71.02	MLNT [12]	73.74
PLC [38]	74.02	DivideMix [13]	74.76
ELR+ [16]	74.81	RCAL+ [37]	74.97
Co-teaching [4]	67.94	**IBC**	**76.15**

3.5. Ablation Study and Further Analysis

All the components contribute to performance gains. We tested the contribution of each component of IBC by comparing its performance when specific components were removed. The results, presented in Table 4, were obtained on CIFAR-10 and CIFAR-100 datasets, with a noise ratio of 0.5 and an imbalance ratio of 100. To better understand the role of each module, we also report performance across different partitions. The following components were ablated:

- **"SL"**: Soft labels (SL).
- **"IPS"**: Instance-Prototype Similarity (IPS).
- **"SRI"**: Semantically Related Instances (SRI).
- **"SME"**: Shot-specific Multi-Experts Ensemble (SME).
- **"SSL"**: Semi-Supervised Learning (SSL).

As shown in Table 4, training with our soft labels (**SL**) consistently contribute to performance gains across all partitions and overall classification performance, with an average improvement of **+3.58%** on CIFAR-10 and CIFAR-100. Also, Instance-prototype similarity (**IPS**) ensures the diversity of redistribution classes, leading to significant improvements. Furthermore, we observe that the shot-specific multi-experts ensemble learning (**SME**) module positively contributes to the learning process, enhancing medium-shot and few-shot performance without compromising many-shot accuracy. Specifically, SME accurately adapts to the distribution of soft labels, enabling better overall model performance. While the performance of IBC on specific shots may not always be optimal, the overall performance remains optimal across all settings. Finally, IBC works effectively with the **SSL** method MixMatch, which fully exploits noisy instances, further boosting overall performance with an average improvement of **+2.38%**.

Compared to randomly selected classes (Label Smoothing), IBC produces a more separable representation space. Our soft label construction fundamentally differs from the evenly "softening" used in label smoothing. Fig. 4 and Fig. 5 show t-SNE plots of the CIFAR-10 training and testing sets, with a noise ratio of 0.5 and an imbalance ratio of 100, in the feature space learned by IBC and label smoothing. As shown in the figures, IBC results in a more separable feature space, particularly for tail classes.

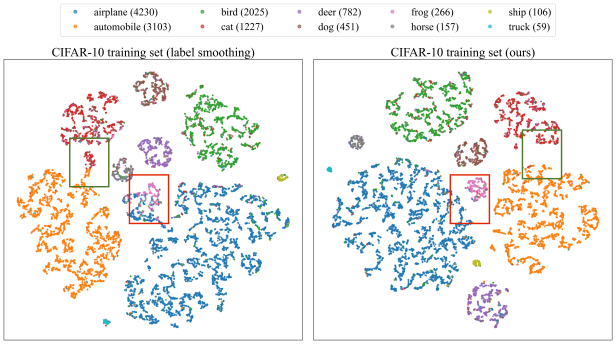

Figure 4. T-SNE visualization of the training set. The left shows features achieved by label smoothing while the right illustrates IBC's. IBC obviously achieves more separable representations.

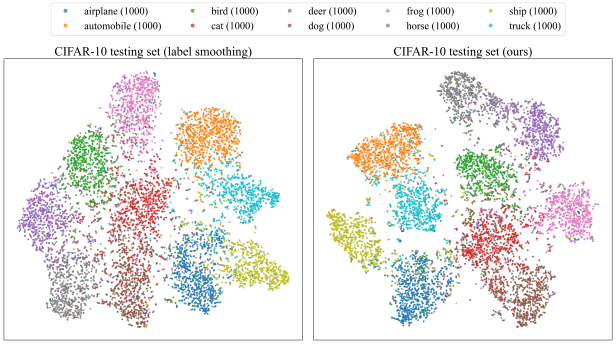

Figure 5. T-SNE visualization of the testing set. The left shows features achieved by label smoothing while the right illustrates IBC's. IBC obviously achieves more separable representations.

Semantically related instances work better than randomly selected instances. In general, IBC leverages semantically related instances (**SRI**) for boosting the representation of classes. A natural question arises: **How about replacing the semantically related instances with randomly selected instances?** As shown in Fig. 6, IBC consistently outperforms method with randomly selected classes, especially when $k = 30$, where we achieve an overall improvement of **+1.10%** and a **+2.10%** improvement for tail classes.

It is also noteworthy that when all classes are randomly selected, except for the hard noisy label class, the random selection method degenerates into label smoothing. The performance of label smoothing is indicated with horizontal green lines. Furthermore, Table 4, line "IBC w/o SRI", provides additional results on CIFAR-10. **As observed, randomly selected instances significantly reduce overall performance, particularly for tail classes. This high-**

Table 4. Comparison of top-1 test accuracies (%) between degenerated methods in two settings.

Datasets	CIFAR-10				CIFAR-100			
Partitions	Many	Medium	Few	All	Many	Medium	Few	All
IBC	**86.20**	**76.78**	72.00	**78.17**	64.27	**42.05**	19.53	**41.96**
IBC w/o SL	79.87	74.26	68.32	74.16	63.41	38.42	14.76	38.82
IBC w/o IPS	84.80	74.56	68.42	75.79	64.11	39.65	15.06	39.61
IBC w/o SRI	85.83	75.21	68.36	76.34	**64.97**	39.81	16.78	40.45
IBC w/o SME	81.95	75.87	**72.49**	76.68	61.67	41.93	**20.02**	41.28
IBC w/o SSL	84.01	73.46	70.31	75.68	63.01	39.25	17.02	39.70

lights the critical importance of using semantically related instances.

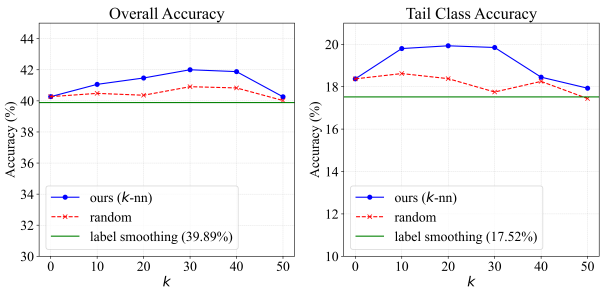

Figure 6. Curves of top-1 test accuracies of IBC and method with randomly selected classes under various k values on CIFAR-100 with a noise ratio of 0.5 and an imbalance ratio of 100. The left is overall accuracy and the right focus the few-shot (tail) accuracy.

Shot-specific multi-experts ensemble learning improves classification accuracy as a whole. To further evaluate the effectiveness of our shot-specific multi-experts ensemble component, we examine degenerated versions of IBC:

- **"Enhance head only"**: Train multi-experts using soft labels $\mathbf{y}^{(h)}$ for the head class only.
- **"Enhance medium only"**: Train multi-experts using soft labels $\mathbf{y}^{(m)}$ for the medium class only.
- **"Enhance tail only"**: Train multi-experts using soft labels $\mathbf{y}^{(t)}$ for the tail class only.

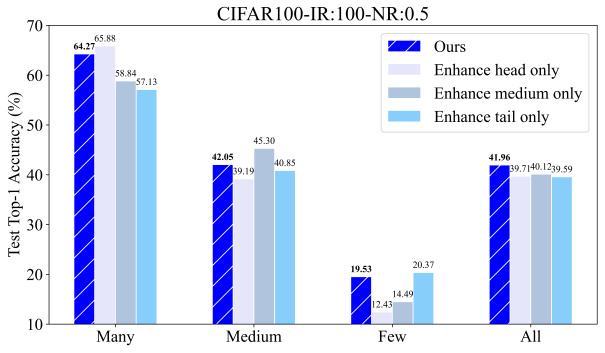

Figure 7. Bar chart of top-1 test accuracies across different partitions and overall for IBC and its degenerated versions on CIFAR-100 with a noise ratio of 0.5 and an imbalance ratio of 100.

As shown in Fig. 7, an interesting phenomenon emerges: using a shot-specific soft label improves classification performance within that shot. This supports our earlier argument in Sec. 1 that semantically related classes can assist each other's learning process, provided they share similar semantic patterns. However, this gain in performance comes at the cost of reduced performance in other partitions. For instance, while the "Enhance tail only" approach achieves a **+0.84%** improvement in the few-shot partition, it results in a **-7.14%** decrease in the many-shot partition and a **-1.20%** decrease in the medium-shot partition. Therefore, our SME component functions as a fusion module, combining the advantages of each shot-specific soft label. **While performance on specific shots may not be optimal, the overall performance remains stable.** These results further demonstrate the robustness of IBC.

4. Conclusion

In this paper, we propose a simple yet effective method, **I**nstances **B**enefitting **C**lasses (IBC), to tackle the challenging problem of *Long-Tailed Noisy Label Learning* (LTNLL). IBC boosts class representations by leveraging semantically related instances. Through soft label construction and a shot-specific multi-expert ensemble learning framework, IBC simultaneously addresses both label noise and class imbalance in a unified manner. Furthermore, the shot-specific multi-expert ensemble framework demonstrates that shot-specific soft labels can controllably and significantly affect the performance of a specific shot. IBC exhibits more separable representations, particularly for tail classes, improving both overall and partition performance. Extensive experiments show that IBC outperforms previous SOTA methods across a range of benchmark and real-world datasets, demonstrating its robustness in extremely noisy and long-tailed environments. We believe our method makes solid contributions and offers valuable insights for future research in this domain.

References

[1] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, 2019. 2, 4

[2] Kaidi Cao, Yining Chen, Junwei Lu, Nikos Arechiga, Adrien Gaidon, and Tengyu Ma. Heteroskedastic and imbalanced deep learning with adaptive regularization. In *International Conference on Learning Representations*, 2021. 2, 5

[3] Aritra Ghosh, Himanshu Kumar, and P Shanti Sastry. Robust loss functions under label noise for deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017. 2

[4] 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 *Advances in Neural Information Processing Systems*, pages 8535–8545, 2018. 7

[5] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9726–9735, 2020. 2, 3, 5

[6] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5375–5384, 2016. 1

[7] Yingsong Huang, Bing Bai, Shengwei Zhao, Kun Bai, and Fei Wang. Uncertainty-aware learning against label noise on imbalanced datasets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6960–6969, 2022. 2

[8] 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 *International Conference on Machine Learning*, pages 2304–2313. PMLR, 2018. 7

[9] Shenwang Jiang, Jianan Li, Ying Wang, Bo Huang, Zhang Zhang, and Tingfa Xu. Delving into sample loss curve to embrace noisy and imbalanced data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7024–7032, 2022. 2

[10] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *International Conference on Learning Representations*, 2020. 5

[11] Hao-Tian Li, Tong Wei, Hao Yang, Kun Hu, Chong Peng, Li-Bo Sun, Xun-Liang Cai, and Min-Ling Zhang. Stochastic feature averaging for learning with long-tailed noisy labels. In *IJCAI*, pages 3902–3910, 2023. 2, 4, 5

[12] Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S Kankanhalli. Learning to learn from noisy labeled data. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5051–5059, 2019. 7

[13] Junnan Li, Richard Socher, and Steven C.H. Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *International Conference on Learning Representations*, 2020. 5, 7

[14] Yifan Li, Hu Han, Shiguang Shan, and Xilin Chen. Disc: Learning from noisy labels via dynamic instance-specific selection and correction. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24070–24079, 2023. 5

[15] Zhuo Li, He Zhao, Zhen Li, Tongliang Liu, Dandan Guo, and Xiang Wan. Extracting clean and balanced subset for noisy long-tailed classification, 2024. 5

[16] Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. In *Advances in Neural Information Processing Systems*, pages 20331–20342, 2020. 7

[17] Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(3):447–461, 2016. 2

[18] Yang Lu, Yiliang Zhang, Bo Han, Yiu-Ming Cheung, and Hanzi Wang. Label-noise learning with intrinsically long-tailed data. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1369–1378, 2023. 2, 5

[19] 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 *International Conference on Machine Learning*, pages 3355–3364. PMLR, 2018. 7

[20] Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *Advances in Neural Information Processing Systems*, 2013. 2

[21] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2233–2241, 2017. 7

[22] Haim Permuter, Joseph Francos, and Ian Jermyn. A study of gaussian mixture models of color and texture features for image classification and segmentation. *Pattern Recognition*, 39(4):695–706, 2006. 4

[23] Jiawei Ren, Cunjun Yu, Xiao Ma, Haiyu Zhao, Shuai Yi, et al. Balanced meta-softmax for long-tailed visual recognition. In *Advances in Neural Information Processing Systems*, pages 4175–4186, 2020. 5

[24] Jiang-Xin Shi, Tong Wei, Yuke Xiang, and Yu-Feng Li. How re-sampling helps for long-tail learning? In *Advances in Neural Information Processing Systems*, pages 75669–75687, 2023. 2

[25] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 34(11):8135–8153, 2023. 2

[26] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016. 2, 4

[27] Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5552–5560, 2018. 7

[28] Yu-Feng Li Tong Wei, Jiang-Xin Shi and Min-Ling Zhang. Prototypical classifier for robust class-imbalanced learning. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2022. 2, 5

[29] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. Deep learning for computer vision: A brief review. *Computational Intelligence and Neuroscience*, 2018(1):7068349, 2018. 1

[30] Xudong Wang, Long Lian, Zhongqi Miao, Ziwei Liu, and Stella Yu. Long-tailed recognition by routing diverse distribution-aware experts. In *International Conference on Learning Representations*, 2021. 5

[31] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 322–330, 2019. 7

[32] Tong Wei, Jiang-Xin Shi, Wei-Wei Tu, and Yu-Feng Li. Robust long-tailed learning under label noise, 2021. 5

[33] Xiaobo Xia, Tongliang Liu, Bo Han, Chen Gong, Nannan Wang, Zongyuan Ge, and Yi Chang. Robust early-learning: Hindering the memorization of noisy labels. In *International conference on learning representations*, 2020. 7

[34] Shiyu Xuan and Shiliang Zhang. Decoupled contrastive learning for long-tailed recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6396–6403, 2024. 2, 5

[35] Kun Yi and Jianxin Wu. Probabilistic end-to-end noise correction for learning with noisy labels. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7017–7025, 2019. 7

[36] Xuanyu Yi, Kaihua Tang, Xian-Sheng Hua, Joo-Hwee Lim, and Hanwang Zhang. Identifying hard noise in long-tailed sample distribution. In *European Conference on Computer Vision*, pages 739–756. Springer, 2022. 2

[37] Manyi Zhang, Xuyang Zhao, Jun Yao, Chun Yuan, and Weiran Huang. When noisy labels meet long tail dilemmas: A representation calibration method. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15844–15854, 2023. 1, 2, 5, 7

[38] Yikai Zhang, Songzhu Zheng, Pengxiang Wu, Mayank Goswami, and Chao Chen. Learning with feature-dependent label noise: A progressive approach. In *International Conference on Learning Representations*, 2021. 7

[39] Yifan Zhang, Bryan Hooi, Lanqing Hong, and Jiashi Feng. Self-supervised aggregation of diverse experts for test-agnostic long-tailed recognition. In *Advances in Neural Information Processing Systems*, pages 34077–34090, 2022. 5

[40] Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. Deep long-tailed learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(9):10795–10816, 2023. 1

[41] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Advances in Neural Information Processing Systems*, 2018. 7

[42] Songzhu Zheng, Pengxiang Wu, Aman Goswami, Mayank Goswami, Dimitris Metaxas, and Chao Chen. Error-bounded correction of noisy labels. In *International Conference on Machine Learning*, pages 11447–11457. PMLR, 2020. 7