Noise-Free Prototype Guided Representation Calibration under Label Noise

Huiting Yuan[a], Tingjin Luo[a,*], Xinghao Wu[b], Jie Jiang[c]

[a]*College of Science, National University of Defense Technology, Changsha, 410073, Hunan, China*
[b]*State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing, 100191, China*
[c]*School of Mathematical Sciences, Shenzhen University, Shenzhen, 518000, Guangdong, China*

Abstract

Real-world datasets inevitably suffer from label noise, which misleads deep networks and disrupts the underlying representation structures, resulting in poor generalization. As category representatives, prototypes are widely adopted in learning with noisy labels due to their strong semantic expressiveness. Existing works typically obtain prototypes by averaging representations within each class and update them during training. However, prototypes achieved by such label-dependent procedures may deviate from their optimal positions under noisy labels, thereby failing to guide the model towards stable and accurate predictions. In this paper, to mitigate noise-induced prototype deviation, and further learn more robust representations, we propose a novel method called Noise-Free Prototype guided Representation Calibration (NFPRC), which introduces fundamentally different, label-independent prototype construction and utilization. Specifically, NFPRC first leverages unsupervised contrastive learning to extract representations and then applies clustering to assign the nearest prototype to each instance. These noise-free prototypes are then fixed to impose directional constraints and guide robust representation learning. Additionally, NFPRC introduces a dynamic weighting strategy that assigns higher importance to instances with larger cross-entropy losses, thereby prioritizing potentially mislabeled instances and enhancing the model's adaptability to more complex noisy label scenarios. Extensive experi-

*Corresponding author

Email addresses: yuanhuiting0724@163.com (Huiting Yuan), tingjinluo@hotmail.com (Tingjin Luo), wuxinghao@buaa.edu.cn (Xinghao Wu), jiangjie2023@email.szu.edu.cn (Jie Jiang)

ments on both synthetic and real-world noisy label benchmarks validate the effectiveness of our method in improving representation learning and combating noisy labels. The code is available at: https://github.com/Huiting-hub/NFPRC.

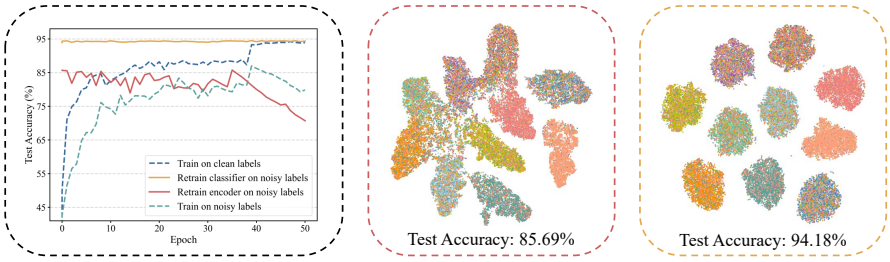
Keywords: Noisy label, Noise-free prototype, Representation calibration, Robust learning

1. Introduction

Deep neural networks have been widely applied across various domains and have achieved remarkable performance [1, 2], largely benefiting from large-scale, high-quality datasets with accurate labels. However, it is both expensive and time-consuming to acquire such perfect datasets in practice, primarily due to errors introduced by manual and automatic annotators [3]. As a result, real-world datasets are inevitably contaminated with label noise. Although deep networks exibit impressive learning capabilities, they struggle to distinguish between clean and noisy labels, which causes the network to ultimately overfit the noisy labels, leading to decline of model performance and generalization ability [4, 5].

Similar to related work [6], by decoupling the training with noisy labels into representation and classifier learning (Fig. 1 (a)), we observe that representation (feature extractor) is indeed more prone to damage from noise than classifier. In essence, the representations of mislabeled instances tend to cluster in the direction of their incorrect labels [7], thus gradually deviating from the ground-truth clustering centers. Therefore, label noise primarily degrades the model's classification ability by disrupting the representation structure (Fig. 1 (b)). Additionally, when the learned representations are closely-clustered, the classifier is still able to establish relatively accurate decision boundaries despite the presence of noisy labels (Fig. 1 (c)). Enlightened by this, our study focuses on discouraging the memorization of representations for mislabeled data and restoring the feature space to a less contaminated structure.

Recently, prototypes have been extensively explored in classification tasks [8, 9] due to their excellent generalization ability and robustness. Prototypes refer to vectors in the representation space that accurately represent the semantics of categories,



(a) Learning curves of different training modes (b) Retrain encoder (c) Retrain classifier

Figure 1: Illustrations of decoupled training and visualizations of representation structure under two decoupling states. Experiments are conducted on synthetic CIFAR-10 with 40% symmetric label noise. (a) Learning curves of different training modes. First, the model is trained on clean labels. Then, the training is decoupled into representation learning and classifier learning. Specifically, we fix the classifier trained on clean labels and retrain the encoder on noisy labels (see the red curve and (b)); we fix the encoder trained on clean labels and retrain the classifier on noisy labels (see the yellow curve and (c)). Given ideal representations, retraining classifier with noisy labels can achieve result comparable to clean training.

i.e., class centers. Existing methods leverage the similarity between sample features and class prototypes to either correct noisy labels [10, 11] or identify clean samples [12, 13]. These methods typically construct prototypes by averaging embeddings of samples within each class, and iteratively update them as the representations evolve throughout the training process. However, the effectiveness of such procedures is *debatable* in the scenario of noisy labels. Specifically, when labels are noisy and representations are misclustered, the computed prototypes can deviate from their true positions, thus failing to guide the model towards accurate predictions. Moreover, the frequent updates to prototypes may prevent the model from focusing on stable representation alignment with the prototypes, thereby inducing instability in representation learning.

In this paper, to address the limitations of previous works and tackle label noise from the perspective of representation learning, we propose a *noise-free prototype guided representation calibration* method, named NFPRC. Unlike prior approaches that rely on label-dependent procedures and suffer from noise-induced instability, NFPRC offers a new perspective by introducing a fundamentally label-independent mechanism for prototype construction and utilization. Specifically, we first pre-train a feature encoder using unsupervised contrastive learning to extract deep representations

for all training samples. Based on these representations, we then perform unsupervised clustering to derive class prototypes and assign the nearest prototype to each instance—entirely free of label supervision. Our motivations hereby are two-fold. First, unsupervised contrastive learning can discover the intrinsic patterns and structures within the data without label information, thereby extracting noise-free and semantically meaningful representations. Second, the process of obtaining prototype for each instance through clustering is also label-independent, thus avoiding the influence of label noise. As a result, the obtained prototypes are naturally robust and remain fixed throughout the training process.

Based upon these noise-free and fixed prototypes, we propose two representation calibration strategies: uniform regularization and loss-based dynamic regularization. In more detail, uniform constraint ensures that the representations learned during subsequent training are close to their corresponding prototypes, pulling the biased representations affected by label noise back towards a more accurate position. For loss-based constraint, we dynamically adjust the strength of representation calibration by assigning higher weights to samples with larger losses, which are more likely to be noisy or hard-but-clean. This flexible, loss-based dynamic weighting strategy enables our method to adapt to more complex high-noise scenarios. Through the above representation calibration procedures, the learned representations are recovered to be relatively unbiased, thereby mitigating the side effects of label noise, following better generalization ability. The contributions of this paper are summarized as follows:

- We reveal that label-derived prototypes in prior works remain vulnerable to label noise. To address this, we introduce a label-independent prototype generation mechanism inspired by unsupervised representation learning, thereby mitigating prototype deviation caused by noisy supervision.
- We propose NFPRC, a novel framework that explicitly restores corrupted representations by enforcing both uniform and dynamic regularization constraints, where noise-free and fixed prototypes serve as robust semantic anchors to guide representation learning.
- We extensively evaluate NFPRC across multiple simulated and real-world noisy

label datasets. The results clearly demonstrate that the proposed method consistently outperforms state-of-the-art baselines, confirming its effectiveness in enhancing representation learning and combating label noise. Detailed ablation studies and discussions are also provided.

The rest of this paper is organized as follows. Section 2 reviews the literature related to this study. Section 3 presents the technical details and the training algorithm of the proposed method. Section 4 reports extensive experimental results. Finally, Section 5 concludes the paper.

2. Related work

In recent years, learning with label noise has attracted increasing attention [14], and considerable efforts have been invested to handle the issue. Beyond standard classification settings, noisy supervision has also been explored in more complex scenarios such as cross-modal retrieval [15] and multi-view learning [16, 17], further highlighting the broad impact of label noise across diverse tasks. In this section, we briefly review related work that is more closely related to our study, including robust loss functions, loss correction, sample selection, and prototype-based approaches.

2.1. Robust loss functions

The traditional cross-entropy loss has been proved to be non-robust in label noise scenarios. To maintain the risk consistency between learning under noisy and clean labels, many new loss functions have been designed to mitigate overfitting to mislabeled data. Generalized cross-entropy Loss (GCE) [18] strikes a balance between the noise robustness of mean absolute error (MAE) and the fast convergence of cross-entropy (CE). LogitClip [19] and OGC [20] mitigates overfitting by clipping logit vectors or gradient to constrain the cross-entropy loss. Asymmetric Unhinged Loss (AUL) [21] is derived from unhinged loss, etc. While robust loss functions typically offer theoretical guarantees under idealized assumptions, their practical effectiveness remains constrained by significantly diminished generalization capacity in complex real-world scenarios. Besides, these works predominantly rely on end-to-end training paradigms, which struggle to directly recover corrupted representation structures.

2.2. Loss correction

Methods with loss correction aim to combat label noise by adjusting the training loss in various ways, such as estimating transition probability matrix [22, 23] to recover the posterior probabilities of samples, reweighting the loss [24], or introducing additional adaptive layers [25], etc. However, the noise transition matrix is difficult to be estimated accurately, especially when the relied anchor points [26] cannot be identified and the number of categories is large. Additionally, these correction methods may further amplify the noise-induced biases and lead to accumulation of errors, since the correction is based on the information from deep networks trained on noisy data.

2.3. Sample selection

Sample selection methods have recently gained popularity. These approaches aim to select samples from the noisy datasets that are more likely to be clean for updating based on specific criteria [27]. Commonly employed criteria include the small-loss trick [28, 29] based on the memorization effect, as well as agreement [30] or disagreement [31] principles. These methods often train two networks simultaneously [32] to reduce error accumulation and assume that the noise rate is known. However, this significantly increases computational costs and may lead to training instability. Moreover, these methods still operate within the label space and rely on noisy supervision, thereby limiting their ability to prevent the propagation of noise into the learned representation.

2.4. Prototypes in learning with noisy labels

Prototype-based methods for learning with noisy labels primarily leverage prototypes for label correction [10, 11] and sample selection [12, 13]. Prototypes can be achieved in various ways, such as warming up with noisy labels to obtain a reasonably reliable feature extractor and then using it to generate initial prototypes [11, 12], selecting high-density samples as prototypes [10], or incorporating learnable prototypes directly into the final loss function [33]. Most approaches typically calculate the feature mean of samples within the same class as the prototypes, leveraging their assigned labels and updating the prototypes as the representations evolve during training. However, accurate estimation of prototypes relies on the correctness of both labels

and representations, which cannot be guaranteed under noisy conditions. On the other hand, representation learning tends to suffer from instability due to noise accumulation caused by continuously updated prototypes, and often lacks clean and effective clustering guidance. Different from prior works, we explore the use of noise-free and fixed prototypes to address label noise from the view of representation calibration.

3. Methodology

3.1. Preliminaries

In the sequel, vectors are denoted by lowercase bold-faced letters, while scalars are in lowercase letters. Let $[z] = \{1, 2, \ldots, z\}$. The ℓ_p norm of a vector is denoted by $\|\cdot\|_p$. We consider a K-class classification problem, where $K \geq 2$. Given a noisy training dataset $\tilde{\mathcal{D}} = \{(\boldsymbol{x}_i, \tilde{y}_i)\}_{i=1}^{n}$, where n is the sample size, $\boldsymbol{x}_i$ denote the i-th instance, $\tilde{y}_i \in [K]$ denotes the actually acquired noisy label, and y_i is the clean version of $\tilde{y}_i$, which is unobservable. Our network is decoupled into two parts, one is the representation learning part, i.e., $\boldsymbol{f}_i = f_\theta(\boldsymbol{x}_i) \in \mathbb{R}^d$, where $\boldsymbol{f}_i$ is the representation for $\boldsymbol{x}_i$ (the output of the second-to-last layer of the network) and $f(\cdot)$ is feature extractor with trainable parameter θ, d is the feature dimension; another part is the classifier, i.e., $p(\boldsymbol{x}_i) = g_w(f_\theta(\boldsymbol{x}_i)) \in \mathbb{R}^k$, where $g(\cdot)$ is the classifier with trainable parameter w, $p(\boldsymbol{x}_i)$ is softmax output of the classifier, and the class with the highest confidence, i.e., $y' = \arg\max_i(p)$, is taken as the prediction for the sample. In this paper, the aim is to learn a robust classifier given the noisily labeled training datasets.

Next, the proposed method NFPRC is elaborated step by step. Overall, NFPRC consists of three stages. Specifically, in the first stage, unsupervised contrastive learning is employed to enhance the representations. In the second stage, the biased representations are calibrated by noise-free and fixed prototypes, thereby improving the robustness of the classifier. In the third stage, the model is trained by jointly optimizing the cross-entropy loss under noisy supervision and the representation calibration loss established through the above analysis. An overview of the proposed method is shown in Fig. 2. The following sections provide more technical details of our approach.

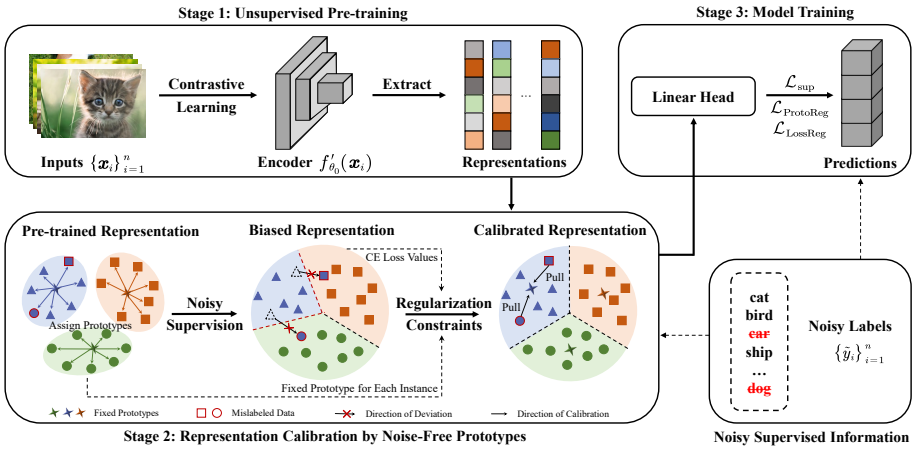


Figure 2: The illustration of the proposed method, which uses noise-free and fixed prototypes achieved by unsupervised contrastive learning to guide representation calibration.

3.2. Unsupervised contrastive learning

Recent works have highlighted the tremendous potential of unsupervised contrastive learning for representation learning across diverse domains, including computer vision [34, 35] and natural language processing [36]. Contrastive learning is a specialized unsupervised learning method aimed at learning discriminative embeddings without human annotated labels. Intuitively, as the contrastive learning is free of labels and can capture the intrinsic structure of the data, the achieved representations will not be influenced by incorrect labels and are naturally robust. Hence, we utilize the deep representations achieved by contrastive learning to enhance the noisy supervised learning.

Specifically, contrastive learning achieves robust representations by maximizing the similarity between visually similar samples while minimizing the similarity between unrelated ones. This enables the model to cluster similar instances closely in the latent embedding space while effectively separating dissimilar ones. In this work, we follow the popular setup of MoCo [35] and pre-train an encoder networks. MoCo learns visual representations by contrasting positive and negative sample pairs in a dynamic memory bank, leveraging a momentum-based update mechanism. For each input $\boldsymbol{x}$, two augmented views are generated: a query $\boldsymbol{x}^q$ and a key $\boldsymbol{x}^{k^+}$, which are then fed into two encoders with identical network structures but different initializations respectively,

mapping the input into a low-dimensional embedding feature z^q and z^{k^+}. These two views form a positive pair, while all other samples serve as negative samples. The contrastive loss for the input $\boldsymbol{x}_i$ can be written as:

$$\mathcal{L}_{\text{con}}(z_i^q, z_i^{k^+}) = -\log \frac{\exp(z_i^q \cdot z_i^{k^+}/\tau)}{\sum_{k\in\{k^+,k^-\}} \exp(z_i^q \cdot z_i{}^k/\tau)}, \tag{1}$$

where the $\{k^+, k^-\}$ is the set of positive and negative keys, and $\tau > 0$ is the temperature parameter. The enhanced contrastive representations can be trained by minimizing the loss in Eq. (1). Finally, the query encoder $f'(\cdot)$ with parameter θ_0 is preserved to extract features for later representation calibration. Furthermore, the preserved feature extractor is used as the initial model for subsequent training, which accelerates the convergence of our method.

3.3. Representation calibration by noise-free prototypes

Label noise impairs feature learning by distorting representation clustering and data structure. To mitigate this, we focus on calibrating the biased representations by leveraging noise-free prototypes from unsupervised contrastive learning.

3.3.1. Assigning prototype for each instance

Firstly, with pre-trained encoder $f'_{\theta_0}(\cdot)$, unsupervised representations can be extracted for all training data by:

$$\boldsymbol{Z} = \{f'_{\theta_0}(\boldsymbol{x}_i)\}_{i=1}^n, \tag{2}$$

where $f'_{\theta_0}(\boldsymbol{x}_i) \in \mathbb{R}^d$ denotes the extracted deep representation of $\boldsymbol{x}_i$. A crucial step that follows is to effectively exploit the meaningful information embedded in these representations. In unsupervised representation spaces, Euclidean distance has been widely observed to align well with semantic similarity. Moreover, it is computationally efficient and stable in high-dimensional feature spaces, making it particularly suitable for our setting. To this end, we follow prior work [37] and perform k-means clustering with Euclidean distance on the representation set $\boldsymbol{Z}$, and record prototype $\boldsymbol{c}_k \in \mathbb{R}^d$ for each class $k \in [K]$:

$$\boldsymbol{c} = \{\boldsymbol{c}_k\}_{k=1}^K = \text{Cluster}(\boldsymbol{Z}), \tag{3}$$

where $\boldsymbol{c}$ represents the set of prototypes for all categories. Based on the prototype set $\boldsymbol{c}$, the distance between each sample's unsupervised representation and the prototypes is computed as follows:

$$d\left(f'_{\theta_0}(\boldsymbol{x}_i),\ \boldsymbol{c}_k\right) = \left\| f'_{\theta_0}\left(\boldsymbol{x}_i\right) - \boldsymbol{c}_k \right\|_2 . \tag{4}$$

For each instance, we assign the nearest prototype from $\boldsymbol{c}$ based on the computed distances and record $\boldsymbol{C} = \{\boldsymbol{C}_i\}_{i=1}^{n}$:

$$\boldsymbol{C}_i = \boldsymbol{c}_{k^*}, \text{where } k^* = \arg\min_{k \in \{1,\ldots,K\}} d\left(f'_{\theta_0}(\boldsymbol{x}_i),\ \boldsymbol{c}_k\right). \tag{5}$$

The prototypes obtained in this way exhibit strong robustness and is beneficial for subsequent representation calibration, as they are derived from unsupervised clustering on representations learned independent of labels, minimizing the interference of label noise to the greatest extent.

3.3.2. Uniform regularization constraint

Under noisy supervision, we initialize the model using the preserved feature extractor and further optimize it while simultaneously training the classifier. For the classification task, a linear head is added and the *cross-entropy* loss ℓ_{ce} is employed. The noisily supervised classification loss is defined as follows:

$$\begin{aligned} \mathcal{L}_{\text{sup}} &= \frac{1}{n}\sum_{i=1}^{n} \ell_{\text{ce}}\left(p\left(\boldsymbol{x}_i\right), \tilde{\boldsymbol{y}}_i\right) \\ &= -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{K} \tilde{\boldsymbol{y}}_i^j \log(p^j(\boldsymbol{x}_i)), \end{aligned} \tag{6}$$

where $\tilde{\boldsymbol{y}}_i$ is the one-hot encoding of $\tilde{y}_i$. $\mathcal{L}_{\text{sup}}$ provides noisy but useful information, which, when combined with prior information from our prototypes, jointly guides the representation learning. The proposed noise-free prototypes serve as robust anchors to guide the direction of subsequent representation learning. They are fixed throughout training, which enhances the stability of representation learning and ensures consistent guidance. Technically, we impose a constraint on the distance between the subsequently learned representations $f_\theta(\boldsymbol{x})$ and their assigned prototypes $\boldsymbol{C}$. The distance

regularization is formulated as:

$$\mathcal{L}_{\text{ProtoReg}} = \frac{1}{n}\sum_{i=1}^{n} \|f_\theta(\boldsymbol{x}_i) - \boldsymbol{C}_i\|_2. \tag{7}$$

This distance constraint effectively pulls the biased features, affected by label noise, back to a more accurate position. In $\mathcal{L}_{\text{ProtoReg}}$, we apply an equal-importance regularization constraint to each instance using a fixed, time-invariant weight, ensuring stable guidance for representation learning in the presence of label noise.

3.3.3. Loss-based dynamic regularization constraint

Although the uniform constraint can effectively calibrate the representations, an individualized strategy is still necessary to cope with some complex, high-noise scenarios. Owing to the memorization effect of deep neural networks (DNNs), DNNs tend to learn simple patterns first, and gradually fit noisy patterns later [4, 5]. As a result, clean samples typically exhibit smaller losses than noisy ones, which is the core idea of the small-loss criterion. Building on this, we observe that the *cross-entropy* loss values generated during training can serve as natural weights. These loss values not only help distinguish clean samples from noise samples to a certain extent, but also decrease as training progresses. Notably, some clean-but-hard samples may exhibit high losses, as they show similar training dynamics to mislabeled data [28]. Such samples may be beneficial for generalization and should also be assigned higher weights to facilitate more effective representation learning. Sparked by this, for each training instance $(\boldsymbol{x}_i, \tilde{\boldsymbol{y}}_i)$, we use its *cross-entropy* loss value $\mathcal{L}_i = \ell_{\text{ce}}\left(p\left(\boldsymbol{x}_i\right), \tilde{\boldsymbol{y}}_i\right)$ from the current epoch to assign weight, introducing the following loss-based dynamic regularization:

$$\mathcal{L}_{\text{LossReg}} = \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}_i \cdot \|f_\theta(\boldsymbol{x}_i) - \boldsymbol{C}_i\|_2. \tag{8}$$

By employing Eq. (8), samples with higher loss values, which are more likely to be noisy or hard, are assigned larger weights. Moreover, as the training goes on and the learned representations improve, the losses gradually decrease, thereby diminishing the role of representation calibration. This allows us to dynamically adjust the strength of representation calibration for each sample. The flexible weighting mechanism enables

the model to focus on potentially noisy or hard samples while relatively relaxing the constraints on clean samples, ultimately contributing to better-learned representations.

Finally, combining the above analyses, the overall objective is formulated as:

$$\mathcal{L} = \mathcal{L}_{\text{sup}} + \lambda_1 \mathcal{L}_{\text{ProtoReg}} + \lambda_2 \mathcal{L}_{\text{LossReg}}, \tag{9}$$

where λ_1 and λ_2 are hyper-parameters which control the strength of representation calibration. λ_1 is consistently set to be greater than 0, ensuring a stable and invariant regularization constraint. In more complex noise scenarios, λ_2 is set to be greater than 0 to provide a dynamic and individualized regularization constraint. The second and third terms of the final loss can be combined into:

$$\mathcal{L}_{\text{Reg}} = \frac{1}{n} \sum_{i=1}^{n} (\lambda_1 + \lambda_2 \mathcal{L}_i) \cdot \|f_\theta(\boldsymbol{x}_i) - \boldsymbol{C}_i\|_2. \tag{10}$$

The full algorithm flow of **N**oise-**F**ree **P**rototype Guided **R**epresentation **C**alibration (NFPRC) is provided in Algorithm 1.

3.4. Impact of clustering accuracy on NFPRC

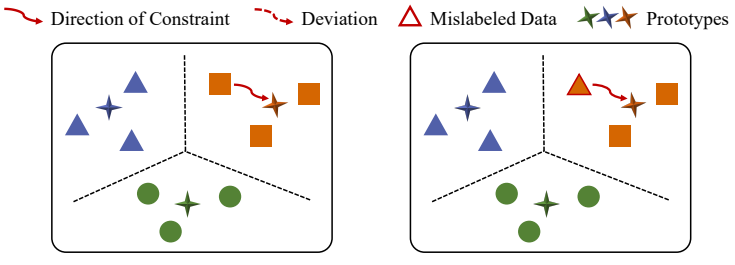
In this part, we discuss how clustering accuracy affects the performance of NFPRC. As analyzed in the Introduction, when the learned representations are accurately and tightly clustered, the classifier can still achieve reliable decision boundaries even under noisy labels. The key insight behind this observation is that if a feature extractor can successfully group training samples of the same class according to their true labels, it is also likely to produce well-clustered representations for unseen test samples. Therefore, the goal of representation calibration is to guide samples to cluster in accordance with their ground-truth labels.

We illustrate the representation learning behavior under varying clustering results in Fig. 3. Let y_{C_i} denote the class label represented by the prototype assigned via clustering. In the figure, the color of each sample point indicates its ground-truth label, while the shape reflects its given noisy label. The relationships among the noisy label, ground-truth label and clustering label can be categorized into four cases. Fig. 3 (a) demonstrates Case 1, where $\tilde{y}_i = y_i = y_{C_i}$. Here, the sample is correctly labeled and assigned to the appropriate prototype, so the regularization constraint encourages the

Algorithm 1 Algorithm of the proposed method NFPRC.

Input: noisy training dataset $\tilde{\mathcal{D}} = \{(\boldsymbol{x}_i, \tilde{y}_i)\}_{i=1}^{n}$, regularization strength λ_1 and λ_2, learning rate η, epoch $T_{\max}$, and iteration $I_{\max}$;
Pre-train unsupervised feature extractor $f'(\cdot)$ with parameter θ_0;
Extract representations $\boldsymbol{Z}$ of all instances by Eq. (2);
Perform k-means clustering on $\boldsymbol{Z}$ and record class prototypes $\boldsymbol{c}$ by Eq. (3);
Assign the nearest prototype for each instance and record $\boldsymbol{C}$ by Eq. (5);
Initialize: feature extractor $f_\theta(\cdot)$ with pre-trained weights θ_0, classifier $g_w(\cdot)$;
for $T = 1, 2, \ldots, T_{\max}$ **do**
  Shuffle training set $\tilde{\mathcal{D}}$;
  for $I = 1, 2, \ldots, I_{\max}$ **do**
    Fetch mini-batch $\tilde{\mathcal{D}}_I$ from $\tilde{\mathcal{D}}$;
    Obtain final loss $\mathcal{L}$ by Eq. (9);
    Update $\theta^{T,I} = \theta^{T,I-1} - \eta \nabla_\theta \mathcal{L}(\tilde{\mathcal{D}}_I)$, $w^{T,I} = w^{T,I-1} - \eta \nabla_w \mathcal{L}(\tilde{\mathcal{D}}_I)$;
  end for
end for
Output: feature extractor $f_\theta(\cdot)$, classifier $g_w(\cdot)$.

representation toward its corresponding prototype, accelerating the training process. Fig. 3 (b) shows Case 2, where $\tilde{y}_i \neq y_i, y_{C_i} = y_i$. In this case, although the sample is mislabeled, it is assigned to the correct prototype. The regularization constraint becomes meaningful, helping to calibrate the representation of mislabeled sample, guiding it toward the toward the correct semantic direction, and enhancing intra-class compactness. Fig. 3 (c) depicts Case 3, where $\tilde{y}_i = y_i,\ y_{C_i} \neq y_i$. Under this scenario, the sample has correct label but is assigned to an incorrect prototype. Although regularization constraint may cause the learned representation to deviate from its true position, $\mathcal{L}_{\text{LossReg}}$ tends to impose a weaker constraint on such clean samples due to their lower loss values, thus mitigating significant deviations. Case 4, shown in Fig. 3 (d), is the worst scenario ($\tilde{y}_i \neq y_i,\ y_{C_i} \neq y_i$), where the sample is both mislabeled and assigned to an incorrect prototype. In this case, applying regularization constraint may further push the learned representation away from its true position. While both Case 3 and Case 4



(a) Case 1: $\tilde{y}_i = y_i = y_{C_i}$ (b) Case 2: $\tilde{y}_i \neq y_i$, $y_{C_i} = y_i$

(c) Case 3: $\tilde{y}_i = y_i$, $y_{C_i} \neq y_i$ (d) Case 4: $\tilde{y}_i \neq y_i$, $y_{C_i} \neq y_i$

Figure 3: Illustrations of representation learning under different clustering results.

are unfavorable, they reflect inevitable errors introduced by the unsupervised clustering process. Nevertheless, as confirmed in our subsequent experiments, these errors are tolerable, since the clustering algorithm is able to correctly group the majority of samples, and the positive effects of our representation calibration method outweigh the negative impact. Furthermore, as scientific research often involves balancing trade-offs, we believe our method remains a reasonable and effective choice.

3.5. Justification of NFPRC

Through the above steps and analysis, NFPRC helps the model learn cleaner and more robust representations that generalize better to unseen data, thereby following improved classification performance. To further illustrate the effectiveness of the proposed method in handling noisy labels, we summarize the following three key aspects:

First, unsupervised contrastive learning aims to actively discover the intrinsic structure of data without relying on label information. The learned representations exhibit strong clustering effects and contain rich information. Therefore, the first key challenge is how to extract and leverage these valuable information. To address this, an effective method is proposed that uses prototypes to embody the useful information

within contrastive representations. Incorporating these robust and noise-free information significantly enhances learning with noisy labels.

Then, after extracting these useful information, the second key challenge is how to effectively utilize it to address the label noise. In this work, the obtained prototypes are regarded as anchor points, i.e., reference points, to provide prior information for representation learning. The obtained noise-free prototypes are fixed to impose constraints on the representation space, guiding the direction of representation learning.

Finally, certain strategies are indispensable. A uniform regularization mechanism is introduced to ensure better alignment between the learned representations and the noise-free prototypes. This strategy helps the model learn more stable representations and implicitly narrows the hypothesis space of the deep network. Furthermore, an adaptive weighting strategy is designed to dynamically adjust the calibration strength based on the scalar value of the classification loss, allowing the model to give more attention to hard samples while reducing focus on simpler ones. Generally, the strategies mentioned above prevent the representations from overfitting mislabeled data and are conducive to better classification boundaries.

4. Experiments

This section first introduces the datasets, implementation details, and comparison methods in Section 4.1. Then, Section 4.2 presents the experimental results on both synthetic and real-world noisy datasets. Section 4.3 provides ablation studies to evaluate the contribution of individual components in the proposed model. Finally, Section 4.4 offers additional analyses on sensitivity and visualization.

4.1. Datasets and implementation details

4.1.1. Simulated noisy datasets

We validate our method on the manually corrupted noisy version of following four datasets: MNIST [38], F-MNIST [39], CIFAR-10 [40], and CIFAR-100 [40]. These four datasets are popularly used to evaluate the algorithm effectiveness in learning with noisy labels [31, 41].

In this paper, four types of synthetic label noise are considered, i.e., (1) Symmetric noise (abbreviated as Sym.): The errors introduced into the labels are random, and each class has the equal chance to be flipped to any of the other classes. (2) Asymmetric noise (abbreviated as Asym.): In this case, the noise is not random but biased, with errors occurring primarily between classes that are more similar in nature. For example, in image classification, an image of a "cat" might be more likely to be mislabeled as "dog" rather than as "car" or "airplane" due to the inherent similarity between cats and dogs. This type of noise can be more challenging to handle. Particularly, we flip $2 \rightarrow 7$, $3 \rightarrow 8$, $5 \rightarrow 6$ for MNIST, Pullover→Coat, Sandals→Sneakers, T-shirt→Shirt for F-MNIST, and Bird→Airplane, Cat→Dog, Deer→Horse, Truck→Automobile for CIFAR-10. Lastly, for CIFAR-100, the 100 classes are divided into 20 super-classes, and each has 5 sub-classes. Each class is flipped to the next within the same super-class. (3) Pairflip noise (abbreviated as Pair.): The errors are restricted to specific pairs of classes, and here each class is flipped to its adjacent class. (4) Instance noise (abbreviated as Ins.): Instance label noise depends on the specific characteristics of the instance, making this type of label errors more aligned with real-world scenarios and more challenging to address. Instance-dependent label noise is generated following previous work [42]. The noise rate is set to 20% and 40%.

For the network structure and optimizer, we apply a LeNet-5 for MNIST, a ResNet-18 for F-MNIST and CIFAR-10, and a ResNet-50 for CIFAR-100. For CIFAR-10 and CIFAR-100, extra data augmentations including random cropping and horizontal flipping are performed. In addition, the networks are trained using SGD optimizer with momentum 0.9 and weight decay 0.001. The batch size is set to 32 for MNIST and F-MNIST, and 64 for CIFAR-10 and CIFAR-100. The initial learning rate is 0.01, which is reduced by a factor of 10 at about 10th epoch for MNIST and F-MNIST (around the 5th epoch for Pair 40% and Ins 40% noise), the 20th epochs for CIFAR-10 (9th epoch for Pair 40% and Ins 40%) , and the 30th epoch CIFAR-100. 50 epochs are set totally.

4.1.2. Real-world noisy datasets

To further evaluate the capability of the proposed method, we also implement experiments on three real-world noisy datasets, i.e., Food-101 [43], Clothing1M [44],

CIFAR-10N and CIFAR-100N [45]. The important statistics of all used datasets are presented in Table 1. During the unsupervised contrastive learning stage, the pre-training is conducted for 300 epochs on MNIST and F-MNIST, 800 epochs on CIFAR-10, CIFAR-100 and Food-101, and 150 epochs on Clothing1M, following standard contrastive learning practice to ensure sufficient convergence [35, 46]. The Food-101 is a large-scale dataset which contains 101 food categories, with a total of 101000 images. For each class, 750 noisy images are designated for training, while the remaining 250 clean images are manually reviewed for testing. Clothing1M comprises 1M noisy images for training and 10k clean-labeled samples for testing. CIFAR-10N and CIFAR-100N provide CIFAR-10 and CIFAR-100 training images with human annotated noisy labels via Amazon Mechanical Turk. Four noisy label versions are used for CIFAR-10 images: three labeled by three independent workers (CIFAR-10N-1/2/3), and for CIFAR-10N-W, the label is randomly selected if any of the three labels are incorrect. For CIFAR-100N, each image is labeled by a single worker.

A ResNet-50 network pre-trained on ImageNet is adopted for Food-101 and Clothing1M. For optimization of Food-101, SGD optimizer with a momentum of 0.9, weight decay of 0.0001, and a batch size of 128 is used. The initial learning rate is set to 0.01 and is decayed by a factor of 10 at the 40th and 80th epoch. The maximum number of epochs is 100. For Clothing1M, we use SGD with momentum 0.9, weight decay 0.005 and a batch size of 32, and an initial earning rate of 0.001, divided by 10 at the 5th epoch over 20 epochs. For CIFAR-10N and CIFAR-100N, the settings are the same as those used for CIFAR-10 and CIFAR100.

Table 1: A brief description of datasets.

Datasets	**# of training**	**# of testing**	**# of classes**	**Size**
MNIST	60,000	10,000	10	28×28×1
F-MNIST	60,000	10,000	10	28×28×1
CIFAR-10	50,000	10,000	10	32×32×3
CIFAR-100	50,000	10,000	100	32×32×3
Food-101	75,750	25,250	101	224×224×3
Clothing1M	1,000,000	10,000	14	224×224×3
CIFAR-10N	50,000	10,000	10	32×32×3
CIFAR-100N	50,000	10,000	100	32×32×3

4.1.3. Comparison methods

For comprehensive evaluations, we compare the proposed method with the following state-of-the-art baselines: (1) Standard, which directly trains the network with softmax *cross-entropy* loss on noisy datasets. (2) CoTeaching [47]: which trains two networks simultaneously and updates parameters with small-loss instances selected by peer networks. (3) CoTeaching+ [31]: which updates by selecting small-loss data with disagreement in predictions. (4) CDR [48]: which categorizes all parameters into critical and non-critical types, and applies distinct update rules to different types of parameters. (5) CNLCU [28]: which uses interval estimation of losses to better select samples. (6) AUL [21]: which is derived from unhinged loss by satisfying the Bayes-optimal condition. (7) Co-Dis [29]: which selects data with high discrepancies in prediction probabilities, allowing more data to be used for training. (8) RTME [49]: which adaptively incorporates large-loss examples by switching between truncated M-estimators and original M-estimators to enhance generalization. (9) ϵ-Softmax [50]: which modifies softmax outputs to approximate one-hot vectors with a controllable error ϵ, thus relaxing the symmetric condition. (10) OGC [20]: which dynamically adjusts the gradient clipping threshold based on the ratio of noise to clean gradients, effectively controlling the influence of noise gradients. Note that we do not directly compare our method with state-of-the-art methods like DivideMix [3], as they are mixtures of multiple techniques and the comparison is unfair.

4.1.4. Measurements

In terms of the performance measurement, we reserve 10% of the noisy training data as the validation set for model selection and record the best test accuracy corresponding to the highest validation accuracy. All experiments are repeated five times, with the mean and standard deviation of the results reported, except for those on Food-101 and Clothing1M, which are conducted only once due to computational costs. All implementations are based on PyTorch. Besides, except for the experiments on Food-101 and Clothing1M which are conducted on NVIDIA A100 GPUs, all other experiments are performed on NVIDIA RTX 4090 GPUs.

4.2. Classification performance on noisy datasets

4.2.1. Results on simulated noisy datasets

Experimental results on four simulated noisy datasets are presented in Table 2, where the highest accuracy is bold faced. The results are analyzed as follows. As shown in the table, our proposed method, NFPRC, consistently outperforms state-of-the-art methods across the all datasets with different types of label noise. In particular, for MNIST, our method surpasses most comparison methods, except in the case of instance-dependent 40% noise, where CoTeaching+ shows slightly better performance, and our method obtains the second-best performance. The advantage of CoTeaching+ on simpler datasets like MNIST can largely be attributed to its emphasis on maintaining clean samples for training rather than focusing on complex feature learning. When dealing with complex datasets such as F-MNIST, CIFAR-10 and CIFAR-100, the limitations of CoTeaching+ become increasingly evident. For these datasets, our method achieves the best performance across a wide range of noise rates. Notably, on CIFAR-100, our method exceeds the best-performing baselines by a large margin. In the asymmetric 20%, pairflip 20%, and instance 20% scenarios, our method demonstrates exceptional performance, with classification accuracies 3.35%, 3.44% and 3.25% higher than the best baseline, respectively. Additionally, our approach is more time-efficient, requiring only half the training time of the Standard baseline. To sum up, the experiments reveal that our method is powerful in handling simulated label noise and exhibits considerable effectiveness and robustness.

4.2.2. Results on real-world noisy datasets

Table 3 presents the experimental results on Food-101, Clothing1M, CIFAR-100N and CIFAR-10N. It can be observed that our method consistently outperforms other competing methods across all datasets, demonstrating its tolerance to real-world label noise. For Food-101 and Clothing1M, our method surpasses all the competing baselines. For CIFAR-100N, our approach shows superior performance, with an improvement of 2.08% over the Standard baseline. Similarly, for CIFAR-10N, our method consistently leads with significant improvements. Notably, on CIFAR-10N-W, a more challenging noisy dataset, our method achieves a marked improvement of 1.76% over

Table 2: Mean and standard deviation of test accuracy (%) on simulated MNIST, F-MNIST, CIFAR-10, CIFAR-100 with varying noise rate. The best results are boldfaced.

Dataset	Method	Sym. 20%	Sym. 40%	Asym. 20%	Asym. 40%	Pair. 20%	Pair. 40%	Ins. 20%	Ins. 40%
MNIST	Standard	98.97 ± 0.09	95.59 ± 0.08	97.16 ± 0.16	96.42 ± 0.23	96.83 ± 0.10	96.27 ± 0.05	96.23 ± 0.15	94.20 ± 0.51
	CoTeaching	98.80 ± 0.04	98.35 ± 0.10	98.52 ± 0.10	98.48 ± 0.15	98.87 ± 0.05	96.00 ± 1.01	98.68 ± 0.09	93.98 ± 0.71
	CoTeaching+	98.80 ± 0.17	98.62 ± 0.05	98.78 ± 0.16	98.11 ± 0.62	98.71 ± 0.12	98.39 ± 0.07	98.75 ± 0.08	**98.32 ± 0.14**
	CDR	99.01 ± 0.10	98.75 ± 0.10	99.16 ± 0.06	98.12 ± 0.38	99.06 ± 0.15	97.58 ± 0.53	98.49 ± 0.10	94.42 ± 0.86
	CNLCU	98.49 ± 0.18	98.07 ± 0.11	99.01 ± 0.08	98.55 ± 0.21	98.68 ± 0.11	94.85 ± 3.22	98.37 ± 0.07	91.50 ± 1.64
	AUL	98.74 ± 0.02	96.61 ± 3.98	70.03 ± 0.04	70.00 ± 0.09	80.30 ± 6.97	58.96 ± 7.08	94.89 ± 4.52	67.65 ± 7.61
	Co-Dis	98.54 ± 0.09	97.96 ± 0.20	99.09 ± 0.09	98.37 ± 0.26	98.50 ± 0.10	93.13 ± 1.99	98.19 ± 0.21	90.47 ± 4.20
	RTME	98.82 ± 0.11	98.42 ± 0.04	98.82 ± 0.09	98.10 ± 0.13	98.66 ± 0.11	97.33 ± 0.51	98.51 ± 0.09	96.20 ± 0.48
	ϵ-Softmax	98.98 ± 0.09	98.66 ± 0.10	99.15 ± 0.12	98.52 ± 0.30	99.05 ± 0.06	98.33 ± 0.15	98.13 ± 0.05	95.92 ± 3.88
	OGC	98.84 ± 0.09	98.53 ± 0.11	98.95 ± 0.06	88.90 ± 9.55	98.81 ± 0.08	98.31 ± 0.15	98.74 ± 0.08	94.51 ± 8.63
	NFPRC	**99.09 ± 0.06**	**98.81 ± 0.07**	**99.23 ± 0.02**	**98.76 ± 0.10**	**99.09 ± 0.05**	**98.48 ± 0.16**	**98.76 ± 0.05**	96.61 ± 0.35
F-MNIST	Standard	91.39 ± 0.26	89.78 ± 0.27	92.57 ± 0.19	89.50 ± 0.88	92.87 ± 0.34	85.40 ± 2.01	90.90 ± 0.46	83.69 ± 2.31
	CoTeaching	91.61 ± 0.48	89.98 ± 0.15	92.45 ± 0.19	90.64 ± 0.70	92.33 ± 0.21	88.95 ± 1.06	91.73 ± 0.40	86.96 ± 2.04
	CoTeaching+	90.57 ± 0.29	89.78 ± 0.24	83.29 ± 4.06	65.32 ± 0.15	90.58 ± 0.33	85.90 ± 5.48	90.32 ± 0.37	71.35 ± 10.68
	CDR	91.40 ± 0.28	89.60 ± 0.49	92.35 ± 0.19	88.72 ± 0.97	92.65 ± 0.27	84.63 ± 4.21	91.12 ± 0.23	83.17 ± 2.82
	CNLCU	91.23 ± 0.26	89.76 ± 0.13	92.72 ± 0.11	88.60 ± 0.59	91.44 ± 0.37	83.37 ± 4.09	90.95 ± 0.28	79.01 ± 6.66
	AUL	89.83 ± 3.30	87.59 ± 3.24	75.49 ± 1.04	67.03 ± 0.19	86.70 ± 3.60	58.34 ± 5.82	76.36 ± 13.75	49.33 ± 5.64
	Co-Dis	90.36 ± 0.35	88.63 ± 0.54	92.46 ± 0.21	89.38 ± 0.70	91.11 ± 0.33	84.87 ± 2.85	90.62 ± 0.38	79.32 ± 4.45
	RTME	91.92 ± 0.18	90.92 ± 0.25	92.31 ± 0.25	87.80 ± 0.23	92.22 ± 0.37	92.40 ± 0.25	91.51 ± 0.57	89.97 ± 0.44
	ϵ-Softmax	92.65 ± 0.17	91.16 ± 0.14	92.23 ± 0.22	86.96 ± 0.85	92.32 ± 0.33	89.19 ± 1.56	92.18 ± 0.28	60.11 ± 19.33
	OGC	92.52 ± 0.20	91.36 ± 0.32	92.41 ± 0.14	82.43 ± 4.86	92.42 ± 0.29	90.71 ± 0.66	91.92 ± 0.31	87.90 ± 5.59
	NFPRC	**93.18 ± 0.25**	**91.75 ± 0.11**	**93.63 ± 0.17**	**91.67 ± 0.13**	**93.97 ± 0.22**	**93.11 ± 0.19**	**93.70 ± 0.17**	**90.56 ± 0.51**
CIFAR-10	Standard	89.73 ± 0.17	86.62 ± 0.32	91.56 ± 0.26	88.12 ± 0.73	91.68 ± 0.18	87.14 ± 0.60	90.62 ± 0.33	84.32 ± 0.94
	CoTeaching	88.53 ± 0.35	85.40 ± 0.52	91.59 ± 0.11	79.14 ± 0.30	88.62 ± 0.44	84.30 ± 0.38	88.64 ± 0.32	83.97 ± 0.92
	CoTeaching+	90.62 ± 0.31	86.94 ± 0.56	90.35 ± 0.12	69.40 ± 1.80	89.59 ± 0.25	83.37 ± 0.42	89.76 ± 0.29	82.08 ± 2.38
	CDR	89.73 ± 0.22	86.50 ± 0.55	91.66 ± 0.42	87.63 ± 1.34	91.68 ± 0.34	87.23 ± 0.31	90.68 ± 0.26	83.63 ± 0.89
	CNLCU	90.11 ± 0.17	86.36 ± 0.67	90.89 ± 0.10	77.73 ± 0.76	89.42 ± 0.45	83.50 ± 0.57	89.47 ± 0.20	83.66 ± 0.55
	AUL	90.38 ± 0.15	86.52 ± 0.74	88.17 ± 0.11	56.33 ± 0.07	89.37 ± 0.38	62.06 ± 5.49	89.64 ± 0.40	74.91 ± 1.74
	Co-Dis	88.36 ± 0.94	84.90 ± 1.17	88.18 ± 0.45	76.21 ± 1.40	87.41 ± 0.37	82.62 ± 0.95	87.12 ± 0.59	82.73 ± 0.46
	RTME	89.62 ± 0.31	86.36 ± 0.37	90.23 ± 0.41	86.02 ± 0.47	89.78 ± 0.23	87.60 ± 0.31	89.37 ± 0.56	86.14 ± 0.20
	ϵ-Softmax	88.70 ± 0.16	86.13 ± 0.29	89.14 ± 0.17	77.43 ± 4.60	88.47 ± 0.41	80.25 ± 7.64	87.35 ± 3.89	50.54 ± 18.73
	OGC	89.74 ± 0.33	85.89 ± 0.54	88.81 ± 0.35	86.58 ± 0.81	87.95 ± 0.37	86.43 ± 0.53	88.20 ± 0.31	86.52 ± 0.50
	NFPRC	**91.20 ± 0.20**	**88.01 ± 0.46**	**92.10 ± 0.09**	**89.00 ± 0.36**	**92.07 ± 0.10**	**87.92 ± 0.15**	**91.28 ± 0.24**	**86.73 ± 0.24**
CIFAR-100	Standard	69.21 ± 0.66	63.25 ± 0.36	71.24 ± 0.50	54.40 ± 0.45	71.25 ± 0.45	54.31 ± 0.93	69.99 ± 0.54	61.18 ± 0.37
	CoTeaching	64.95 ± 0.45	57.82 ± 0.60	62.60 ± 0.89	42.44 ± 0.57	62.58 ± 0.42	43.81 ± 2.04	63.11 ± 0.72	52.32 ± 1.04
	CoTeaching+	59.87 ± 1.95	48.55 ± 3.30	60.02 ± 1.48	42.47 ± 1.92	60.23 ± 1.65	48.60 ± 1.13	60.16 ± 1.90	49.76 ± 1.10
	CDR	69.27 ± 0.33	63.64 ± 0.33	71.43 ± 0.67	55.76 ± 1.01	71.36 ± 0.40	55.42 ± 1.52	70.18 ± 0.61	61.42 ± 1.45
	CNLCU	65.33 ± 0.49	56.59 ± 1.65	60.80 ± 0.55	40.89 ± 0.61	60.14 ± 0.90	40.27 ± 0.77	63.02 ± 0.74	45.94 ± 0.96
	AUL	64.28 ± 0.47	56.48 ± 0.54	61.49 ± 0.51	41.04 ± 1.09	61.12 ± 0.67	40.05 ± 2.70	61.78 ± 0.92	50.80 ± 1.34
	Co-Dis	58.90 ± 2.03	52.85 ± 2.19	58.15 ± 1.01	40.07 ± 0.49	57.79 ± 2.58	40.34 ± 0.96	57.98 ± 0.90	49.13 ± 1.54
	RTME	69.74 ± 0.17	66.58 ± 0.57	69.96 ± 0.38	63.09 ± 0.73	70.16 ± 0.37	63.05 ± 0.55	70.09 ± 0.53	65.58 ± 0.53
	ϵ-Softmax	69.86 ± 0.60	60.84 ± 0.71	61.57 ± 1.33	46.60 ± 1.67	65.91 ± 0.21	53.26 ± 1.94	70.41 ± 0.35	57.89 ± 1.17
	OGC	66.98 ± 0.60	63.45 ± 0.97	66.63 ± 0.33	52.42 ± 1.25	66.70 ± 0.55	54.28 ± 0.52	66.03 ± 0.73	60.63 ± 1.72
	NFPRC	**72.12 ± 0.25**	**67.20 ± 0.32**	**74.78 ± 0.26**	**63.56 ± 0.62**	**74.80 ± 0.15**	**63.37 ± 0.22**	**73.66 ± 0.23**	**67.11 ± 0.45**

the best baseline method CNLCU. These improvements are likely attributed to the fact that our method learns a better representation structure, which enables the classifier to establish more accurate decision boundaries. The results fully show the effectiveness

and clear advantage of our proposed method in handling real-world label noise.

Table 3: Test accuracy (%) on real-world noisy datasets. The best results are boldfaced.

Method	Food-101	Clothing1M	CIFAR-100N	CIFAR-10N-1	CIFAR-10N-2	CIFAR-10N-3	CIFAR-10N-W
Standard	86.91	67.79	60.94 ± 0.54	88.57 ± 0.43	88.65 ± 0.37	88.53 ± 0.60	81.41 ± 0.32
CoTeaching	86.34	66.48	58.04 ± 0.48	89.61 ± 0.32	89.08 ± 0.19	89.45 ± 0.29	82.79 ± 0.30
CoTeaching+	84.53	69.36	55.12 ± 0.61	88.83 ± 0.39	89.11 ± 0.19	88.88 ± 0.20	81.89 ± 0.18
CDR	86.87	69.14	60.67 ± 0.35	88.83 ± 0.28	88.28 ± 0.50	88.32 ± 0.19	80.89 ± 0.60
CNLCU	84.02	69.78	55.51 ± 0.74	89.49 ± 0.10	89.29 ± 0.22	89.52 ± 0.16	82.86 ± 0.25
AUL	58.34	67.18	43.62 ± 0.82	88.80 ± 0.07	88.28 ± 0.27	88.52 ± 0.21	77.44 ± 3.68
Co-Dis	77.39	70.01	52.17 ± 0.52	87.43 ± 0.48	87.27 ± 0.28	87.13 ± 0.35	82.55 ± 0.56
RTME	87.49	68.56	60.53 ± 0.43	88.95 ± 0.33	88.69 ± 0.32	88.61 ± 0.28	82.05 ± 0.45
ϵ-Softmax	85.89	68.74	58.44 ± 0.17	88.51 ± 0.37	88.24 ± 0.36	88.77 ± 0.32	78.92 ± 3.15
OGC	81.21	60.59	58.46 ± 1.12	87.77 ± 0.41	87.99 ± 0.33	87.83 ± 0.29	82.60 ± 0.17
NFPRC	**87.62**	**70.38**	**63.02 ± 0.27**	**90.37 ± 0.16**	**90.31 ± 0.22**	**90.16 ± 0.29**	**84.62 ± 0.54**

4.3. Ablation studies

4.3.1. Impact of each component

We analyze the effect of each component by splitting our method into three modules: unsupervised contrastive learning (abbreviated as UCL), uniform constraint (abbreviated as UC), and loss-based constraint (abbreviated as LC). Experiments are conducted on F-MNIST and CIFAR-10 under two challenging high-noise scenarios: asymmetric 40% and instance 40%. Note that our dataset choices are made flexibly, based on the characteristics of each task or by following common practices in prior studies. Results are summarized in Table 4, from which the following insights are drawn.

First, without pre-training or any regularization constraints, what we obtain is the baseline results of Standard. Then, incorporating unsupervised pre-training as initialization leads to a slight improvement in performance over the baseline, though it remains ineffective in handling noisy labels. Next, adding the uniform constraint (UC) results in a noticeable gain across all settings, highlighting the significant role of noise-free prototype based representation calibration. To further evaluate the individual contribution of the loss-based constraint (LC), we apply LC alone on top of the pre-trained model and observe a clear improvement over the baseline. However, its effectiveness is still limited compared to the full model. Finally, combining both UC and LC based on

the pre-trained model further enhances the model's robustness to noisy labels, demonstrating their positive effect and supplementary contributions to performance. Overall, these findings clearly underline the complementary benefits of the three modules. Combining these modules in our method achieves a consistent improvement in performance compared to the baseline, validating the necessity of each component.

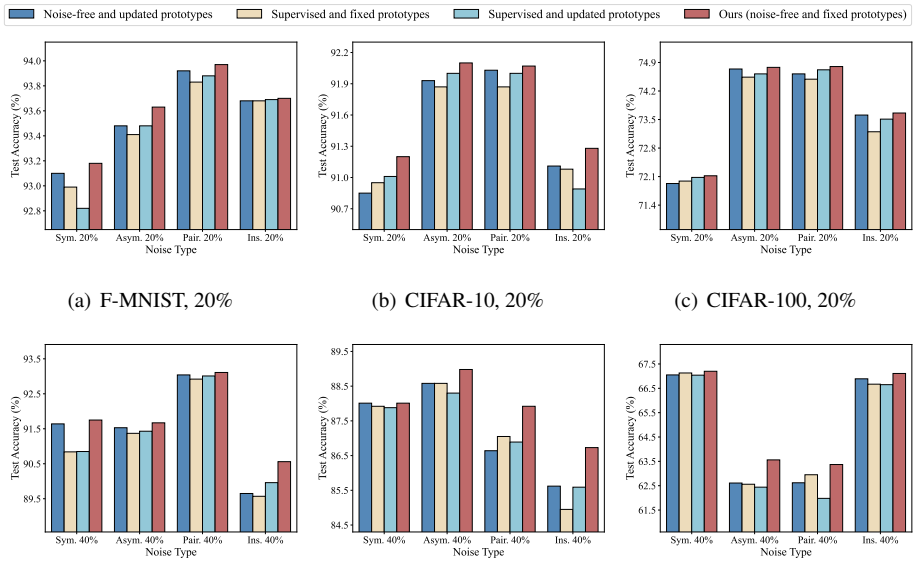
Table 4: Ablation study results on test accuracy (%) for simulated F-MNIST and CIFAR-10 with unsupervised contrastive learning (UCL), uniform constraint (UC), and loss-based constraint (LC). Bold results correspond to our full method NFPRC with all components enabled.

Component			F-MNIST		CIFAR-10	
UCL	UC	LC	Asym. 40%	Ins. 40%	Asym. 40%	Ins. 40%
✗	✗	✗	89.50 ± 0.88	83.69 ± 2.31	88.12 ± 0.73	84.32 ± 0.94
✓	✗	✗	90.77 ± 0.38	89.13 ± 1.02	88.44 ± 0.88	84.69 ± 0.56
✓	✓	✗	91.25 ± 0.23	89.81 ± 1.05	88.57 ± 0.33	85.16 ± 0.61
✓	✗	✓	91.33 ± 0.22	90.09 ± 0.72	88.91 ± 0.40	85.19 ± 1.06
✓	✓	✓	**91.52 ± 0.49**	**90.56 ± 0.51**	**88.98 ± 0.12**	**86.73 ± 0.24**

4.3.2. A closer look on noise-free and fixed prototypes

To further demonstrate the advantages of noise-free and fixed prototypes in our method, as well as their individual impacts, we design three variants of our method for comparison: noise-free updated prototypes, supervised fixed prototypes, and supervised updated prototypes. More precisely, for noise-free updated prototypes, we use the prototypes in our NPFRC as initial values and update them by clustering on the current representations every 5 epochs. For supervised fixed prototypes, we replace the prototypes in our method with those derived from noisy supervision, which means first training a model with the Standard method on noisy labels to obtain representations, and then compute the average representation for each class based on these noisy labels. Lastly, for supervised updated prototypes, it is a combination of the two variants mentioned above. As shown in Fig. 4, our method consistently outperforms the other three variants in nearly all scenarios. Specifically, for noise-free updated prototypes, although it shows competitive results, frequent updates of prototypes cause instability

and substantial computational costs, thereby increasing the training time. In the case of supervised fixed prototypes, the performance is generally inferior to other variants, which supports our claim that supervised prototypes are inevitably susceptible to the adverse effects of noisy labels. Similarly, both of the above issues exist for supervised updated prototypes. In contrast, the noise-free and fixed prototypes proposed in our method effectively mitigate the adverse effects of label noise and reduce the instability introduced by representation updates.



(a) F-MNIST, 20% (b) CIFAR-10, 20% (c) CIFAR-100, 20%

(d) F-MNIST, 40% (e) CIFAR-10, 40% (f) CIFAR-100, 40%

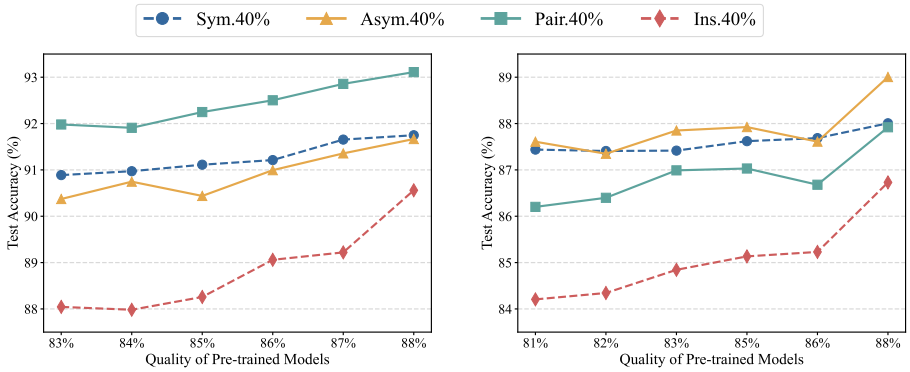
Figure 4: Illustrations of the test accuracy across different variants of our method.

4.4. Further analyses and insights

4.4.1. Stability across different pre-trained models

To investigate the relationship between the capability of our method and the quality of unsupervised pre-training, we evaluate several pre-trained models with varying levels of quality. The quality of the these models is controlled by the total number of pre-training epochs. We use k-Nearest Neighbors (kNN) to assess the quality of the pre-trained models based on the representations extracted after pre-training. The

experiments are conducted on F-MNIST and CIFAR-10 with 40% noise rate. The experimental results are illustrated in Fig. 5. As shown, the performance of our method consistently improves with the enhancement of the pre-trained models. Moreover, our method does not rely heavily on the quality of pre-trained models, demonstrating remarkable stability across different model settings. However, in the case of instance-dependent noise, our method shows some sensitivity but still outperforms the Standard baseline, even with the worst model. Notably, our approach requires only a relatively good pre-trained model to achieve results comparable to the best baseline in Table. 2. These observations confirm that NFPRC maintains superior performance and stability across pre-trained models with different qualities, underscoring its adaptability and robustness in noisy label scenarios.



(a) F-MNIST, 40%

(b) CIFAR-10, 40%

Figure 5: Illustrations of relationship between the test accuracy and the quality of pre-trained models. The experiments are conducted on (a) F-MNIST and (b) CIFAR-10 with 40% noise rate.

4.4.2. Robustness across different clustering algorithms

To further evaluate the flexibility and robustness of NFPRC to different prototype construction strategies, we replace the default k-means clustering with several alternative algorithms, including Gaussian Mixture Models (GMM), Hierarchical Clustering, and Spectral Clustering. As shown in Table. 5, NFPRC not only achieves strong performance with k-means clustering, but also performs competitively when applying

other clustering methods such as GMM and Hierarchical Clustering. These results suggest that our framework is robust and adaptable to various prototype construction strategies. In addition, we provide a justification for choosing k-means as the default clustering method over other clustering algorithms, based on its efficiency, scalability, and stability in high-dimensional feature spaces. Specifically, k-means is computationally efficient and has been widely adopted in unsupervised representation learning [37], demonstrating its practical effectiveness in constructing semantically meaningful prototypes. In contrast, GMM involves more complex parameter estimation and is sensitive to initialization, while Spectral Clustering and Hierarchical Clustering are computationally expensive and scale poorly to large datasets such as Food101 and Clothing1M. Overall, k-means offers a favorable trade-off between computational cost and performance stability, making it a reasonable and effective default choice in the NFPRC framework.

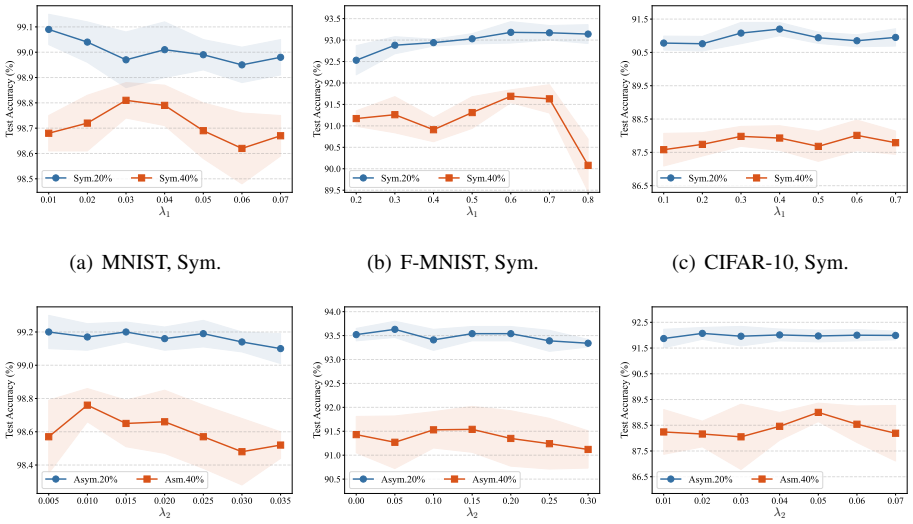
Table 5: Mean and standard deviation of test accuracy (%) on F-MNIST and CIFAR-10 with 40% noise rate under different clustering algorithms. The best results are boldfaced.

Dataset	F-MNIST				CIFAR-10			
Method	Sym. 40%	Asym. 40%	Pair. 40%	Ins. 40%	Sym. 40%	Asym. 40%	Pair. 40%	Ins. 40%
Standard	89.78 ± 0.27	89.50 ± 0.88	85.40 ± 2.01	83.69 ± 2.31	86.62 ± 0.32	88.12 ± 0.73	87.14 ± 0.60	84.32 ± 0.94
Gaussian Mixture Models	**91.78 ± 0.36**	91.54 ± 0.27	93.20 ± 0.55	90.14 ± 1.19	**88.10 ± 0.29**	88.75 ± 1.01	87.34 ± 0.64	**87.66 ± 0.70**
Hierarchical Clustering	91.54 ± 0.23	91.48 ± 0.24	**93.38 ± 0.39**	89.86 ± 1.37	87.94 ± 0.11	88.46 ± 0.63	87.58 ± 0.79	86.98 ± 0.87
Spectral Clustering	91.64 ± 0.18	91.23 ± 0.49	93.32 ± 0.24	90.48 ± 1.11	87.94 ± 0.49	88.95 ± 0.35	87.79 ± 0.88	87.27 ± 1.09
k-means Clustering (ours)	91.75 ± 0.11	**91.67 ± 0.13**	93.11 ± 0.19	**90.56 ± 0.51**	88.01 ± 0.46	**89.00 ± 0.36**	**87.92 ± 0.15**	86.73 ± 0.24

4.4.3. Sensitivity analysis of hyper-parameters

We present the sensitivity analyses of hyper-parameters to evaluate the robustness of the proposed method. Notably, most hyper-parameters related to contrastive learning are kept fixed, with only a few (e.g., batch size and learning rate) requiring minor tuning. In addition, the k-means clustering step does not involve any tunable hyper-parameters. Our method involves only two hyper-parameters controlling the strength of representation calibration, i.e., λ_1 and λ_2. Their optimal values can be efficiently determined via grid search. Intuitively, calibration strength should neither be too high nor too low, but rather at an appropriate level. We empirically validate this under symmet-

ric and asymmetric noise of two ratios by fixing one parameter and adjusting the other to gain insight into how the model responds to each parameter. As shown in Fig. 6, our method demonstrates high stability under a low noise rate of 20%. When the noise rate increases to 40%, the test accuracy follows a trend of first increasing and then decreasing as the parameter increases. This confirms our hypothesis and suggests that our method performs more effectively in high-noise conditions. Moreover, these findings indicate that the values of hyper-parameters can be easily determined by analyzing the trend of test accuracy as the parameters change.



(a) MNIST, Sym. (b) F-MNIST, Sym. (c) CIFAR-10, Sym.

(d) MNIST, Asym. (e) F-MNIST, Asym. (f) CIFAR-10, Asym.

Figure 6: Illustrations of test accuracy with different values of hyper-parameters.

4.4.4. Experiments with higher noise levels

To evaluate the robustness of our method under high noise levels, we conduct experiments on CIFAR-10 and CIFAR-100 datasets with symmetric noise rates of 60%, 70%, and 80%. These two datasets are selected because they are more challenging and representative compared to simpler datasets like MNIST and F-MNIST, making them better suited for evaluating the effectiveness of our method in high-noise settings. In symmetric noise case, the diagonal dominance property can be satisfied. As shown in

Table 6, for CIFAR-10, our proposed NFPRC outperforms all state-of-the-art methods across all noise levels. Similarly, our method maintains a notable performance advantage on more difficult dataset, i.e., CIFAR-100. It is noteworthy that under higher noise rates, the contribution of the information provided by our noise-free and fixed prototypes becomes increasingly significant. These results highlight the remarkable robustness and resilience of our method under challenging high-noise levels.

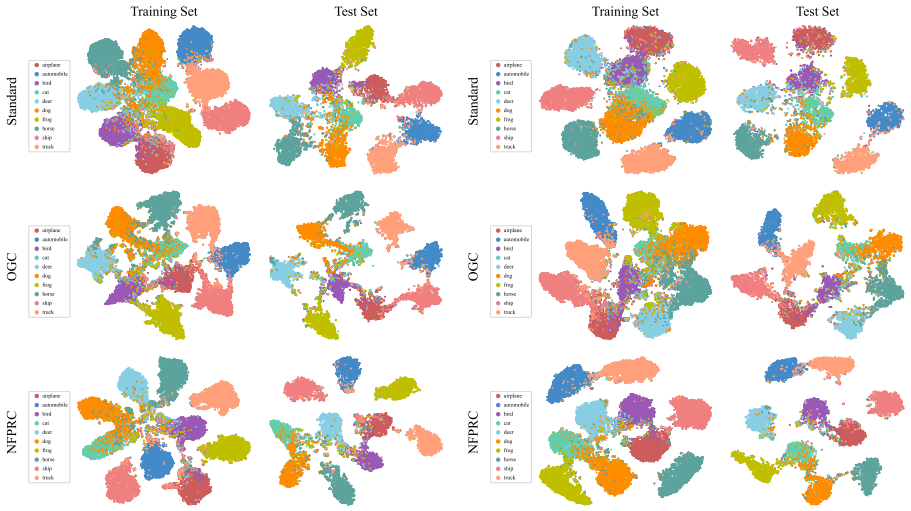
Table 6: Mean and standard deviation of test accuracy (%) on simulated datasets with high noise rate (60%, 70%, 80%). The best results are boldfaced.

Dataset	CIFAR-10			CIFAR-100		
Method	Sym. 60%	Sym. 70%	Sym. 80%	Sym. 60%	Sym. 70%	Sym. 80%
Standard	79.48 ± 0.41	72.84 ± 0.67	48.82 ± 1.73	51.52 ± 0.40	40.39 ± 1.29	21.22 ± 0.62
CoTeaching	77.22 ± 1.13	62.84 ± 6.11	23.64 ± 2.42	45.54 ± 1.57	32.98 ± 1.36	14.78 ± 1.10
CoTeaching+	46.01 ± 5.49	30.26 ± 1.15	19.57 ± 1.95	11.81 ± 3.39	6.65 ± 0.92	4.51 ± 0.63
CDR	79.00 ± 0.76	70.54 ± 1.67	44.50 ± 1.74	52.85 ± 0.45	40.30 ± 0.57	20.01 ± 0.74
CNLCU	75.43 ± 1.33	59.84 ± 2.28	28.70 ± 1.92	42.50 ± 1.64	27.98 ± 0.89	12.56 ± 0.97
AUL	79.45 ± 0.62	66.53 ± 2.12	38.23 ± 6.14	44.47 ± 0.43	34.81 ± 0.75	22.24 ± 0.46
Co-Dis	74.32 ± 1.27	58.85 ± 2.01	30.04 ± 2.85	40.54 ± 3.08	26.62 ± 1.39	11.83 ± 1.55
RTME	80.27 ± 0.45	69.24 ± 0.66	20.33 ± 2.97	52.69 ± 0.55	39.43 ± 1.04	22.72 ± 0.88
ϵ-Softmax	71.99 ± 3.97	55.14 ± 5.09	35.22 ± 3.22	41.37 ± 1.62	32.07 ± 1.3	20.35 ± 0.83
OGC	79.68 ± 0.48	68.76 ± 2.14	38.94 ± 3.08	43.15 ± 2.74	26.82 ± 1.3	20.70 ± 0.74
NFPRC	**82.94 ± 0.70**	**77.01 ± 0.68**	**59.78 ± 0.87**	**59.86 ± 0.32**	**54.54 ± 0.35**	**46.16 ± 0.62**

4.4.5. t-SNE visualization on representations

We utilize t-SNE [51] to visualize the learned representations. In our setup, representations are high-dimensional vectors extracted from the penultimate layer of a deep neural network, which capture the learned characteristics of the input data. Given that CIFAR-10 is a widely used benchmark dataset with a moderate number of classes and data volume, it is particularly suitable for visualization experiments. Therefore, we perform the visualization on the CIFAR-10 dataset under two types of noise, using both training and test sets. As illustrated in Fig. 7, compared to the Standard and OGC [20] baselines, the proposed method NFPRC produces more distinct and well-separated clusters in the t-SNE visualization, enabling clearer differentiation between classes. These results suggest that our method can effectively calibrate representations

to generate more apparent class boundaries, which is crucial to enhance classification performance with imperfect data.



(a) Sym. 40% (b) Ins. 40%

Figure 7: t-SNE visualization of representations for simulated CIFAR-10.

5. Discussion

To better position the proposed NFPRC within the landscape of learning with noisy labels, we discuss its connections to three major lines of related work: self-supervised learning-based methods, prototype-based methods, and calibration methods. This section is organized progressively, highlighting both similarities and, more importantly, the key distinctions that differentiate our approach from each category.

5.1. Relations to self-supervised learning-based methods

Self-supervised learning has been widely adopted to enhance robustness against noisy labels [52, 53, 54], leveraging the strengths of unsupervised representation learning. These methods typically combine supervised and self-supervised training by either jointly optimizing two encoders with structural consistency constraints or applying

self-supervised pretraining during a warm-up stage. However, such designs often introduce additional computational overhead and may suffer from incompatibility between supervised and unsupervised representation structures, requiring extra loss terms for alignment. In contrast, NFPRC adopts a one-time pre-trained unsupervised encoder that remains fixed during the entire training process, significantly reducing computational cost and mitigating the propagation of label noise. More importantly, NFPRC distinguishes itself by extracting noise-free prototypes from unsupervised representations, which serve as reliable structural priors to guide representation calibration.

5.2. Relations to prototype-based methods

Existing prototype-based approaches for learning with noisy labels share the same goal as our proposed NFPRC: leveraging prototypes to guide the training of both feature encoder and classifier by encouraging alignment between sample representations and their corresponding prototypes. However, our work bears two critical differences in terms of prototype construction and utilization. For prototype construction, prior works typically average representations within each class and updating prototypes dynamically [11], or select high-density samples as representatives [10]. In contrast, NFPRC constructs prototypes without any label supervision, making them inherently robust and fixed throughout training. For prototype utilization, existing methods primarily leverage prototypes for label correction [10], sample selection [12, 13], or directly incorporate learnable prototypes into the loss function [33]. In comparison, NFPRC introduces two complementary calibration strategies based on noise-free prototypes: enforcing prototype consistency and applying loss-based dynamic weighting. In summary, our method offers a fundamentally different, label-independent approach to prototype construction and utilization.

5.3. Relations to calibration methods

Calibration strategies have been widely explored to enhance the robustness of deep networks trained on noisy labels. Existing methods can be broadly categorized into two groups: loss correction and label correction. Loss correction approaches adjust the training loss by estimating transition matrix [22, 23], reweighting the loss [24], or

introducing additional adaptive layers [25]. Label correction methods aim to revise noisy annotations using historical model predictions or data topology [55, 56]. However, since these corrections rely on information from models trained with noisy data, their effectiveness cannot be guaranteed in the presence of label noise and may even lead to error accumulation. Different from these mainstream approaches, we adopt a fundamentally different route by explicitly calibrating feature representations under label noise, leveraging prior knowledge acquired from unsupervised learning. A recent study, RCAL [46] also utilizes unsupervised representations for calibration. However, our method differs in two key aspects. In terms of problem setting, RCAL addresses noisy labels in long-tailed data, while our focus is on enhancing robustness against label noise in general settings. In terms of methodology, RCAL performs individual representation calibration by constraining the distance between each sample's current representation and its corresponding unsupervised counterpart, using a regularization term: $\mathcal{L}_{\text{Reg}} = \|f_\theta(\boldsymbol{x}_i) - f'_{\theta_0}(\boldsymbol{x}_i)\}\|_2$. In contrast, NFPRC introduces semantically meaningful prototypes derived from unsupervised representations and use them to impose directional constraints on representation learning.

Table 7: Comparison between NFPRC and RCAL on the CIFAR-10 dataset with various noise settings. The best results are boldfaced.

Method	Sym. 40%	Asym. 40%	Pair. 40%	Ins. 40%
Standard	86.62 ± 0.32	88.12 ± 0.73	87.14 ± 0.60	84.32 ± 0.94
RCAL	86.98 ± 0.32	88.59 ± 0.07	86.52 ± 1.50	85.23 ± 0.89
NFPRC	**88.01 ± 0.46**	**89.00 ± 0.36**	**87.92 ± 0.15**	**86.73 ± 0.24**

In Table 7, we compare the performance of NFPRC and RCAL in the context of learning with noisy labels, where RCAL replaces our prototype-based calibration with individual calibration. The results show that NFPRC consistently outperforms RCAL across all noise settings. Intuitively, individual calibration can improve the performance to some extent because it also incorporates robust prior from unsupervised learning. However, it enforces more strict constraints on all training examples and brings about more errors, which may make RCAL suboptimal in this task. In con-

trast, NFPRC introduces prototype-level calibration, which provides more stable and semantically meaningful guidance, thus leading to superior performance.

6. Conclusions

In this paper, we focus on tackling noisy labels from the perspective of representation learning. We discuss that label-dependent prototypes in previous works are inherently vulnerable to label noise, thereby failing to provide reliable guidance for representation learning. Motivated by this, a novel method is proposed to recover the underlying representations by leveraging prototypes derived from unsupervised contrastive learning and clustering. The proposed noise-free prototypes are completely unaffected by noise, and are used to apply regularization constraints on the direction of representation learning, thereby refining and calibrating the learned representations. Extensive experiments demonstrate that the proposed method can obtain more reliable representations and effectively tackle the noisy labels. A key strength of our approach lies in the incorporation of unsupervised prior knowledge into noisy label learning through a plug-and-play design, enabling easy integration into broader weakly supervised learning fields such as missing labels and partial labels. Nonetheless, the current design focuses solely on instance-level calibration. In future work, we aim to extend our method to jointly calibrate both instance-level representations and overall cluster structures. Furthermore, scaling the approach to larger and more complex real-world noisy datasets represents another promising direction of future exploration.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China under Grant No. 62376281 and the Key NSF of China under Grant No. 62136005. Tingjin Luo is the corresponding author.

References

[1] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.

[2] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (6) (2016) 1137–1149.

[3] J. Li, R. Socher, S. C. Hoi, Dividemix: Learning with noisy labels as semi-supervised learning, in: International Conference on Learning Representations, 2020, pp. 1–14.

[4] D. Arpit, S. Jastrzębski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio, et al., A closer look at memorization in deep networks, in: International Conference on Machine Learning, PMLR, 2017, pp. 233–242.

[5] C. Zhang, S. Bengio, M. Hardt, B. Recht, O. Vinyals, Understanding deep learning (still) requires rethinking generalization, Communications of the ACM 64 (3) (2021) 107–115.

[6] H. Zhang, Q. Yao, Decoupling representation and classifier for noisy label learning, arXiv preprint arXiv:2011.08145 (2020) 1–12.

[7] L. Yi, S. Liu, Q. She, A. I. McLeod, B. Wang, On learning contrastive representations for learning with noisy labels, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 16682–16691.

[8] X.-S. Wei, S.-L. Xu, H. Chen, L. Xiao, Y. Peng, Prototype-based classifier learning for long-tailed visual recognition, Science China Information Sciences 65 (6) (2022) 160105.

[9] Y. Du, D. Zhou, Y. Xie, Y. Lei, J. Shi, Prototype-guided feature learning for unsupervised domain adaptation, Pattern Recognition 135 (2023) 109154.

[10] J. Han, P. Luo, X. Wang, Deep self-learning from noisy labels, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 5138–5147.

[11] X. Liu, B. Zhou, Z. Yue, C. Cheng, Plremix: Combating noisy labels with pseudo-label relaxed contrastive representation learning, in: 2025 IEEE/CVF Winter Conference on Applications of Computer Vision, IEEE, 2025, pp. 6517–6527.

[12] R. Zhu, H. Liu, R. Wu, M. Lin, T. Lv, C. Fan, H. Wang, Rethinking noisy label learning in real-world annotation scenarios from the noise-type perspective, arXiv preprint arXiv:2307.16889 (2023).

[13] X. Yang, H. Wang, J. Sun, S. Zhang, C. Chen, X.-S. Hua, X. Luo, Prototypical mixing and retrieval-based refinement for label noise-resistant image retrieval, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 11239–11249.

[14] H. Song, M. Kim, D. Park, Y. Shin, J.-G. Lee, Learning from noisy labels with deep neural networks: A survey, IEEE Transactions on Neural Networks and Learning Systems 34 (11) (2022) 8135–8153.

[15] R. Pu, Y. Sun, Y. Qin, Z. Ren, X. Song, H. Zheng, D. Peng, Robust self-paced hashing for cross-modal retrieval with noisy labels, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2025, pp. 19969–19977.

[16] S. Xu, Y. Sun, X. Li, S. Duan, Z. Ren, Z. Liu, D. Peng, Noisy label calibration for multi-view classification, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2025, pp. 21797–21805.

[17] Y. Sun, Y. Li, Z. Ren, G. Duan, D. Peng, P. Hu, Roll: Robust noisy pseudo-label learning for multi-view clustering with noisy correspondence, in: Proceedings of the Computer Vision and Pattern Recognition Conference, 2025, pp. 30732–30741.

[18] Z. Zhang, M. Sabuncu, Generalized cross entropy loss for training deep neural networks with noisy labels, Advances in Neural Information Processing Systems 31 (2018) 8792–8802.

[19] H. Wei, H. Zhuang, R. Xie, L. Feng, G. Niu, B. An, Y. Li, Mitigating memorization of noisy labels by clipping the model prediction, in: International Conference on Machine Learning, PMLR, 2023, pp. 36868–36886.

[20] X. Ye, Y. Wu, W. Zhang, X. Li, Y. Chen, C. Jin, Optimized gradient clipping for noisy label learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2025, pp. 9463–9471.

[21] X. Zhou, X. Liu, D. Zhai, J. Jiang, X. Ji, Asymmetric loss functions for noise-tolerant learning: Theory and applications, IEEE Transactions on Pattern Analysis and Machine Intelligence 45 (7) (2023) 8094–8109.

[22] J. Li, T.-W. Chang, K. Kuang, X. Li, L. Chen, J. Zhou, Learning causal transition matrix for instance-dependent label noise, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2025, pp. 18305–18313.

[23] Y. Liu, H. Cheng, K. Zhang, Identifiability of label noise transition matrix, in: International Conference on Machine Learning, PMLR, 2023, pp. 21475–21496.

[24] T. Liu, D. Tao, Classification with noisy labels by importance reweighting, in: IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 38, 2016, pp. 447–461.

[25] J. Goldberger, E. Ben-Reuven, Training deep neural-networks using a noise adaptation layer, in: International Conference on Learning Representations, 2017, pp. 1–9.

[26] X. Xia, T. Liu, N. Wang, B. Han, C. Gong, G. Niu, M. Sugiyama, Are anchor points really indispensable in label-noise learning?, Advances in Neural Information Processing Systems 32 (2019) 1–12.

[27] W. Pan, W. Wei, F. Zhu, Y. Deng, Enhanced sample selection with confidence tracking: Identifying correctly labeled yet hard-to-learn samples in noisy data, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2025, pp. 19795–19803.

[28] X. Xia, T. Liu, B. Han, M. Gong, J. Yu, G. Niu, M. Sugiyama, Sample selection with uncertainty of losses for learning with noisy labels, in: International Conference on Learning Representations, 2022, pp. 1–26.

[29] X. Xia, B. Han, Y. Zhan, J. Yu, M. Gong, C. Gong, T. Liu, Combating noisy labels with sample selection by mining high-discrepancy examples, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 1833–1843.

[30] H. Wei, L. Feng, X. Chen, B. An, Combating noisy labels by agreement: A joint training method with co-regularization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 13726–13735.

[31] X. Yu, B. Han, J. Yao, G. Niu, I. Tsang, M. Sugiyama, How does disagreement help generalization against label corruption?, in: International Conference on Machine Learning, PMLR, 2019, pp. 7164–7173.

[32] M. Sheng, Z. Sun, G. Pei, T. Chen, H. Luo, Y. Yao, Enhancing robustness in learning with noisy labels: An asymmetric co-training approach, in: Proceedings of the 32nd ACM International Conference on Multimedia, 2024, pp. 4406–4415.

[33] X. Zhou, X. Liu, D. Zhai, J. Jiang, X. Gao, X. Ji, Prototype-anchored learning for learning with imperfect annotations, in: International Conference on Machine Learning, Vol. 162, 2022, pp. 27245–27267.

[34] M. Ye, X. Zhang, P. C. Yuen, S.-F. Chang, Unsupervised embedding learning via invariant and spreading instance feature, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 6210–6219.

[35] K. He, H. Fan, Y. Wu, S. Xie, R. Girshick, Momentum contrast for unsupervised visual representation learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 9729–9738.

[36] T. Gao, X. Yao, D. Chen, SimCSE: Simple contrastive learning of sentence embeddings, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, 2021, pp. 6894–6910.

[37] J. Li, P. Zhou, C. Xiong, S. C. Hoi, Prototypical contrastive learning of unsupervised representations, in: International Conference on Learning Representations, 2021, pp. 1–12.

[38] Y. LeCun, C. Cortes, C. J. Burges, The mnist database of handwritten digits.

[39] H. Xiao, K. Rasul, R. Vollgraf, Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms, arXiv preprint arXiv:1708.07747 (2017).

[40] A. Krizhevsky, Learning multiple layers of features from tiny images, Technical Report (2009).

[41] J. Wang, X. Xia, L. Lan, X. Wu, J. Yu, W. Yang, B. Han, T. Liu, Tackling noisy labels with network parameter additive decomposition, IEEE Transactions on Pattern Analysis and Machine Intelligence 46 (9) (2024) 6341–6354.

[42] Z. Zhu, T. Liu, Y. Liu, A second-order approach to learning with instance-dependent label noise, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 10113–10123.

[43] L. Bossard, M. Guillaumin, L. Van Gool, Food-101–mining discriminative components with random forests, in: European Conference on Computer Vision, 2014, pp. 446–461.

[44] T. Xiao, T. Xia, Y. Yang, C. Huang, X. Wang, Learning from massive noisy labeled data for image classification, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2015, pp. 2691–2699.

[45] J. Wei, Z. Zhu, H. Cheng, T. Liu, G. Niu, Y. Liu, Learning with noisy labels revisited: A study using real-world human annotations, in: International Conference on Learning Representations, 2022, pp. 1–23.

[46] M. Zhang, X. Zhao, J. Yao, C. Yuan, W. Huang, When noisy labels meet long tail dilemmas: A representation calibration method, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 15890–15900.

[47] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, M. Sugiyama, Co-teaching: Robust training of deep neural networks with extremely noisy labels, Advances in Neural Information Processing Systems 31 (2018) 8536–8546.

[48] X. Xia, T. Liu, B. Han, C. Gong, N. Wang, Z. Ge, Y. Chang, Robust early-learning: Hindering the memorization of noisy labels, in: International Conference on Learning Representations, 2020, pp. 1–15.

[49] X. Xia, P. Lu, C. Gong, B. Han, J. Yu, J. Yu, T. Liu, Regularly truncated m-estimators for learning with noisy labels, IEEE Transactions on Pattern Analysis and Machine Intelligence 46 (5) (2024) 3522–3536.

[50] J. Wang, X. Zhou, D. Zhai, J. Jiang, X. Ji, X. Liu, ϵ-softmax: Approximating one-hot vectors for mitigating label noise, Advances in Neural Information Processing Systems 37 (2024) 32012–32038.

[51] L. Van der Maaten, G. Hinton, Visualizing data using t-sne, Journal of Machine Learning Research 9 (11) (2008) 2579–2605.

[52] D. Mandal, S. Bharadwaj, S. Biswas, A novel self-supervised re-labeling approach for training with noisy labels, in: 2020 IEEE/CVF Winter Conference on Applications of Computer Vision, 2020, pp. 1381–1390.

[53] C. Tan, J. Xia, L. Wu, S. Z. Li, Co-learning: Learning from noisy labels with self-supervision, in: Proceedings of the 29th ACM International Conference on Multimedia, 2021, pp. 1405–1413.

[54] E. Zheltonozhskii, C. Baskin, A. Mendelson, A. M. Bronstein, O. Litany, Contrast to divide: Self-supervised pre-training for learning with noisy labels, in: 2022 IEEE/CVF Winter Conference on Applications of Computer Vision, 2022, pp. 1657–1667.

[55] Y. Zhang, S. Zheng, P. Wu, M. Goswami, C. Chen, Learning with feature-dependent label noise: A progressive approach, in: International Conference on Learning Representations, 2021, pp. 1–13.

[56] Y. Li, H. Han, S. Shan, X. Chen, Disc: Learning from noisy labels via dynamic instance-specific selection and correction, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2023, pp. 24070–24079.