Tackling Instance-Dependent Label Noise with Class Rebalance and Geometric Regularization

Shuzhi Cao
Xi'an Jiaotong university
China
cao309615@gmail.com

Jianfei Ruan
Xi'an Jiaotong university
China
jianfei.ruan@hotmail.com

Bo Dong
Xi'an Jiaotong university
China
dong.bo@xjtu.edu.cn

Bin Shi
Xi'an Jiaotong university
China
shibin@xjtu.edu.cn

ABSTRACT

In label-noise learning, accurately identifying the transition matrix is crucial for developing statistically consistent classifiers. This task is challenged by instance-dependent noise, which introduces identifiability issues in the absence of strict assumptions. Existing methods use neural networks to estimate the transition matrix by first extracting confident clean instances. However, this extraction process suffers from severe inter-class imbalance and a bias towards selecting unambiguous inner-class instances, resulting in skewed noise pattern comprehension. To tackle these issues, our paper presents a Class Rebalance and Geometric Regularization-based Framework (CRGR). CRGR employs a smoothed noise-tolerant reweighting mechanism to balance inter-class representation, thus preventing model overfitting to dominant classes. Furthermore, recognizing that instances with similar characteristics often exhibit parallel noise patterns, we propose that the transition matrix should mirror the similarity of the feature space. This insight leads to the inclusion of ambiguous instances in training, serving as geometric regularization. Such a strategy enhances the model's ability to navigate various noise patterns and bolsters generalization. By addressing both inter-class and inner-class biases, CRGR presents a more equitable and robust classification model. Extensive experiments on both synthetic and real-world datasets demonstrate CRGR's superiority over existing state-of-the-art methods, significantly boosting classification accuracy and showcasing its effectiveness in handling instance-dependent noise.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**.

KEYWORDS

Instance-dependent label noise, Class rebalance, Geometric regularization, Confident clean instance

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

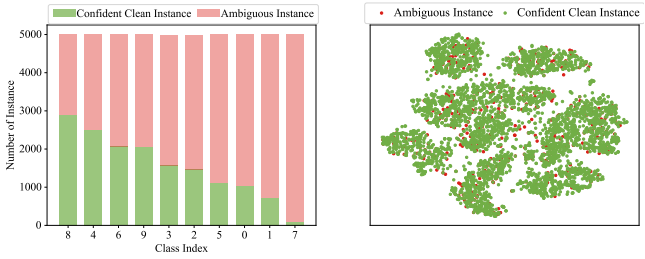
ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

ACM Reference Format:
Shuzhi Cao, Jianfei Ruan, Bo Dong, and Bin Shi. 2018. Tackling Instance-Dependent Label Noise with Class Rebalance and Geometric Regularization. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX)*. ACM, New York, NY, USA, 11 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Deep learning methods have excelled in various computer vision tasks, such as image classification [25, 34] and object detection [12, 27, 47], among others. Despite their success, these methods heavily rely on extensive, fully labeled datasets, which are costly to obtain due to the required expertise for accurate labeling. A common workaround involves sourcing large-scale training data via mobile crowdsourcing [16, 42] or online queries [3]. Nonetheless, these techniques tend to introduce label noise into the datasets [7, 43], which can significantly impair the generalization capabilities of deep learning models. The susceptibility of these models to noisy data, due to their complex architectures and strong fitting abilities, highlights the importance of developing robust algorithms that can effectively handle label noise.

Recent methods for addressing label noise can be categorized into heuristic approaches [19, 20, 24, 33, 36, 37, 45] and statistically consistent algorithms [6, 7, 39, 40, 43, 44]. Heuristic approaches, based on empirical techniques like selecting presumably clean data [14, 19, 20, 36, 37, 45], label correction [22, 31, 33], or integrating extra regularization constraints [24, 32], may offer practical efficacy but lack theoretical guarantees and may not converge to an optimal classifier [26, 40]. Contrastingly, statistically consistent algorithms aim to model label noise directly, estimating a transition matrix $T(\boldsymbol{x}) \in \mathbb{R}^{c\times c}$, where c represents the number of classes, and $T_{ij}(\boldsymbol{x}) = P(\bar{Y} = j|Y = i, X = \boldsymbol{x})$ represents the probability of an instance with true label i being mislabeled as noisy label j, given its features $\boldsymbol{x}$. This matrix allows for the inference of clean class posterior probabilities from noisy ones, thereby constructing a consistent classifier. To simplify the form of $T(\boldsymbol{x})$, some researchers present class-conditional noise (CCN) model [26, 40, 44], which assumes a constant mislabeling probability across instances within a class, i.e., $P(\bar{Y} = j|Y = i, X = \boldsymbol{x}) = P(\bar{Y} = j|Y = i)$. However, this label noise model does not take into account the impact of instance features on noisy labels, making it difficult to describe real-world noise patterns. To mitigate this issue, instance-dependent noise



(a) (b)

Figure 1: (a) exhibits the imbalanced inter-class distribution of confident clean instances in *CIFAR-10*. (b) visualizes the distributions of the confident clean and ambiguous instances to show the inner-class selection bias problem by using the T-SNE technique.

(IDN) [5, 6, 39, 43] is proposed, which accounts for variability in mislabeling probabilities among instances, even within the same class, based on their true class and specific features. This paper primarily focuses on IDN due to its broader practical relevance and realism.

Estimating $T(\boldsymbol{x})$ under IDN conditions is challenging due to its reliance on instance-specific features, leading to a wide array of potential matrix configurations. Unlike CCN, IDN lacks the convenience of using anchor points [30, 40] for a fixed matrix estimation. Researchers have tried to circumvent this complexity by introducing restrictive assumptions like part-dependent noise [39], confidence scores [2], or noise bounds [9], but these solutions often compromise practical utility. To mitigate reliance on such assumptions, recent advancements [6, 43] employ neural networks to adaptively fit transition matrix $T(\boldsymbol{x})$, using a subset of confident clean instances as a learning foundation without extra assumptions. Methods such as distillation [9], early-stop strategies [1], and small-loss-based techniques [14] have been used to gather these instances. However, these methods tend to select clear-cut instances from easily identifiable categories [20, 37], leading to two main challenges: **(1) Inter-class imbalance:** As Fig.1a shows, there exists a significant disparity in the number of confident clean instances across classes, causing model overfitting to overrepresented classes. **(2) Inner-class selection bias:** Fig.1b illustrates the selection of easily identifiable instances as confident clean ones, ignoring ambiguous instances near the decision boundaries. This leads to an inadequate representation of complex noise generation patterns. These issues pose challenges to current state-of-the-art approaches [6, 43] that utilize the neural network to fit $T(\boldsymbol{x})$, as they directly train the neural network on such biased extracted instances, leading to a skewed classifier.

To tackle these issues, we propose a novel label noise learning method CRGR. Our first solution targets the imbalance in the inter-class distribution of extracted instances. We propose a smoothed noise-tolerant reweighting strategy that equalizes the neural network's exposure to noise generation patterns across different classes. The second solution draws inspiration from psychological and physiological studies [10, 28], which suggest that instances sharing similar features are more likely to be mislabeled into the same class and vice versa. This observation guides us to posit that $T(\boldsymbol{x})$ should mirror the similarity of the feature space, ensuring that the similarity relation between instances in the feature space aligns consistently with those in the transition matrix space. This insight leads us to further include ambiguous instances in the training process as geometric regularization, aiding the network in grasping complex noise dynamics. In summary, CRGR refines the neural network's training and enhances its capacity to depict noise patterns, leading to a more equitable and robust classifier. The main contributions are shown as follows:

- We propose a novel label noise learning method, CRGR, which enhances the neural network's capacity to depict comprehensive noise patterns by solving both the inter-class imbalance and inner-class selection bias problems among the extracted confident clean instances and ultimately results in a more equitable and robust classifier.
- In this paper, we first propose a novel smoothed noise-tolerant reweighting strategy that equalizes the network's exposure to noise patterns in each class to balance inter-class representation. Subsequently, recognizing that instances with similar features often exhibit parallel noise patterns, we propose that the transition matrix should mirror the similarity of the feature space. This insight guides us to leverage the characteristic information of ambiguous instances and incorporate them into training as geometric regularization. These two enhancements improve the model's ability to depict complex noise patterns, eliminating both the inter-class and inner-class fitting bias.
- Extensive experiments on both synthetic and real-world datasets demonstrate that CRGR outperforms existing state-of-the-art methods in handling IDN.

The remainder of the paper is organized as follows: Section 2 reviews related work. Section 3 describes preliminary concepts and notations. Section 4 details the proposed method. Section 5 presents experimental results and discussions. Finally, Section 6 concludes the paper.

2 RELATED WORK

In this section, we provide a concise overview of the relevant literature pertaining to label noise models and learning with IDN.

Label noise models. The label noise model depicts the process of generating noisy labels. In general, label noise models are commonly categorized into class-conditional noise (CCN) [7, 26, 30, 40, 44] and instance-dependent noise (IDN) [5, 6, 9, 39, 43]. CCN considers that an instance's noisy label is exclusively related to its true class. To be precise, CCN ignores the effect of instance features on noisy labels and assumes that all instances in the same class share the same noise generation patterns. As the result, all the instances belonging to the same class have the fixed probability to be mislabeled as the other class. In contrast, IDN considers the probability of mislabeling to vary among instances, even within the same class, depending on their true class and specific features. In comparison, IDN is more realistic [5, 39] since humans always assign labels to instances based on their unique features. For example, during the annotation process, fuzzy photos with less information are more likely to be mislabeled as other categories. Despite the realism of

the IDN model, learning with IDN is formidable since the transition matrix $T(\boldsymbol{x})$, which plays an essential role in building statistically consistent classifiers, is unidentifiable under IDN without extra assumptions.

Learning with IDN. In summary, existing IDN-based label noise learning methods can be classified into two categories: heuristics approaches [14, 19, 20, 22, 31, 33, 36, 37, 45] and statistically consistent algorithms [6, 7, 26, 30, 39, 40, 43, 44]. Heuristics utilizes empirical techniques, such as selecting presumably clean data [14, 19, 20, 36, 37, 45], label correction [22, 31, 33], or adding extra regularization constraints [24, 32] to mitigate the adverse effects of label noise. While these approaches may empirically work well, their performance ceiling remains limited as they lack theoretical guarantees and may not converge to the ideal classifier that would result from using accurately labeled data. Driven by this concern, the development of statistically consistent algorithms has emerged, aiming to build consistent classifiers. In this pursuit, the estimation of the transition matrix $T(\boldsymbol{x})$ plays a crucial role. Under IDN, $T(\boldsymbol{x})$ is unidentifiable without extra constraints, hence, to uniquely ascertain the $T(\boldsymbol{x})$, several extra assumptions have been proposed. For instance, [39] hypothesis that the noise of an instance depends only on its parts; [2] necessitates additional confident scores, and [9] studies a special case of IDN where the noise rate has an upper bound. While these methods partially address the matrix estimation problem, their practical application is impeded by the heavy dependence on assumptions. To get rid of the extra assumptions, recent advancements like [6, 43] employ extra neural networks to adaptively fit $T(\boldsymbol{x})$. These approaches leverage a subset of confident clean instances as a foundation for learning noise patterns without extra presuppositions, enhancing their applicability and performance. Despite these advantages, these approaches encounter two significant challenges. First, the imbalanced inter-class distributions of extracted instances lead the neural network to overfit to classes with more instances and overlook others. Second, current methods tend to select clean-cut instances inner-class as extracted instances and neglect ambiguous instances. This selection bias poses a challenge for the network to learn intricate noise patterns, leading to a skewed classifier. As a result, how to handle IDN remains a challenging problem.

3 PRELIMINARIES

In this section, we first present the definition of label noise learning, followed by systematic formulations of transition matrix and confident clean instances.

Problem setting. Let D be the distribution of a pair of random variables $(X, Y) \in \mathcal{X} \times \mathcal{Y}$, where X denotes the random variable of instances, Y is the corresponding clean labels. Moreover, $\mathcal{X} \in \mathbb{R}^d$ is defined as the feature space, and $\mathcal{Y} = \{1, 2, ..., c\}$ is the label space, where d and c stands for the dimension of feature space and number of classes respectively. In the task of classification, given an instance $\boldsymbol{x} \in \mathcal{X}$, our goal aims to predict its true label $y \in \mathcal{Y}$. However, in real-world scenarios, collecting large-scale training data through crowdsourcing or online queries inevitably introduces label noise. Therefore, we define $\bar{D}$ the distribution of noisy instances $(X, \bar{Y}) \in \mathcal{X} \times \bar{\mathcal{Y}}$, where $\bar{Y}$ denotes the variable of noisy labels. In IDN, only a noisy training dataset $\bar{\mathcal{D}} = \{\boldsymbol{x}_i, \bar{y}_i\}_{i=1}^n$ with n instances that independently drawn from the distribution $\bar{D}$ is available.

Transition matrix. The transition matrix $T(\boldsymbol{x}) \in \mathcal{T}$ depicts the generation process of the noisy label of instance $\boldsymbol{x}$, where $\mathcal{T} \in \mathbb{R}^{c \times c}$ stands for the transition matrix space, and the ij-th entry of the matrix, i.e., $T_{ij}(\boldsymbol{x}) = P(\bar{Y} = j | Y = i, X = \boldsymbol{x})$, represents the probability that an instance $\boldsymbol{x}$ belongs to the true class $Y = i$ be mislabeled as noisy class $\bar{Y} = j$. In fact, the clean class posterior $P(Y|\boldsymbol{x}) = [P(Y = 1|X = \boldsymbol{x}), ..., P(Y = c|X = \boldsymbol{x})]^\top$ can be inferred by the noisy class posterior $P(\bar{Y}|\boldsymbol{x}) = [P((\bar{Y} = 1|X = \boldsymbol{x}), ..., P((\bar{Y} = c|X = \boldsymbol{x})]^\top$ and the transition matrix $T(\boldsymbol{x})$,i.e., $P(\bar{Y}|\boldsymbol{x}) = T(\boldsymbol{x})P(Y|\boldsymbol{x})$. By learning with noisy instances, the noisy class posterior $P(\bar{Y}|\boldsymbol{x})$ can be directly estimated. As a result, once the $T(\boldsymbol{x})$ is identified, constructing statistically consistent classifiers becomes a straightforward task. Nevertheless, $T(\boldsymbol{x})$ is unidentifiable under IDN without extra assumptions, how to uniquely ascertain $T(\boldsymbol{x})$ remains a non-trivial task.

Confident clean instances. Clean instances are indispensable for training the transition network $T(\boldsymbol{x}; \theta)$ that fits the transition matrix $T(\boldsymbol{x})$. However, in real-world scenarios, obtaining even a small number of clean instances is difficult sometimes. When the clean data is unavailable, it is necessary to extract a subset of instances with confident true labels, i.e., confident clean instances, from the noisy data automatically. Current off-the-shelf extracting approaches include techniques like the distillation method [9, 43], sample sieve approach [8], small-loss based methods [14, 45], and early-stop strategy [1, 18, 38]. In this paper, we directly employ the distillation method [9] to extract confident clean instances.

4 METHOD

This section provides an in-depth examination of CRGR, which enhances the neural network's capacity to depict noise patterns by resolving both inter-class imbalance and inner-class selection bias issues among the extracted instances, leading to a more equitable and robust classifier. To be specific, we first extract a subset of confident clean instances as training instances (Section 4.1). Secondly, we propose a novel smoothed noise-tolerant reweighting technique (Section 4.2) to calibrate the imbalanced inter-class distribution of extracted instances. Subsequently, by keeping the similarity relation between instances in the features space to be consistent with those in the transition matrix space, we further include the ambiguous instances in the training framework as a geometric regularization, which assists the transition network in simulating the complicated noise generation patterns (Section 4.3). Finally, with the well-trained transition network, we exploit the F-Correction [30] paradigm to access a consistent classifier (Section 4.4).

4.1 Confident Clean Instance Extraction

Confident clean instances, defined as instances with clean labels, are indispensable for transition network training. However, in real-world scenarios, acquiring even a small number of clean instances is challenging. Motivated by this fact, some empirical techniques [1, 9, 18, 38] have attempted to extract clean instances from noisy data automatically. In this paper, we employ an off-the-shelf distillation method [9] to extract a subset of confident clean instances $\mathcal{D}_{clean} = \left\{\boldsymbol{x}_i, \bar{y}_i, y_i^*\right\}_{i=1}^{n_c}$, where y^* represents the estimated latent

clean label, n_c denotes the nuumber of extracted instances. To be precise, for all the instance $\boldsymbol{x}$ in the noisy training data $\bar{\mathcal{D}}$, we calculate its corresponding noisy class posterior $P(\bar{Y}|\boldsymbol{x}) = [P(\bar{Y} = 1|X = \boldsymbol{x}), ..., P(\bar{Y} = c|X = \boldsymbol{x})]^\top$ at first and then select the instances who satisfies $max\left\{P(\bar{Y}|\boldsymbol{x})\right\} > \frac{1+\rho_{max}}{2}$ as confident clean instances while taking the remaining instances as ambiguous instances, where ρ_{max} is a pre-defined threshold. Simultaneously, $y^* = \text{argmax}_{\bar{y}\in\{1,2,...,c\}} P(\bar{Y} = \bar{y}|\boldsymbol{x})$ is taken as the estimated latent clean label. According to the above criteria, the original noisy training data $\bar{\mathcal{D}}$ is divided into confident clean instances $\mathcal{D}_{clean}$ and ambiguous instances $\mathcal{D}_{ambig}$,i.e., $\bar{\mathcal{D}} = \mathcal{D}_{clean} \cup \mathcal{D}_{ambig}$. With the extracted confident clean instances, we are able to train the transition network $T(\boldsymbol{x};\theta)$. To be specific, for the confident clean instance $\boldsymbol{x}$, by minimizing the empirical risk on the inferred noisy label $\boldsymbol{y}^* T(\boldsymbol{x};\theta)$ and its ground-truth noisy label $\bar{\boldsymbol{y}}$, the transition network is optimized. The empirical risk is formulated as follows:

$$\mathcal{R}(\theta) = -\frac{1}{n_c}\sum_{i=1}^{n_c} \bar{\boldsymbol{y}}_{\boldsymbol{i}} \log(\boldsymbol{y}_{\boldsymbol{i}}^* \cdot T(\boldsymbol{x}_{\boldsymbol{i}};\theta)), \tag{1}$$

where $\bar{\boldsymbol{y}}_{\boldsymbol{i}} \in \mathbb{R}^{1\times c}$ and $\boldsymbol{y}_{\boldsymbol{i}}^* \in \mathbb{R}^{1\times c}$ denotes the noisy label $\bar{y}_i$ and the clean label y_i^* in one-hot vector form respectively.

4.2 Class Rebalance

Despite the fact that transition network $T(\boldsymbol{x};\theta)$ can be trained by minimizing the risk formulated by Eq.(1), there still exists a main problem with this optimization approach. To be precise, considering the severe imbalanced inter-class distribution of extracted confident clean instances, directly minimizing the risk in Eq.(1) inevitably leads the neural network to overfit the classes with more instances and overlook others. As a consequence, the transition network has a skewed comprehension of noise generation patterns, and thus its generalization ability is greatly degraded. To tackle this issue, we propose a novel smoothed noise-tolerant reweighting technique to calibrate the imbalanced inter-class distribution. By assigning different weights to different instances, we transform the empirical risk in Eq.(1) to a cost-sensitive risk [11, 13], which is illustrated as follows:

$$\mathcal{R}_{weighted}(\theta) = -\frac{1}{n_c}\sum_{i=1}^{n_c} w(\boldsymbol{x}_{\boldsymbol{i}})\, \bar{\boldsymbol{y}}_{\boldsymbol{i}} \log\left((\boldsymbol{y}_{\boldsymbol{i}}^* \cdot T(\boldsymbol{x}_{\boldsymbol{i}};\theta)\right), \tag{2}$$

where $w(\boldsymbol{x})$ represents the weight corresponding to the instance $\boldsymbol{x}$. In general, greater weight should be given to the class with fewer extracted instances to equalize the imbalanced inter-class distribution. However, under IDN, the estimated latent clean label y^* may not be consistent with its clean label. As a result, directly using the inverse class frequency [17, 35] to weight instances may lead the transition network to erroneously concentrate misestimated labels. To mitigate this issue, we use the clean class posterior $P(Y = y^*|\boldsymbol{x})$ to evaluate the confidence degree of the estimated latent clean label y^* and assign different weights to various instances based on their label confidence. Based on this criterion, we determine the weight $w(x)$ as follows:

$$w(\boldsymbol{x}) = \begin{cases} \frac{n_c}{\sum_{i=1}^{n_c} 1[y_i^* = y^*]}, & \text{if } P(Y = y^*|\boldsymbol{x}) \geq w_{max}, \\ \sqrt{\frac{n_c}{\sum_{i=1}^{n_c} 1[y_i^* = y^*]}}, & \text{if } P(Y = y^*|\boldsymbol{x}) < w_{max}, \end{cases} \tag{3}$$

where $1[\cdot]$ is the indicator function, and w_{max} is a self-adaptive coefficient, which is defined as follows:

$$w_{max} = \frac{1}{n_c}\sum_{i=1}^{n_c} P(Y = y_i^*|X = \boldsymbol{x}_{\boldsymbol{i}}), \tag{4}$$

As previously demonstrated, we directly employ the inverse class frequency to weight the instance whose clean class posterior exceeds the threshold w_{max}. In the case of instances whose clean class posterior is less than the threshold, we utilize the smoothed form of the inverse square root of the class frequency to weight these instances to prevent the latent overfitting problem. Hence, by optimizing the weighted risk in Eq.(2), we equalize the transition network's exposure to noise generation patterns in each class to overcome the overfitting problem, leading to class rebalance.

4.3 Geometric Regularization

Existing approaches tend to extract easily identifiable instances inner each class as confident clean instances and neglect ambiguous instances lying around the decision boundary. Hence, the transition network is unable to capture the intricate noise patterns since it does not encounter any ambiguous instances during the training process. To tackle this issue, we consider leveraging the information of ambiguous instances $\mathcal{D}_{ambig}$ to guide the network training. To be specific, inspired by the evidence [10, 28] that instances with similar characteristics often exhibit parallel noise patterns, we propose that the transition matrix $T(\boldsymbol{x})$ should mirror the similarity of the feature space. i.e., keeping the similarity relation between instances in the features space to be consistent with those in the transition matrix space. This insight leads to the inclusion of ambiguous instances in training, serving as geometric regularization. Specifically, we use $d_{\mathcal{X}}(\boldsymbol{x}_{\boldsymbol{i}}, \boldsymbol{x}_{\boldsymbol{j}}) = ||\boldsymbol{x}_{\boldsymbol{i}} - \boldsymbol{x}_{\boldsymbol{j}}||_2$ and $d_{\mathcal{T}}(T(\boldsymbol{x}_{\boldsymbol{i}}), T(\boldsymbol{x}_{\boldsymbol{j}})) = ||T(\boldsymbol{x}_{\boldsymbol{i}}) - T(\boldsymbol{x}_{\boldsymbol{j}})||_2$ to measure the similarity between the instances $\boldsymbol{x}_{\boldsymbol{i}}$ and $\boldsymbol{x}_{\boldsymbol{j}}$ in feature and transition matrix spaces respectively, where $||\cdot||_2$ represents the ℓ_2 norm of vector. Following the property that $T(\boldsymbol{x})$ mirror the similarity of the feature space, we claim that the distance between instances in the feature and transition matrix space should be proportional and thus have the following relationship:

$$d_{\mathcal{X}}(\boldsymbol{x}_{\boldsymbol{i}}, \boldsymbol{x}_{\boldsymbol{j}}) = \kappa \cdot d_{\mathcal{T}}(T(\boldsymbol{x}_{\boldsymbol{i}}), T(\boldsymbol{x}_{\boldsymbol{j}})), \tag{5}$$

where $\kappa \in \mathbb{R}$ is a constant scaling factor, $\mathcal{X}$ and $\mathcal{T}$ denote the feature and transition matrix space, respectively. Considering computational efficiency, we only calculate the distance between the instance $\boldsymbol{x}$ and the center of each class to measure its global similarity relations. To be specific, let μ_i ($i \in \{1, ..., c\}$) denotes the center of class i, which can be estimated using confident clean instances as follows:

$$\mu_i = \frac{\sum_{(\boldsymbol{x}_j, y_j^*)\in\mathcal{D}_{clean}} \boldsymbol{x}_{\boldsymbol{j}} \cdot 1(y_j^* = i)}{\sum_{(\boldsymbol{x}_j, y_j^*)\in\mathcal{D}_{clean}} 1(y_j^* = i)}. \tag{6}$$

We define vectors $S_{\mathcal{X}}(\boldsymbol{x}) = [d_{\mathcal{X}}(\boldsymbol{x}, \boldsymbol{\mu_1}), ..., d_{\mathcal{X}}(\boldsymbol{x}, \boldsymbol{\mu_c})]$ and $S_{\mathcal{T}}(\boldsymbol{x}) = [d_{\mathcal{T}}(T(\boldsymbol{x}), T(\boldsymbol{\mu_1})), ..., d_{\mathcal{T}}(T(\boldsymbol{x}), T(\boldsymbol{\mu_c}))]$ to reflect global similarity relations of $\boldsymbol{x}$ and use the transition network's prediction $T(\boldsymbol{x};\theta)$ to approximate its ground truth. Following the Eq.(5), $\kappa \cdot S_{\mathcal{T}}(\boldsymbol{x})$ should be consistent with $S_{\mathcal{X}}(\boldsymbol{x})$. To eliminate the effect of scale discrepancy, we minimize the difference between the normalized

$S_{\mathcal{X}}(\boldsymbol{x})$ and $S_{\mathcal{T}}(\boldsymbol{x})$ and offer a geometric regularization as follows:

$$\mathcal{R}_{regular}(\theta) = -\frac{1}{n}\sum_{i=1}^{n} || \frac{S_{\mathcal{X}}(\boldsymbol{x_i})}{||S_{\mathcal{X}}(\boldsymbol{x_i})||_2} - \frac{S_{\mathcal{T}}(\boldsymbol{x_i})}{||S_{\mathcal{T}}(\boldsymbol{x_i})||_2} ||_2. \quad (7)$$

Finally, the overall objective function can be expressed as Eq.(8), where α is the hyper-parameter to balance the $\mathcal{R}_{weighted}$ and $\mathcal{R}_{regular}$.

$$\min_{\theta} \mathcal{L}(\theta) = \mathcal{R}_{weighted}(\theta) + \alpha \mathcal{R}_{regular}(\theta). \quad (8)$$

By minimizing the overall objective function mentioned above, the parameter θ of the transition network $T(x)$ can be learned.

4.4 Classifier Training

By optimizing the overall object function in Eq.(8), the transition network $T(\boldsymbol{x};\theta)$ is well trained. Our ultimate goal is to obtain a classifier $f(\boldsymbol{x};\omega)$ parameterized by ω that can predict the clean class posterior of the instance $\boldsymbol{x}$, i.e., $P(Y|\boldsymbol{x}) = f(\boldsymbol{x};\omega)$. Specifically, the noisy class posterior $P(\bar{Y}|\boldsymbol{x})$ can be inferred as $P(\bar{Y}|\boldsymbol{x}) = T(\boldsymbol{x})P(Y|\boldsymbol{x})$. Here, we approximate $T(\boldsymbol{x})$ and $P(Y|\boldsymbol{x})$ by using $T(\boldsymbol{x};\theta)$ and $f(\boldsymbol{x};\omega)$ respectively, where θ is a fixed parameter and ω is a trainable parameter. Furthermore, we use them to approximate $P(\bar{Y}|\boldsymbol{x})$ and calculate the cross-entropy loss between the inferred noisy class posterior and the given noisy label to learn the parameter ω of the classifier $f(\boldsymbol{x};\omega)$ as follows:

$$\min_{\omega} \mathcal{R}(\omega) = -\frac{1}{n}\sum_{i=1}^{n} \bar{\boldsymbol{y}}_{\boldsymbol{i}} \log(f(\boldsymbol{x_i};\omega) \cdot T(\boldsymbol{x_i};\theta)), \quad (9)$$

In conclusion, the details of CRGR are summarized in Algorithm 1 with a visual representation shown in Fig.2.

Algorithm 1 CRGR

Input: Noisy training dataset $\bar{\mathcal{D}} = \{\boldsymbol{x_i}, \bar{y}_i\}_{i=1}^{n}$

1: Warm up the classifier $f(\boldsymbol{x};\omega)$ with early-stop strategy and divide $\bar{\mathcal{D}}$ into confident clean instances $\mathcal{D}_{clean}$ and ambiguous instances $\mathcal{D}_{ambig}$.
2: **while** $epoch \leq Max\text{-}Epoch$ **do**
3: Calculate the risk $\mathcal{R}_{weighted}(\theta)$ shown in Eq.(2) and the risk $\mathcal{R}_{regular}(\theta)$ shown in Eq.(7).
4: Minimize the overall object function shown in Eq.(8) to learn the parameter θ of transition network;
5: **end while**
6: Fix the learned θ and optimize the classifier $f(\boldsymbol{x};\omega)$ by minimizing the risk shown in Eq.(9).

Output: The classifier $f(\boldsymbol{x};\omega)$

5 EXPERIMENTS

In this section, we conduct comprehensive experiments on both synthetic and real-world instance-dependent noisy datasets to verify the effectiveness and superiority of the proposed GRTR method.

5.1 Experiment Setup

Datasets We conduct extensive experiments on synthetic noisy datasets, i.e., *SVHN*, *CIFAR-10* and *CIFAR-100*, and real-world noisy dataset *Clothing-1M* to verify the effectiveness of the proposed CRGR method. *SVHN* has 10 classes of images with 73,257 training instances and 26,032 test instances of varying sizes, *CIFAR-10* and *CIFAR-100* contain 10 and 100 classes respectively and both of them have 50k training images and 10k test images of size 32×32. As for the real-world dataset, *Clothing-1M* is a large-scale image dataset with 1M training images that contain noisy labels collected by crowdsourcing and online queries and 10k images with clean labels for testing.

Algorithm 2 Instance-dependent Label Noise Generation

Input: Clean instances $\mathcal{D} = \{\boldsymbol{x_i}, y_i\}_{i=1}^{n}$; Noise rate τ.

1: Sample instance flip rates $q \in R^n$ from the truncated normal distribution $\mathcal{N}(\tau, 0.1^2, [0,1])$;
2: Independently sample $w_1, w_2, ..., w_c$ from the standard normal distribution $\mathcal{N}(0, 1^2)$;
3: **for** $i = 0$ to n **do**
4: $p = \boldsymbol{x_i} \times w_{y_i}$; // generate instance-dependent flip rates
5: $p_{y_i} = -\infty$; // control the diagonal entry of the instance-dependent transition matrix
6: $p = q_i \times softmax(p)$; // make the sum of the off-diagonal entries of the y_i-th row to be q_i
7: $p_{y_i} = 1 - q_i$; // set the diagonal entry to be $1 - q_i$
8: Randomly choose a label from the label space according to possibilities p as the noisy label $\bar{y}_i$;
9: **end for**

Output: Noisy instances $\bar{\mathcal{D}} = \{\boldsymbol{x_i}, \bar{y}_i\}_{i=1}^{n}$

IDN generation In our experiments, we utilize the following commonly used instance-dependent label noise generation approach (see in Algorithm 2) to generate synthetic noisy datasets. To be specific, given the clean instances $\mathcal{D} = \{\boldsymbol{x_i}, y_i\}_{i=1}^{n}$ and noise rate τ, we first sample instance flip rates (noise rates) $q \in R^n$ from the truncated normal distribution $\mathcal{N}(\tau, 0.1^2, [0,1])$ for each instance, where the average flip rate is set as τ. Subsequently, we independently sample parameters $w_1, w_2, ..., w_c$ from the standard normal distribution $\mathcal{N}(0, 1^2)$ for generating instance-dependent label noise, where the dimensionality of each parameter is $d \times c$, d denotes the dimensionality of the instance, and c stands for the number of classes. With the sampled parameter w_i $i \in \{1, 2, ..., c\}$, we are able to generate instance-dependent flip rates for all the instances that belong to the class $Y = i$. Specifically, for the instance x_i with clean label y_i, its noisy label's generation is only related to the y_i-th row of the transition matrix, we use vector p to represent the y_i-th row of the transition matrix. In order to calculate p for each instance, in Steps 4-7, we first use parameter w_i and the feature x_i to initialize $p = \boldsymbol{x_i} \times w_{y_i}$, and then we set the diagonal entry $p_{y_i} = 1 - q_i$ while making the sum of the off-diagonal entries of y_i-th row to be q_i. Finally, according to the possibilities p, we randomly choose noisy labels for all the instances to generate synthetic noisy datasets. For all the experiments on synthetic datasets, we split the training images into the training set and validation set according to the ratio

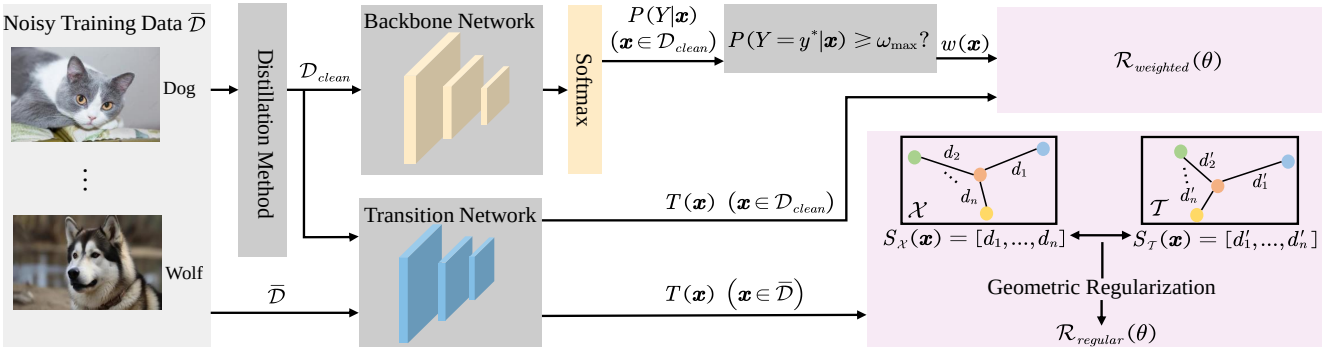


Figure 2: The overview of CRGR.

Table 1: Means and standard deviations (percentage) of classification accuracy on *CIFAR-10* with different label noise levels.

Method	IDN-10%	IDN-20%	IDN-30%	IDN-40%	IDN-50%
CE	74.31 ± 0.39	68.11 ± 0.81	59.63 ± 0.98	49.56 ± 1.57	39.22 ± 2.79
Co-teaching+	73.95 ± 0.98	70.25 ± 1.27	64.36 ± 1.49	47.11 ± 2.31	39.26 ± 2.98
JoCoR	76.21 ± 0.47	73.56 ± 0.51	68.21 ± 0.49	53.69 ± 3.21	43.22 ± 4.09
T-Revision	77.72 ± 0.31	75.31 ± 0.43	73.51 ± 0.55	57.92 ± 1.02	50.10 ± 2.76
CCR	74.47 ± 0.19	68.47 ± 0.32	65.46 ± 0.29	56.07 ± 0.89	48.05 ± 1.49
VolMinNet	80.02 ± 0.23	76.73 ± 0.31	72.02 ± 0.44	65.83 ± 0.97	49.96 ± 1.21
PTD	78.36 ± 0.43	76.69 ± 0.39	72.04 ± 0.45	58.32 ± 1.32	42.69 ± 2.99
BLTM	80.01 ± 0.34	79.58 ± 0.23	73.17 ± 1.02	64.78 ± 1.92	56.49 ± 3.69
MEIDTM	79.37 ± 0.39	70.65 ± 0.47	62.43 ± 0.41	52.98 ± 1.23	42.97 ± 2.92
CRGR	**81.51 ± 0.57**	**80.01 ± 0.51**	**77.01 ± 1.75**	**68.28 ± 4.06**	**60.86 ± 3.18**

of 4:1 and conducted five repeated experiments with different seeds to make experimental results dependable.

Implementation details. We regard ResNet-34 [15] as the backbone network for all the synthetic noisy datasets and use ResNet-50 [15] for real-world noisy dataset *Clothing-1M*. As for the transition network, while maintaining the overall architecture of the backbone network, we make a modification to the last linear layer to accommodate the specific shape of the transition matrix. For all the experiments, we first use the early-stop strategy [1, 38] to warm up the backbone network for five epochs on the noisy dataset and then extract confident clean instances with the pre-defined threshold $\rho_{max} = 0.3$ referring to the previous work [43]. Subsequently, we use the SGD [4] optimizer with a momentum of 0.9, batch size of 128, and an initial learning rate of 0.01 to train the transition network. Finally, we fix the well-trained transition network to learn the parameters of the classification network. In this stage, the classification network is trained by utilizing the Adam [21] optimizer with weight decay 1×10^{-4}, batch size of 128, and learning rate 5×10^{-7}. For a fair comparison, it is worth noting we do not use any data augmentation technique in all the experiments on synthetic noisy datasets as in [39]. All the experiments documented in this paper are executed using Pytorch [29] on two GPUs (NVIDIA RTX 3090) functioning in parallel.

Comparison methods. To demonstrate the superiority of the proposed CRGR method, we compare it with the following approaches: (1) CE, which trains the classifier directly on the noisy dataset with the standard cross-entropy loss and is considered as baseline; (2) Co-teaching+ [45], which simultaneously develops two neural networks to select small-loss instances with prediction disagreement for network training; (3) JoCoR [36], which adopts a joint training method with co-regularization; (4) T-Revision [40], which first initialize transition matrix by exploiting instances that are similar to anchor points and then introduces a slack variable to modify the initial estimator; (5) CCR [7], which estimates the transition matrix under a forward-backward cycle-consistency regularization and constructs the classifier; (6) VolMinNet [26], which solves label noise learning via optimizing the volume of the simplex formed by the columns of the transition matrix; (7) PTD [39], which approximates the IDN by exploiting part-dependent label noise; (8) BLTM [43], which estimates the IDN transition matrix using a deep neural network; (9) MEIDTM [6], which proposes a

Table 2: Means and standard deviations (percentage) of classification accuracy on *SVHN* with different label noise levels.

Method	IDN-10%	IDN-20%	IDN-30%	IDN-40%	IDN-50%
CE	90.27 ± 0.41	89.69 ± 0.53	85.99 ± 0.97	65.24 ± 1.69	48.76 ± 4.12
Co-teaching+	92.96 ± 0.77	91.65 ± 0.81	85.77 ± 1.98	56.33 ± 1.77	42.68 ± 3.38
JoCoR	88.57 ± 0.54	80.26 ± 0.67	77.52 ± 0.99	65.86 ± 1.92	47.52 ± 3.82
T-Revision	92.84 ± 0.33	92.33 ± 0.71	83.16 ± 0.88	73.60 ± 1.45	66.91 ± 4.49
CCR	92.64 ± 0.39	90.46 ± 0.57	88.69 ± 0.93	76.84 ± 1.85	70.24 ± 2.01
VolMinNet	93.89 ± 0.21	93.23 ± 0.53	78.37 ± 0.49	69.21 ± 0.99	56.91 ± 3.09
PTD	94.71 ± 0.57	94.33 ± 0.66	91.11 ± 0.89	90.32 ± 1.44	53.65 ± 4.32
BLTM	94.35 ± 0.42	91.55 ± 1.29	89.78 ± 3.58	83.71 ± 3.99	70.41 ± 6.53
MEIDTM	91.74 ± 0.65	85.52 ± 0.72	79.71 ± 1.64	66.22 ± 3.12	60.76 ± 4.22
CRGR	**95.12 ± 0.33**	**94.59 ± 0.88**	**93.23 ± 1.13**	**91.35 ± 1.99**	**77.58 ± 4.24**

Table 3: Means and standard deviations (percentage) of classification accuracy on *CIFAR-100* with different label noise levels.

	IDN-10%	IDN-20%	IDN-30%	IDN-40%	IDN-50%
CE	46.45 ± 0.29	41.51 ± 0.55	38.71 ± 1.76	30.01 ± 0.98	27.81 ± 1.86
Co-teaching+	49.05 ± 0.58	43.21 ± 1.33	41.55 ± 1.01	38.11 ± 0.95	30.09 ± 2.55
JoCoR	49.31 ± 0.27	43.55 ± 0.77	40.11 ± 0.92	37.21 ± 1.82	31.02 ± 2.34
T-Revision	47.74 ± 0.54	43.31 ± 0.91	36.19 ± 1.56	29.25 ± 1.33	25.23 ± 1.19
CCR	45.28 ± 0.19	40.72 ± 0.24	34.68 ± 0, 58	28.53 ± 0.97	27.08 ± 1.22
VolMinNet	49.17 ± 0.32	47.02 ± 0.27	43.11 ± 0.94	38.62 ± 1.41	29.88 ± 1.88
PTD	44.77 ± 0.49	41.39 ± 0.68	36.77 ± 1.97	30.59 ± 1.37	26.87 ± 2.26
BLTM	44.65 ± 1.13	42.09 ± 0.61	37.85 ± 1.14	33.23 ± 0.61	27.73 ± 1.07
MEIDTM	45.21 ± 0.99	41.14 ± 0.51	34.55 ± 1.78	29.58 ± 1.99	26.25 ± 2.06
CRGR	**51.03 ± 0.67**	**48.94 ± 0.27**	**44.91 ± 2.15**	**39.76 ± 1.68**	**32.04 ± 2.25**

manifold-regularized technique to facilitate the estimation of the IDN transition matrix and finally construct the consistent classifier.

5.2 Experimental Results

In this subsection, We conclude the experimental results from the following three aspects:

(1) How are the results on synthetic noisy datasets? Table 1, 2 and 3 demonstrate the classification accuracy on synthetic noisy datasets *CIFAR-10*, *SVHN*, and *CIFAR-100* respectively, where the best classification results are emphasized in bold. These results indicate that CRGR outperforms current state-of-the-art methods in tackling IDN. To be specific, compared with the previous approaches, CRGR outperforms the former at most by a margin of 4.37%, 7.17%, and 2.16% on the datasets *CIFAR-10*, *SVHN*, and *CIFAR-100* respectively. At the same time, with the increase in noise rate, CRGR gradually reveals its superiority, CRGR outperforms the second-best method by an average margin of 1.21% and 4.19% under IDN-10% and IDN-50%, exhibiting its robustness under extreme label noise.

(2) How are the results on the real-world dataset? To evaluate the performance of CRGR in real-world scenarios, we conduct experiments on the real-world noisy dataset *Clothing-1M*. The classification accuracy is exhibited in Table 4. The results show that CRGR surpasses all the comparison methods and achieves the best classification accuracy, exhibiting its effectiveness and superiority in reality.

(3) How well are the inter-class imbalance and inner-class selection bias problems solved? We conduct experiments on the synthetic *CIFAR-10* dataset with a 50% noise rate to evaluate CRGR's performance on different classes and instances. As shown in Fig.4, compared with the base method, CRGR improves the classification

Table 4: Classification accuracy on the *Clothing-1M* dataset.

Methods	CE	Co-teaching+	JoCoR	CCR	T-Revision	VolMinNet	PTD	BLTM	MEIDTM	CRGR
Accuracy(%)	68.64	66.37	70.39	71.85	70.82	72.38	71.53	73.09	73.57	**74.29**

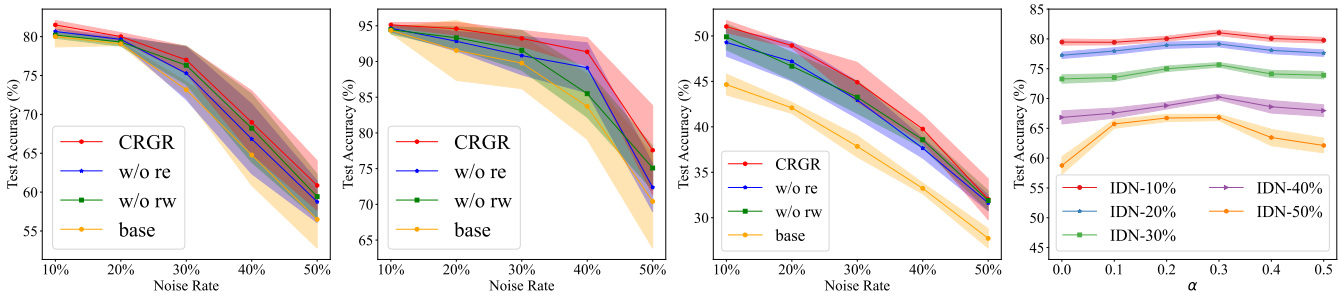


Figure 3: Illustration of the ablation study of each component and the hyper-parameter sensitivity. Figure (a)-(c) demonstrate the ablation studies on synthetic noisy datasets *CIFAR-10*, *SVHN*, and *CIFAR-100* with different noise rates respectively. Figure (d) illustrates the classification accuracy with various values of hyper-parameter α on *CIFAR-10* noisy dataset with different noise rates.

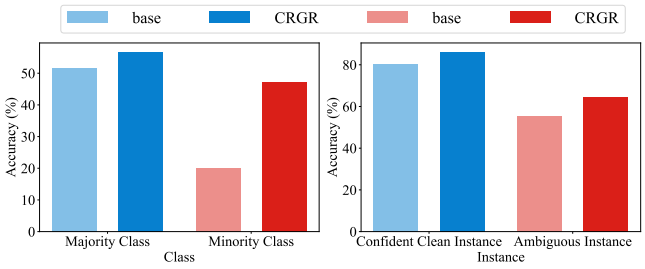


Figure 4: An illustration of classification accuracy on different classes and instances, where the base method represents the method that trains transition network on biased extracted instances with cross-entropy loss, majority and minority classes denote the class with the most and least extracted instances respectively.

accuracy by 5.25% and 27.16% on the classes with the most and least extracted instances, respectively, greatly reducing the gap between them. Analogously, CRGR improves the classification accuracy on confident clean instances and ambiguous instances by 6.05% and 9.08%, respectively, narrowing the gap by 3.03%. These results indicate that CRGR effectively mitigates both the inter-class and inner-class fitting bias and leads to a more equitable and robust classifier.

5.3 Ablation Study

In this subsection, we further conduct ablation experiments to investigate the effect of reweighting technique and geometric regularization and hyper-parameter tuning.

Effect of reweighting technique and geometric regularization. In this paper, we utilize a smoothed noise-tolerant reweighting technique to calibrate the imbalanced inter-class distribution of extracted instances and incorporate the ambiguous instances into the training framework by adding a geometric regularization. In the subsection, ablation experiments are conducted to assess the individual contributions of the reweighting technique and geometric regularization in enhancing the performance of the classifier. Specifically, we use 'w/o rw' and 'w/o re' to indicate the approach that does not apply geometric regularization and reweighting methodology, respectively, and use the method that directly trains the transition network on the biased extracted instances without any improvements as the base method. We compare the classification accuracy of different methods on synthetic datasets with different noise rates, the corresponding results are shown in Fig 3a - 3c. From the results, we find that either removing the reweighting technique (w/o rw) or geometric regularization (w/o re) degrades the performance of the classifier. At the same time, the basic method (base), which removes both of the above improvements, exhibits the worst performance. All these observations collectively demonstrate the effectiveness of both the reweighting technique and geometric regularization in constructing a more robust classifier.

Hyper-parameter tuning. To investigate the influence of the hyper-parameter α on the proposed method. We conduct experiments on synthetic noisy dataset *CIFAR-10* with various values of α under the noise rates from 10% to 50%. The corresponding results are shown in Fig 3d. The findings indicate that the classification accuracy is not sensitive to α when the noise rate is low, but it is sensitive when the noise rate is high. At the same time, with the increase of α, the classification accuracy first rises and then

descends and achieves the best value when the hyper-parameter is set as 0.3. Therefore, we set $\alpha = 0.3$ in all the experiments.

The selection of hyperparameter ρ_{max} We use the distillation method in [6] to extract the confident clean instances. According to [6], once ρ_{max} is selected as the upper bound of noise rate, the distilled confident clean instances' inferred latent clean labels are identical to the labels assigned for them by the Bayes optimal classifier. Hence, the clean labels can be accurately identified. However, when the noise rate is high, the distilled number of confident clean instances drops. Therefore, we should trade-off between the distilled number and distillation accuracy. [6] has provided enough evidence that $\rho_{max} = 0.3$ is reasonable for distillation. To be specific, they set the hyperparameter ρ_{max} as 0.1, 0.2, 0.3, 0.4, 0.5, and 0.6, respectively, and calculate the instance-dependent transition matrix approximation error under different thresholds on CIFAR-10 dataset with IDN-30% noise rate. The experimental results are shown in Table 5. When the threshold ρ_{max} is set as 0.3, the error achieves the minimum value. Therefore, we set $\rho_{max} = 0.3$ in all the experiments.

Table 5: Matrix approximation error under different ρ_{max}

ρ_{max}	0.1	0.2	0.3	0.4	0.5	0.6
Error	0.71	0.68	0.65	0.66	0.66	0.67

6 CONCLUSION AND LIMITATION

Conclusion. In this paper, we propose a novel label noise learning framework CRGR, which solves both the inter-class imbalance and inner-class selection bias two primary problems existing in confident clean instances and finally leads to a more equitable and robust classifier. Specifically, we first propose a novel smoothed noise-tolerant reweighting technique that equalizes the network's exposure to noise patterns in each class to calibrate the imbalanced inter-class distribution. Secondly, following the evidence that similar instances tend to have comparable noise patterns, we posit that $T(\boldsymbol{x})$ should mirror the similarity of the feature space. This insight leads to the inclusion of ambiguous instances in training, serving as geometric regularization. Such regularization helps the model overcome the inner-class selection bias and understand complex noise patterns. Extensive experiments on both synthetic and real-world datasets demonstrate that CRGR outperforms the current state-of-the-art methods, exhibiting its superiority.

Limitation. One major limitation of our study is that we use the Euclidean distance to measure the similarity relations between instances in both feature and transition matrix spaces. However, Euclidean distance may not be a good indicator as the high-dimension space may actually be embedded in a manifold. In the future, we will use the geodesic distance to measure the similarity of instances to effectively grasp their relationship and learn a more robust classifier.

ACKNOWLEDGMENTS

This research was partially supported by the Key Research and Development Project in Shaanxi Province No. 2022GXLH-01-03, the National Science Foundation of China under Grant Nos. 62002282, 62250009, and 6219278, and the Major Technological Innovation Project of Hangzhou No. 2022AIZD0113.

REFERENCES

[1] Yingbin Bai, Erkun Yang, Bo Han, Yanhua Yang, Jiatong Li, Yinian Mao, Gang Niu, and Tongliang Liu. 2021. Understanding and improving early stopping for learning with noisy labels. *Advances in Neural Information Processing Systems* 34 (2021), 24392–24403.

[2] Antonin Berthon, Bo Han, Gang Niu, Tongliang Liu, and Masashi Sugiyama. 2021. Confidence scores make instance-dependent label-noise learning possible. In *International conference on machine learning*. PMLR, 825–836.

[3] Avrim Blum, Adam Kalai, and Hal Wasserman. 2003. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM (JACM)* 50, 4 (2003), 506–519.

[4] Léon Bottou. 2010. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France, August 22-27, 2010 Keynote, Invited and Contributed Papers*. Springer, 177–186.

[5] Pengfei Chen, Junjie Ye, Guangyong Chen, Jingwei Zhao, and Pheng-Ann Heng. 2021. Beyond class-conditional assumption: A primary attempt to combat instance-dependent label noise. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 11442–11450.

[6] De Cheng, Tongliang Liu, Yixiong Ning, Nannan Wang, Bo Han, Gang Niu, Xinbo Gao, and Masashi Sugiyama. 2022. Instance-dependent label-noise learning with manifold-regularized transition matrix estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16630–16639.

[7] De Cheng, Yixiong Ning, Nannan Wang, Xinbo Gao, Heng Yang, Yuxuan Du, Bo Han, and Tongliang Liu. 2022. Class-Dependent Label-Noise Learning with Cycle-Consistency Regularization. *Advances in Neural Information Processing Systems* 35 (2022), 11104–11116.

[8] Hao Cheng, Zhaowei Zhu, Xingyu Li, Yifei Gong, Xing Sun, and Yang Liu. 2021. Learning with Instance-Dependent Label Noise: A Sample Sieve Approach. In *International Conference on Learning Representations*.

[9] Jiacheng Cheng, Tongliang Liu, Kotagiri Ramamohanarao, and Dacheng Tao. 2020. Learning with bounded instance and label-dependent label noise. In *International conference on machine learning*. PMLR, 1789–1799.

[10] Uri Cohen, SueYeon Chung, Daniel D Lee, and Haim Sompolinsky. 2020. Separability and geometry of object manifolds in deep neural networks. *Nature communications* 11, 1 (2020), 746.

[11] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. 2019. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9268–9277.

[12] Tausif Diwan, G Anirudh, and Jitendra V Tembhurne. 2023. Object detection using YOLO: Challenges, architectural successors, datasets and applications. *multimedia Tools and Applications* 82, 6 (2023), 9243–9275.

[13] Charles Elkan. 2001. The foundations of cost-sensitive learning. In *International joint conference on artificial intelligence*, Vol. 17. Lawrence Erlbaum Associates Ltd, 973–978.

[14] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in neural information processing systems* 31 (2018).

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[16] Danula Hettiachchi, Vassilis Kostakos, and Jorge Goncalves. 2022. A survey on task assignment in crowdsourcing. *ACM Computing Surveys (CSUR)* 55, 3 (2022), 1–35.

[17] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. 2016. Learning deep representation for imbalanced classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5375–5384.

[18] Huaxi Huang, Hui Kang, Sheng Liu, Olivier Salvado, Thierry Rakotoarivelo, Dadong Wang, and Tongliang Liu. 2023. Paddles: Phase-amplitude spectrum disentangled early stopping for learning with noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 16719–16730.

[19] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. 2018. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International conference on machine learning*. PMLR, 2304–2313.

[20] Nazmul Karim, Mamshad Nayeem Rizve, Nazanin Rahnavard, Ajmal Mian, and Mubarak Shah. 2022. Unicon: Combating label noise through uniform selection and contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9676–9686.

[21] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[22] Jan Kremer, Fei Sha, and Christian Igel. 2018. Robust active label correction. In *International conference on artificial intelligence and statistics*. PMLR, 308–316.
[23] Alex Krizhevsky and Geoffrey Hinton. 2009. *Learning multiple layers of features from tiny images*. Technical Report. Citeseer.
[24] Mingchen Li, Mahdi Soltanolkotabi, and Samet Oymak. 2020. Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In *International conference on artificial intelligence and statistics*. PMLR, 4313–4324.
[25] Shutao Li, Weiwei Song, Leyuan Fang, Yushi Chen, Pedram Ghamisi, and Jon Atli Benediktsson. 2019. Deep learning for hyperspectral image classification: An overview. *IEEE Transactions on Geoscience and Remote Sensing* 57, 9 (2019), 6690–6709.
[26] Xuefeng Li, Tongliang Liu, Bo Han, Gang Niu, and Masashi Sugiyama. 2021. Provably end-to-end label-noise learning without anchor points. In *International conference on machine learning*. PMLR, 6403–6413.
[27] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. 2020. Deep learning for generic object detection: A survey. *International journal of computer vision* 128 (2020), 261–318.
[28] Stephen E Palmer. 1977. Hierarchical structure in perceptual representation. *Cognitive psychology* 9, 4 (1977), 441–474.
[29] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).
[30] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. 2017. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1944–1952.
[31] Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. 2018. Joint optimization framework for learning with noisy labels. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5552–5560.
[32] Arash Vahdat. 2017. Toward robustness against label noise in training deep discriminative neural networks. *Advances in neural information processing systems* 30 (2017).
[33] Kai Wang, Xiangyu Peng, Shuo Yang, Jianfei Yang, Zheng Zhu, Xinchao Wang, and Yang You. 2022. Reliable label correction is a good booster when learning with extremely noisy labels. *arXiv preprint arXiv:2205.00186* (2022).
[34] Wei Wang, Yujing Yang, Xin Wang, Weizheng Wang, and Ji Li. 2019. Development of convolutional neural network and its application in image classification: a survey. *Optical Engineering* 58, 4 (2019), 040901–040901.
[35] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. 2017. Learning to model the tail. *Advances in neural information processing systems* 30 (2017).
[36] Hongxin Wei, Lei Feng, Xiangyu Chen, and Bo An. 2020. 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*. 13726–13735.
[37] Qi Wei, Haoliang Sun, Xiankai Lu, and Yilong Yin. 2022. Self-filtering: A noise-aware sample selection for label noise with confidence penalization. In *European Conference on Computer Vision*. Springer, 516–532.
[38] Xiaobo Xia, Tongliang Liu, Bo Han, Chen Gong, Nannan Wang, Zongyuan Ge, and Yi Chang. 2020. Robust early-learning: Hindering the memorization of noisy labels. In *International conference on learning representations*.
[39] Xiaobo Xia, Tongliang Liu, Bo Han, Nannan Wang, Mingming Gong, Haifeng Liu, Gang Niu, Dacheng Tao, and Masashi Sugiyama. 2020. Part-dependent label noise: Towards instance-dependent label noise. *Advances in Neural Information Processing Systems* 33 (2020), 7597–7610.
[40] Xiaobo Xia, Tongliang Liu, Nannan Wang, Bo Han, Chen Gong, Gang Niu, and Masashi Sugiyama. 2019. Are anchor points really indispensable in label-noise learning? *Advances in neural information processing systems* 32 (2019).
[41] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. 2015. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2691–2699.
[42] Yan Yan, Rómer Rosales, Glenn Fung, Ramanathan Subramanian, and Jennifer Dy. 2014. Learning from multiple annotators with varying expertise. *Machine learning* 95 (2014), 291–327.
[43] Shuo Yang, Erkun Yang, Bo Han, Yang Liu, Min Xu, Gang Niu, and Tongliang Liu. 2022. Estimating instance-dependent bayes-label transition matrix using a deep neural network. In *International Conference on Machine Learning*. PMLR, 25302–25312.
[44] Yu Yao, Tongliang Liu, Bo Han, Mingming Gong, Jiankang Deng, Gang Niu, and Masashi Sugiyama. 2020. Dual t: Reducing estimation error for transition matrix in label-noise learning. *Advances in neural information processing systems* 33 (2020), 7260–7271.
[45] Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor Tsang, and Masashi Sugiyama. 2019. How does disagreement help generalization against label corruption?. In *International Conference on Machine Learning*. PMLR, 7164–7173.
[46] Netzer Yuval. 2011. Reading digits in natural images with unsupervised feature learning. In *Proceedings of the NIPS Workshop on Deep Learning and Unsupervised Feature Learning*.
[47] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. 2019. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems* 30, 11 (2019), 3212–3232.

A APPENDIX

A.1 Details of Datasets

- *SVHN* [46]: The SVHN (Street View House Numbers) dataset consists of a large collection of color digit images extracted from Google Street View images. SVHN has 10 classes of images in total with 73,257 training instances and 26,032 test instances of varying sizes.
- *CIFAR-10 and CIFAR-100* [23]: Both the CIFAR-10 and CIFAR-100 datasets are created by the Canadian Institute for Advanced Research, where the images were collected from various sources and cover a wide range of object categories. CIFAR-10 and CIFAR-100 datasets contain 60,000 RGB images in total, which are divided into 50,000 training images and 10,000 test images. Each image has a resolution of 32×32 pixels. CIFAR-10 consists of 10 object classes, including airplanes, automobiles, birds, cats, deer, dogs, frogs, horses, ships, and trucks. On the other hand, CIFAR-100 contains 100 object classes, covering a wider range of fine-grained categories, such as different breeds of dogs, flowers, and household objects.
- *Clothing-1M* [41]: Clothing-1M is a large-scale RGB image dataset, it has 1M training images with noisy labels and 10k images with clean labels for testing. Clothing-1M has 14 classes in total and all the training images are collected from online shopping websites. Without expert annotation, the labels of training images are assigned based on their surrounding environment automatically, leading to label noise. The clean images are manually annotated to ensure their labels are clean for testing the classifier's performance.

A.2 Experimental Results of Ablation Study

In the text, we have conducted ablation experiments to illustrate the contributions of both the reweighting technique and geometric regularization in improving the classifier's performance. However, due to the space limit, we only provide the illustration by exploiting the figures. In this supplementary material, more detailed results including means and standard deviations of classification accuracy about the ablation experiments on CIFAR-10, SVHN, and CIFAR-100 are shown in Table 6, 7 and 8 respectively. To be specific, we use 'w/o re' and 'w/o rw' to indicate the approach that does not apply geometric regularization and reweighting methodology, respectively, and use the method that directly trains the transition network on the biased extracted instances without any improvements as the base method. From the results, we find that either removing the reweighting technique (w/o rw) or geometric regularization (w/o re) degrades the performance of the classifier. At the same time, the base method, which removes both of the above improvements, exhibits the worst performance. All these observations collectively demonstrate the effectiveness of both the reweighting technique and geometric regularization in constructing a more robust classifier.

Table 6: Means and standard deviations (percentage) of classification accuracy on CIFAR-10 with IDN levels.

	IDN-10%	IDN-20%	IDN-30%	IDN-40%	IDN-50%
base	80.01 ± 1.34	79.08 ± 0.23	73.17 ± 1.02	64.78 ± 3.92	56.49 ± 3.69
w/o re	80.66 ± 0.31	79.65 ± 0.37	75.28 ± 3.38	66.84 ± 4.49	58.76 ± 2.67
w/o rw	80.21 ± 0.42	79.31 ± 0.53	76.32 ± 2.47	68.22 ± 4.39	59.42 ± 2.89
CRGR	**81.51 ± 0.57**	**80.01 ± 0.51**	**77.01 ± 1.75**	**68.28 ± 4.06**	**60.86 ± 3.18**

Table 7: Means and standard deviations (percentage) of classification accuracy on SVHN with IDN levels.

	IDN-10%	IDN-20%	IDN-30%	IDN-40%	IDN-50%
base	94.35 ± 0.42	91.55 ± 4.19	89.78 ± 3.58	83.71 ± 4.60	70.41 ± 6.53
w/o re	94.65 ± 0.51	92.83 ± 1.37	90.82 ± 2.69	89.12 ± 3.50	72.37 ± 3.37
w/o rw	94.45 ± 0.62	93.32 ± 1.53	91.57 ± 2.78	85.49 ± 3.29	75.09 ± 3.09
CRGR	**95.12 ± 0.33**	**94.59 ± 0.89**	**93.23 ± 1.13**	**91.35 ± 1.99**	**77.58 ± 6.25**

Table 8: Means and standard deviations (percentage) of classification accuracy on CIFAR-100 with IDN levels.

	IDN-10%	IDN-20%	IDN-30%	IDN-40%	IDN-50%
base	44.65 ± 1.14	42.09 ± 0.61	37.85 ± 1.18	33.23 ± 0.60	27.73 ± 1.07
w/o re	49.28 ± 1.51	47.19 ± 2.16	42.95 ± 2.02	37.65 ± 1.09	31.63 ± 0.87
w/o rw	49.90 ± 1.42	46.67 ± 1.53	43.26 ± 1.78	38.58 ± 0.89	31.89 ± 1.09
CRGR	**51.03 ± 0.67**	**48.94 ± 0.27**	**44.91 ± 2.15**	**39.76 ± 1.54**	**32.04 ± 2.25**