# COMBATING NOISY LABELS WITH STOCHASTIC NOISE-TOLERATED SUPERVISED CONTRASTIVE LEARNING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Learning with noisy labels (LNL) aims to achieve good generalization performance given a label-corrupted training set. In this work, we consider a more challenging situation of LNL on *fine-grained* datasets (LNL-FG). Due to large inter-class ambiguity among those fine-grained classes, deep models are more prone to overfitting to noisy labels, leading to poor generalization performance. To handle this problem, we propose a novel framework called stochastic noise-tolerated supervised contrastive learning (SNSCL) that can enhance discriminability of deep models. Specifically, SNSCL contains a noise-tolerated contrastive loss and a stochastic module. To play against fitting noisy labels, we design a noise-tolerated supervised contrastive learning loss that incorporates a weight-aware mechanism for noisy label correction and selectively updating momentum queue lists. By this mechanism, SCL mitigates the effects of noisy anchors and avoids inserting noisy labels into the momentum-updated queue. Besides, to avoid manually-defined augmentation strategies in SCL, we propose an efficient stochastic module that samples feature embeddings from a generated distribution, which can also enhance the representation ability of SCL. Our proposed SNSCL is general and compatible with prevailing robust LNL strategies to improve their performance for LNL-FG. Extensive experiments on four noisy benchmarks and an open-world dataset with variant noise ratios demonstrate that our proposed framework significantly improves the performance of current LNL methods for LNL-FG.

## 1 INTRODUCTION

Learning from noisy labels (Long & Servedio, 2008; Bossard et al., 2014; Han et al., 2018; Xu et al., 2019; Li et al., 2020; Wei et al., 2022) poses great challenges for training deep models, whose performance heavily relies on large-scaled labeled datasets. Annotating training data with high confidence would be resource-intensive, especially for some domains, such as medical and remote sensing images. Thus, label noise would inevitably arise.

LNL on fine-grained classification tasks is challenging. *Random classification noise* is common in real applications, which has been broadly studied in (Long & Servedio, 2008; Tewari & Bartlett, 2007; Tanaka et al., 2018; Liu et al., 2020). *Dependent noise* is caused by uncertain annotation that catches attention in recent years (Han et al., 2018; Wei et al., 2020; Shu et al., 2019; Wei et al., 2020; Li et al., 2020; Wei et al., 2022). Existing methods usually simulate these two types of noise on generic image datasets to evaluate their algorithms. In this work, we extend LNL to *fine-grained* classification (LNL-FG). This scenario is more realistic since annotators are easier to be misguided by indistinguishable characteristics among fine-grained images and give an uncertain target. Fig. 1 illustrates comparison between two types of noise simulated on generic and fine-grained sets. The results in Fig. 2 show that deep models are earlier to overfit to noise for fine-grained classification tasks. Intuitively, due to large inter-class ambiguity in LNL-FG, the margin between noisy samples and the decision boundary in the fine-grained dataset is smaller than that in the generic dataset, leading to severe overfitting of deep models to noisy labels.

Contrastive learning (CL), as a powerful self-supervised approach for unsupervised representation learning (Chen et al., 2020a; He et al., 2020; Grill et al., 2020; Jaiswal et al., 2020; Park et al., 2020), has attracted the attention of LNL (Li et al., 2022). CL methods usually design objective functions as supervised learning to perform pretext similarity measurement tasks derived from an

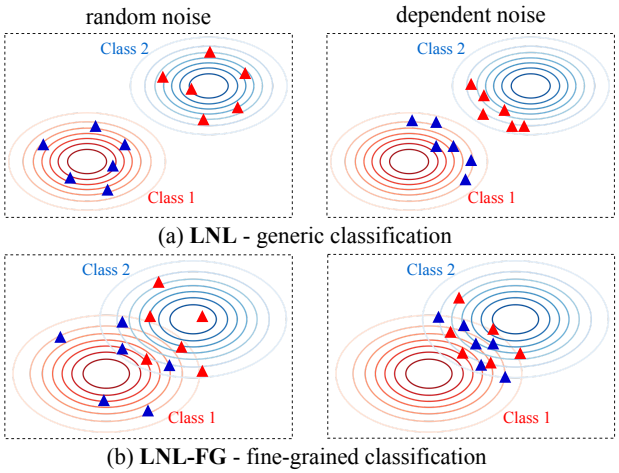



Figure 1: LNL-FG is more challenging than LNL on generic classification. ▲ and ▲ denote mislabeled samples.

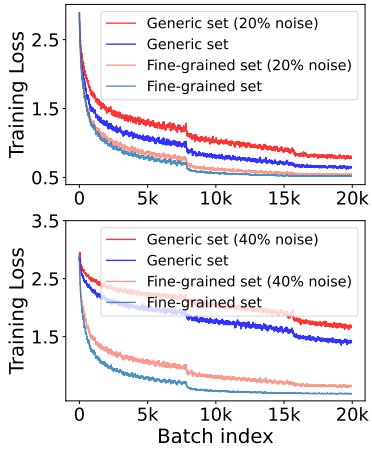



Figure 2: Overfitting is severe on fine-grained datasets for vanilla CE loss.

unlabeled dataset, which can learn effective visual representations in downstream tasks, especially for fine-grained classification. The following work, supervised contrastive learning (Khosla et al., 2020), leverages label information to further enhance representation learning, which can avoid a vast training batch and reduce the memory cost. Since the goal of LNL is eventually learning the discrimitive feature embedding, enhancing representation ability of deep models via SCL can play against overfitting to noisy labels. However, SCL cannot be directly applied to the noisy scenario as it is lack of noise-tolerated mechanism.

To resolve the noise-sensitivity of SCL, we propose a novel framework named stochastic noise-tolerated supervised contrastive learning (SNSCL), which contains a noise-tolerated contrastive loss and a stochastic module. For the noise-tolerated contrastive loss, we roughly categorize the noise-sensitive property of SCL into two parts of noisy anchors and noisy query keys in the momentum queue. To mitigate the negative effect introduced by noisy anchors or query keys, we design a weight mechanism for measuring the reliability score of each sample and modify the label of noisy anchors in current training batch. Then, we selectively update the momentum queue for decreasing the probability of noisy query keys. These operations are adaptive and can achieve a progressive learning process. Besides, to avoid manual adjustment of strong augmentation strategies for SCL, we propose a stochastic module for more complex feature transformation. In practice, this module generates the probabilistic distribution of feature embedding, which can achieve better generalization performance for LNL-FG.

Our contributions can be summarized as

- We design a novel framework dubbed stochastic noise-tolerated supervised contrastive learning (SNSCL), which alters the noisy labels for anchor samples and selectively updates the momentum queue, avoiding the effects of noisy labels on SCL.
- We design a stochastic module to avoid manually-defined augmentation, improving the performance of SNSCL on representation learning.
- Our proposed SNSCL is generally applicable to prevailing LNL methods and further improves their performance on LNL-FG.

Our method achieves state-of-the-art performance on four fine-grained datasets, demonstrating great effectiveness of noise-tolerated robust learning.

## 2 PRELIMINARIES

**Problem definition**. Given a training set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n} \in (\mathcal{X}, \mathcal{Y})$ with partial corrupted labels, where $y_i \in \{1, 2, \cdots, C\}$. Supposing there is a deep neural network with the learnable parameters $\theta$. The goal of our algorithm is finding the optimal parameter $\theta^*$ which can achieve admirable generalization performance on the clean testing set.

**Contrastive learning meets noisy labels**. *Contrastive learning* (Chen et al., 2020a; He et al., 2020; Grill et al., 2020) is a prevailing framework for representation learning, enhancing class discrimination of the feature extractor. Supposing a feature anchor $\mathrm{q}$ and a set of feature keys $\{\hat{\mathrm{q}}, \mathrm{k}_1, \cdots, \mathrm{k}_\mathrm{D}\}$ are given, where $\hat{\mathrm{q}}$ is a positive data point for $\mathrm{q}$, and the others are negative. In CL, a widely used loss function for measuring the similarity of each data point is InfoNCE (Oord et al., 2018) and can be summarised as

$$\mathcal{L}_{\mathrm{INFO}} = -\log \frac{\exp\left(\mathrm{q} \cdot \hat{\mathrm{q}}/\tau\right)}{\exp\left(\mathrm{q} \cdot \hat{\mathrm{q}}/\tau\right) + \sum_{d=1}^{\mathrm{D}} \exp\left(\mathrm{q} \cdot \mathrm{k}_d/\tau\right)}, \tag{1}$$

where $\tau$ is a hyper-parameter for temperature scaling. In most applications, CL is built as a pre-task. $\mathrm{q}$ and $\hat{\mathrm{q}}$ are extracted from two augmented views of the same example, and negative keys $\{\mathrm{k}_1, \cdots, \mathrm{k}_\mathrm{D}\}$ represent feature embeddings of other samples in the current training batch. CL is naturally independent of noisy labels, but there exists a drawback in that it lacks a mechanism to utilize potential labels into model training, leaving useful discriminative information on the shelf (Wang et al., 2021a). Currently, *supervised contrastive learning* (Khosla et al., 2020) solves this issue by constructing the positive and the negative lists according to the labels. For anchor point $\mathrm{q}$, the objective function can be written as

$$\mathcal{L}_{\mathrm{SCL}} = -\log \frac{\sum_{\mathrm{k_P} \in \mathrm{Pos}} \exp\left(\mathrm{q} \cdot \mathrm{k_P}/\tau\right)}{\sum_{\mathrm{k_P} \in \mathrm{Pos}} \exp\left(\mathrm{q} \cdot \mathrm{k_P}/\tau\right) + \sum_{\mathrm{k_N} \in \mathrm{Neg}} \exp\left(\mathrm{q} \cdot \mathrm{k_N}/\tau\right)}, \tag{2}$$

where $\mathrm{Pos}$ and $\mathrm{Neg}$ represent the positive and negative list, respectively.

However, SCL is sensitive to noisy labels, which can be introduced into the anchor point, $\mathrm{Pos}$, and $\mathrm{Neg}$. Our goal is to utilize the valuable information of the labels underlying the noisy training set $\mathcal{D}$ and overcome the misguidance of noisy labels.

## 3 Proposed Method

**Overview**. In this section, we first introduce a noise-tolerated supervised contrastive learning method that incorporates a weight-aware mechanism for measuring the reliability score of each example. Based on this mechanism, we dynamically alter the unreliable labels and selectively insert them into the momentum-updated queue, combating two noise-sensitive issues of SCL, respectively. Then, we design a stochastic module for the transformation of feature embeddings, which samples from a generated probabilistic distribution.

### 3.1 Noise-tolerated supervised contrastive learning

**Weight-aware mechanism**. We aim to measure the reliability score of each sample in the training set $\mathcal{D}$ and generate the corresponding weight. For this, we use the *small-loss* criterion, a common strategy in LNL, and leverage a two-component GMM to generate this reliability score. We evaluate the training set $\mathcal{D}$ after each training epoch. For clarity, we omit the epoch sequence and attain a list of empirical losses $\{l_i\}_{i=0}^{n}$ among all samples, where $l_i = L(F(x_i; \theta), y_i)$. Note that $F(\cdot)$ denotes the classifier network and $L(\cdot)$ is the cross-entropy loss. GMM fits to this list and gives the reliability score of the probability that the sample is clean. For sample $x_i$, the reliability score $\gamma_i$ can be written as $\gamma_i = \mathrm{GMM}(l_i \mid \{l_i\}_{i=0}^{n})$, where $\gamma_i \in [0, 1]$. Then, we design a function to dynamically adjust the weight for all training samples according to the reliability score. The weight of sample $x_i$ is

$$\omega_i = \begin{cases} 1 & \text{if} \quad \gamma_i > t \\ \gamma_i & \text{otherwise} \end{cases}, \tag{3}$$

where $t$ is a hyper-parameter in the interval of $[0, 1]$ and denotes the threshold of the reliability score. The computation of $\gamma$ and $\omega$ restarts after each training round, ensuring that the values benefit from the improvement of the model performance.

Based on this mechanism, we design two strategies that modify two noise-sensitive issues summarised in the overview. First, to solve the misguidance of the noisy anchor sample, we propose a **weighted correction strategy** to alter the labels of unreliable samples. For the unreliable sample $x \in \{(x_i, y_i) | \omega_i \neq 1\}_{i=1}^{n}$, the weighted label $\hat{y}$ is written as

$$\hat{y} = \omega y^{cls} + (1 - \omega) y, \tag{4}$$

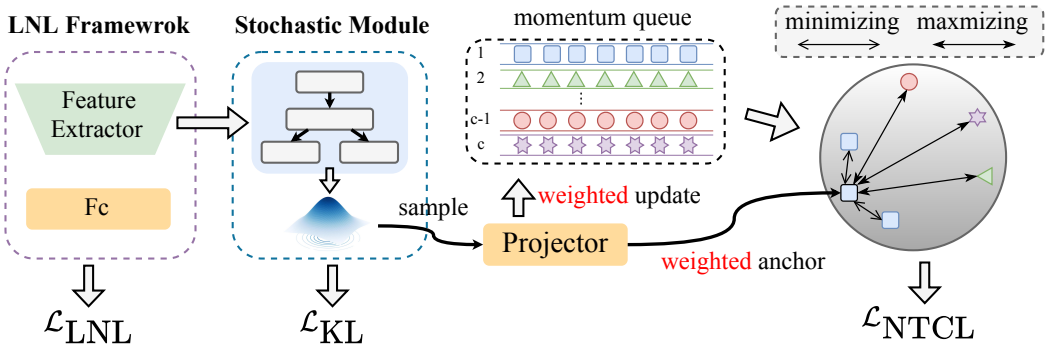



Figure 3: **Illustration of training framework**. Examples in the momentum queue with the same color and shape belong to the same category. The *Projector* is set as a single-layer MLP structure. Overall, the total training framework includes a LNL method and our proposed SNSCL, which consists of two parts: 1) **stochastic module**, which provides more competitive feature transformation; 2) **noise-tolerated contrastive loss**, which is noise-aware and contains two weighting strategies.

where $y^{cls} = \arg\max(\text{Softmax}(F(x;\theta)))$ and represents the prediction result of the classifier network. Additionally, to make the alteration of labels more stable, we use the idea of moving-average. At epoch $e$, the moving-average corrected label over multiple training epochs is

$$\hat{y}^e = \alpha \hat{y}^{(e-1)} + (1-\alpha)\hat{y}^e. \tag{5}$$

The coefficient is set as $\alpha = 0.99$. Therefore, the set of unreliable samples can be formulated as $\{(x_i, \hat{y}^e)|\omega_i \neq 1\}_{i=1}^n$ in epoch $e$.

Second, to solve the noise tolerance properties of the momentum queue, we propose a **weighted update strategy** to solve the noise-tolerant property of the previous momentum queue. This strategy can be simply summarized as updating this queue according to the weight in Eq. 3. Given a sample $x_i$, its weight value is $\omega_i$. For the sample $x$ which satisfies to $\omega_i = 1$, we update the queue by $x_i$ via the First-in First-out principle. Otherwise, we update the queue by $x_i$ with probability $\omega_i$. Intuitively, the weighted-update strategy avoids inserting unreliable samples into the queue, helping enhance the quality of the momentum queue.

### 3.2 Stochastic Feature Embedding

Typical CL heavily relies on sophisticated augmentation strategies and need specify them for different datasets. We build a stochastic module to avoid manually-defined strategies. Given a sample $x$, let $\mathrm{z} = f(x)$ represent the output of the backbone network (*i.e.*, feature extractor) and belongs to the embedding space $\mathbb{R}^D$. We formulate a probability distribution $p(Q|\mathrm{z})$ for embedding z as a normal distribution, which can be written as

$$p(Q|\mathrm{z}) \sim \mathcal{N}(\mu, \sigma^2), \tag{6}$$

where $\mu$ and $\sigma$ can be learned by a three-layers fully-connected network. From feature embedding distribution $p(Q|\mathrm{z})$, we sample an embedding $\mathrm{z}'$ to represent the augmented version of original feature embedding z. Here, we use reparameterization trick (Kingma et al., 2015),

$$\mathrm{z}' = \mu + \epsilon \cdot \sigma \quad \text{with} \quad \epsilon \sim \mathcal{N}(0, \mathbf{I}). \tag{7}$$

After that, the sampled feature embedding $\mathrm{z}'$ is utilized to update the momentum queue and compute contrastive learning loss. The merits of this module are 1) more complex representations stimulate the potential of CL, and 2) the property of stochasticity helps the model escape from memorizing the noisy signal to some degree. Experiments of module architecture can be found in Appx. A.3

### 3.3 Total Objective

Our proposal can be easily integrated with current LNL algorithms and improve their performance on LNL-FG. Therefore, the total training objective contains three following parts.

**Algorithm 1** The pseudocode of proposed SNSCL

**Require:** Training set $\mathcal{D}^N = \{(x_i, y_i)\}_{i=1}^n$, the momentum queue size D, a reliability threshold $t \in [0, 1]$, two trade-off coefficients $\lambda_1, \lambda_2$, an average-moving coefficient $\alpha$, the training batch size $B$.
**Require:** (Networks) Classifier network $F$ (feature extractor $f$), Stochastic module $\mathcal{M}$.
**Ensure:** Optimal parameters of classifier network $\theta^*$

WarmUp $(F(\theta); \mathcal{D}^N)$ // initialize the parameter $\theta$ via cross-entropy
**while** $e <$ MaxEpoch **do**
  $\gamma_i \leftarrow \text{GMM}(l_i \mid \{l_i = L(F(x_i; \theta), y_i)\}_{i=1}^n)$ // compute the cross-entropy loss and reliability score $\gamma$ for each sample
  $\omega_i \leftarrow (\gamma_i, t)$ // compute the weight value $\omega$ for each sample ▷ Eq. 3
  $\hat{y}_i^e \leftarrow \alpha \hat{y}_i^{(e-1)} + (1 - \alpha)\hat{y}_i^e$ // refurbish the unreliable labels with average-moving ▷ Eq. 4, 5
  **for** $iter \in \{1, ..., num\_iters\}$ **do**
    Draw a mini-batch $\{(x_l, y_l)\}_{l=1}^B$ from the label-corrected training set
    **for** $l \in \{1, ..., B\}$ **do**
      $\text{z}'_l \leftarrow p(Q|\text{z}_l)$, where $p(Q|\text{z}_l) \sim \mathcal{N}(\mu, \sigma^2)$ // generate a distribution via $\mathcal{M}$ and then sample ▷ Eq. 6, 7
      Weighted-update momentum queue according to $y_l$
      Compute three losses $\mathcal{L}_{\text{LNL}}$, $\mathcal{L}_{\text{KL}}$, $\mathcal{L}_{\text{NTCL}}$ ▷ Eq. 8, 9
    **end for**
    $\theta^{(e)} \leftarrow \text{SGD}(\frac{1}{B}\sum_{b=1}^B(\mathcal{L}_{\text{LNL}} + \lambda_1\mathcal{L}_{\text{ours}} + \lambda_2\mathcal{L}_{\text{KL}}); \theta^{(e)})$
  **end for**
**end while**
**return** $\theta^*$

**LNL loss**. There exists a LNL loss dubbed $\mathcal{L}_{\text{LNL}}$ for classifier learning. Note that the input set of the LNL framework can be written as $\{(x_i, \hat{y}^e)|\omega_i \neq 1\}_{i=1}^n$ and $\{(x_i, y_i)|\omega_i = 1\}_{i=1}^n$ at epoch $e$, which contains less noisy labels compared with the original training set $\mathcal{D}$.

**Noise-tolerated contrastive loss**. For each sample $x_i$ in the training set $\mathcal{D}$, we attain its weight $\omega_i$ by the strategy in section 3.1. If $\omega_i$ satisfies to $\omega_i = 1$, the label of sample $x_i$ keeps the original label $y_i$, otherwise it is replaced with the moving-averaged label $\hat{y}^e$ in Eq. 5. Then, we omit the subscripts and formulate this sample as $(x, \hat{\text{y}})$ where q denotes its feature embedding. In our weighted momentum queue, the positive keys $\{\text{k}_1^{\hat{\text{y}}}, \cdots, \text{k}_{\text{D}}^{\hat{\text{y}}}\}$ are found according to the label $\hat{\text{y}}$. Complementarily, the remaining key points in the momentum queue are regarded as negative keys with size $[\text{D} \times (C-1)]$. Note that the size of the total momentum queue is $[\text{D} \times C]$. Formally, our noise-tolerated contrastive loss is summarized as

$$\begin{aligned} \mathcal{L}_{\text{NTCL}} &= -\frac{1}{\text{D}}\sum_{d=1}^{\text{D}} \log \frac{\exp(\text{q} \cdot \text{k}_d^{\hat{\text{y}}}/\tau)}{L_{\text{Pos}} + L_{\text{Neg}}} \quad \text{with} \\ L_{\text{Pos}} &= \sum_{j=1}^{\text{D}} \exp(\text{q} \cdot \text{k}_j^{\hat{\text{y}}}/\tau) \quad \text{and} \quad L_{\text{Neg}} = \sum_{c=1}^{\{1,\cdots,C\}\setminus\hat{y}} \sum_{j=1}^{\text{D}} \exp(\text{q} \cdot \text{k}_j^{c}/\tau), \end{aligned} \tag{8}$$

where the $L_{\text{Pos}}$ denotes positive keys from the same class $\hat{y}$ while $L_{\text{Neg}}$ denotes the negative keys from other classes $\{1, \cdots, C\}\setminus\hat{y}$.

**KL regularization**. We employ the KL regularization term between the feature embedding distribution $Q$ and unit Gaussian prior $\mathcal{N}(0, \mathbf{I})$ to prevent the predicted variance from collapsing to zero. The regularization can be formulated as

$$\mathcal{L}_{\text{KL}} = \text{KL}[p(z|Q)||\mathcal{N}(0, \mathbf{I}))]. \tag{9}$$

The overall loss function can be formulated with two hyper-parameters $\lambda_1$ and $\lambda_2$ as

$$\mathcal{L} = \mathcal{L}_{\text{LNL}} + \lambda_1\mathcal{L}_{\text{NTCL}} + \lambda_2\mathcal{L}_{\text{KL}}. \tag{10}$$

The training flowchart is shown in Fig. 3. Our proposed weighting strategies can be easily integrated into the typical SCL method, deriving a general LNL framework. The main operation is summarized in Algorithm 1. Compared to typical SCL, the weighting strategies would not cause much extra computational cost.

Table 1: Comparisons with test accuracy on *symmetric* label noise. The average **best** and the **last** accuracy among three times are reported. ↑ denotes the performance improvement of *SNSCL*.

| | Stanford Dogs | | Standford Cars | | Aircraft | | CUB-200-2011 | |
|---|---|---|---|---|---|---|---|---|
| | 20% | 40% | 20% | 40% | 20% | 40% | 20% | 40% |
| Cross-Entropy | 73.01 (63.82) | 69.20 (50.45) | 65.74 (64.08) | 51.42 (45.62) | 56.51 (54.67) | 45.67 (38.89) | 64.01 (60.77) | 54.14 (45.85) |
| + SNSCL | 76.33 (75.83) | 75.27 (75.00) | 83.24 (82.99) | 76.72 (76.36) | 76.45 (76.45) | 70.48 (69.64) | 73.32 (72.99) | 68.83 (68.67) |
| Label Smooth | 73.51 (64.42) | 70.22 (50.97) | 65.45 (64.24) | 51.57 (45.19) | 58.21 (54.73) | 45.24 (38.01) | 64.76 (60.60) | 54.39 (45.28) |
| + SNSCL | 76.85 (76.12) | 74.64 (74.60) | 83.21 (83.01) | 76.07 (75.90) | 76.24 (75.70) | 70.36 (70.06) | 73.46 (73.09) | 69.14 (68.64) |
| Conf. Penalty | 73.22 (66.89) | 68.69 (52.98) | 64.74 (64.46) | 48.15 (43.71) | 56.32 (55.51) | 43.64 (39.54) | 62.75 (61.10) | 52.04 (45.13) |
| + SNSCL | 76.14 (75.73) | 74.72 (74.49) | 83.07 (83.00) | 75.67 (75.38) | 75.04 (74.23) | 67.99 (66.85) | 73.90 (73.51) | 68.42 (67.86) |
| GCE | 66.96 (66.93) | 61.47 (60.32) | 62.77 (61.23) | 47.44 (46.13) | 39.54 (39.24) | 32.34 (32.28) | 58.74 (57.20) | 49.71 (48.11) |
| + SNSCL | 75.99 (74.56) | 71.68 (70.62) | 73.78 (73.55) | 58.11 (57.41) | 72.67 (71.53) | 60.19 (59.83) | 70.83 (70.56) | 61.67 (61.46) |
| SYM | 69.20 (62.13) | 65.76 (46.99) | 74.65 (73.21) | 52.83 (51.61) | 62.29 (60.51) | 54.36 (45.39) | 65.34 (63.60) | 50.19 (50.15) |
| + SNSCL | 77.55 (77.24) | 76.28 (76.25) | 84.59 (83.54) | 79.07 (78.87) | 79.64 (79.09) | 74.02 (73.63) | 76.67 (76.06) | 72.71 (72.58) |
| Co-teaching | 63.71 (58.43) | 49.15 (48.92) | 68.60 (67.95) | 56.92 (55.95) | 42.55 (40.62) | 35.21 (32.16) | 57.84 (55.98) | 46.57 (46.22) |
| + SNSCL | 74.18 (73.09) | 60.71 (58.84) | 78.94 (78.13) | 75.98 (75.06) | 74.61 (74.19) | 65.47 (63.81) | 69.77 (69.34) | 60.59 (58.94) |
| JoCoR | 66.94 (60.81) | 49.62 (48.62) | 69.99 (68.25) | 57.95 (56.71) | 61.37 (59.16) | 52.11 (49.93) | 58.79 (57.74) | 52.64 (49.35) |
| + SNSCL | 75.79 (74.99) | 63.42 (62.84) | 79.67 (78.77) | 76.80 (76.21) | 75.88 (75.16) | 71.65 (70.67) | 71.86 (70.90) | 64.43 (63.81) |
| MW-Net | 71.99 (69.20) | 68.14 (65.17) | 74.01 (73.88) | 58.30 (55.81) | 64.97 (61.84) | 57.61 (55.90) | 67.44 (65.20) | 58.49 (54.81) |
| + SNSCL | 77.49 (77.08) | 74.92 (74.38) | 85.96 (85.37) | 77.76 (77.13) | 80.08 (78.94) | 73.55 (73.18) | 76.94 (76.24) | 69.51 (68.83) |
| MLC | 74.08 (70.51) | 69.44 (66.28) | 76.02 (71.24) | 59.44 (55.76) | 63.81 (60.33) | 58.11 (54.86) | 69.44 (68.19) | 60.27 (58.49) |
| + SNSCL | 78.92 (78.56) | 76.49 (78.96) | 85.92 (84.91) | 78.49 (77.80) | 79.19 (78.40) | 75.21 (74.67) | 77.58 (76.68) | 71.54 (70.86) |
| DivideMix | 79.22 (77.86) | 77.93 (76.28) | 78.35 (77.99) | 62.54 (62.50) | 80.62 (80.50) | 66.76 (66.13) | 75.11 (74.54) | 67.35 (66.96) |
| + SNSCL | 81.40 (81.16) | 79.12 (78.91) | 86.29 (85.94) | 80.09 (79.51) | 82.31 (82.03) | 76.22 (75.67) | 78.36 (78.04) | 73.66 (73.28) |
| **Avg. ↑** | **5.88 (9.34)** | **7.76 (15.83)** | **12.44 (13.29)** | **20.82 (23.06)** | **18.60 (19.86)** | **21.41 (24.49)** | **9.87 (11.25)** | **12.22 (16.46)** |

# 4 EXPERIMENTS

To evaluate the performance of our proposed SNSCL, we conduct extensive experiments, including 1) **task variety**: we select four fine-grained visual datasets and an open-world noisy dataset; 2) **noise condition**: we construct two noisy types with varying noise ratios on four fine-grained datasets.

## 4.1 IMPLEMENTATION DETAILS

**Noisy test benchmarks**. We introduce four typical datasets in fine-grained classification tasks and manually construct noisy labels. By a noise transition matrix $\mathrm{T}$, we change partial labels of clean datasets. Given a noise ratio $r$, for a sample $(x, y)$, the transition from clean label $y = i$ to wrong label $y = j$ can be represented by $\mathrm{T}_{ij} = \mathrm{P}(y = j|y = i)$ and $\mathrm{P} = r$, where $r$ is the preset noise ratio. According to the structure of $\mathrm{T}$, the noisy labels can be divided into two types: 1) **Symmetric** (random) noise. The diagonal elements of $\mathrm{T}$ are $1 - r$ and the off-diagonal values are $r/(c - 1)$; 2) **Asymmetric** (dependent) noise. The diagonal elements of $\mathrm{T}$ are $1-r$, and there exists another value $r$ in each row. Noise ratio $r$ is set as $r \in \{10\%, ..., 40\%\}$. An illustration of the noise transition matrix $\mathrm{T}$ and basic statistics of these datasets are shown in Appx. A.2.

Besides, we also select a large-scale dataset collected from a clothing website to evaluate the effectiveness of our algorithm on real-world applications. Clothing-1M (Xiao et al., 2015) contains one million training images from 14 categories, with approximately 39.45% noisy labels.

**Training settings**. The code is implemented by Pytorch 1.9.0 with single GTX 3090. For four fine-grained noisy benchmarks, the optimizer is SGD with the momentum of 0.9, while initialized learning rate is 0.001 and the weight decay is 1e-3. The number of total training epochs is both 100, and the learning rate is decayed with the factor 10 by 20 and 40 epoch. For Clothing-1M, refers to (Wei et al., 2022), we train the classifier network for 15 epochs and use SGD with 0.9 momentum, weight decay of 5e-4. The learning rate is set as 0.002 and decayed with the factor of 10 after 10 epochs, while warm up stage is one epoch. For all experiments, we set the training batch size as 32. The data augmentation strategies include randomly cropping from 255×255 to 224×224, horizontally flipping. In addition, we adopt a default temperature $\tau = 0.07$ for scaling.

**Hyper-parameters**. Our framework includes two hyper-parameters, *i.e.*, the reliability threshold $t$ in Eq. 3 and the length of momentum queue $\mathrm{D}$ Eq. 8. For all experiments, we set $t = 0.5$ and $\mathrm{D} = 32$. In addition, the trade-off parameters in Eq. 10 are set as $\lambda_1 = 1, \lambda_2 = 0.001$.

## 4.2 COMPARISON WITH STATE-OF-THE-ARTS

**Baselines**. We evaluate the effectiveness of our method by adding the proposal into current LNL algorithm and compare the improvements on LNL-FG task. The basic methods we compared include

Table 2: Comparisons with test accuracy on *asymmetric* label noise. The average **best** and the **last** accuracy among three times are reported. ↑ denotes the performance improvement of *SNSCL*.

| | Stanford Dogs | | Standford Cars | | Aircraft | | CUB-200-2011 | |
|---|---|---|---|---|---|---|---|---|
| | 10% | 30% | 10% | 30% | 10% | 30% | 10% | 30% |
| Cross-Entropy | 74.24 (71.32) | 63.76 (56.86) | 74.58 (74.57) | 58.08 (57.43) | 65.98 (62.53) | 51.10 (47.85) | 68.26 (68.00) | 56.02 (54.13) |
| + SNSCL | 76.24 (74.88) | 64.49 (62.37) | 83.73 (83.41) | 70.04 (69.61) | 78.28 (78.22) | 65.44 (65.11) | 74.80 (74.47) | 61.48 (60.70) |
| Label Smooth | 74.70 (71.81) | 64.99 (57.04) | 74.28 (74.13) | 58.47 (57.80) | 65.29 (63.34) | 51.88 (47.71) | 68.78 (67.67) | 56.80 (53.69) |
| + SNSCL | 75.84 (75.16) | 65.23 (63.69) | 84.27 (84.13) | 70.49 (70.20) | 78.67 (77.98) | 66.28 (65.56) | 75.51 (75.42) | 62.05 (61.43) |
| Conf. Penalty | 74.41 (72.04) | 64.50 (57.92) | 73.78 (73.67) | 56.96 (56.53) | 64.90 (63.01) | 49.38 (47.53) | 67.66 (67.62) | 54.33 (52.80) |
| + SNSCL | 76.01 (75.62) | 67.53 (66.32) | 84.26 (83.91) | 72.23 (71.96) | 78.34 (78.01) | 66.88 (66.34) | 75.34 (74.97) | 62.69 (62.67) |
| GCE | 67.13 (66.83) | 54.53 (53.92) | 68.75 (68.71) | 60.57 (60.21) | 44.22 (44.16) | 34.18 (33.66) | 62.92 (60.77) | 50.05 (49.79) |
| + SNSCL | 75.91 (74.63) | 68.45 (67.13) | 80.33 (80.04) | 64.64 (64.38) | 73.85 (73.89) | 64.33 (63.91) | 73.77 (73.23) | 61.37 (60.96) |
| SYM | 69.57 (66.75) | 61.61 (51.11) | 76.74 (76.18) | 58.30 (57.42) | 69.31 (67.45) | 50.23 (47.55) | 68.81 (68.00) | 52.16 (51.83) |
| + SNSCL | 77.37 (76.64) | 74.74 (74.41) | 86.71 (86.54) | 78.98 (78.66) | 82.30 (81.46) | 69.61 (69.37) | 77.89 (77.27) | 67.43 (66.95) |
| Co-teaching | 59.95 (59.77) | 50.50 (50.44) | 72.88 (72.71) | 61.02 (60.86) | 55.94 (49.85) | 45.18 (38.97) | 61.00 (60.92) | 50.06 (48.55) |
| + SNSCL | 70.46 (70.24) | 65.83 (65.41) | 82.17 (81.63) | 66.84 (66.49) | 74.73 (74.28) | 62.17 (61.88) | 70.92 (70.63) | 64.55 (64.10) |
| JoCoR | 61.34 (60.11) | 53.39 (52.35) | 74.68 (73.21) | 63.54 (62.27) | 67.12 (64.99) | 52.25 (50.28) | 62.99 (61.88) | 51.70 (49.60) |
| + SNSCL | 74.26 (72.96) | 70.40 (70.01) | 83.67 (83.28) | 71.74 (71.22) | 78.84 (78.29) | 67.50 (66.48) | 74.52 (73.97) | 66.07 (65.26) |
| MW-Net | 73.68 (72.19) | 65.81 (65.19) | 76.27 (75.89) | 65.19 (63.32) | 72.76 (70.18) | 54.88 (51.80) | 67.44 (65.08) | 57.49 (56.10) |
| + SNSCL | 78.52 (78.03) | 72.68 (72.20) | 85.73 (85.44) | 75.69 (75.28) | 80.69 (80.22) | 70.49 (69.90) | 76.07 (76.70) | 68.95 (68.26) |
| MLC | 75.84 (74.99) | 69.81 (69.03) | 77.80 (77.29) | 67.93 (67.28) | 74.40 (73.91) | 59.44 (59.00) | 68.84 (68.21) | 58.73 (58.29) |
| + SNSCL | 79.22 (78.96) | 75.92 (75.57) | 87.05 (86.70) | 79.44 (79.21) | 82.75 (82.43) | 72.30 (71.96) | 76.91 (76.47) | 69.70 (69.24) |
| DivideMix | 79.39 (78.47) | 75.51 (73.67) | 79.34 (77.92) | 68.69 (68.63) | 76.57 (76.24) | 63.97 (63.28) | 72.76 (71.24) | 63.65 (62.68) |
| + SNSCL | 81.90 (81.72) | 77.19 (77.02) | 88.18 (87.94) | 81.44 (80.96) | 84.17 (84.03) | 74.80 (74.57) | 78.92 (78.56) | 71.28 (70.83) |
| **Avg.** ↑ | **5.55 (6.57)** | **7.81 (10.6)** | **9.70 (9.87)** | **11.28 (11.62)** | **13.61 (15.31)** | **16.73 (18.74)** | **8.51 (9.23)** | **10.46 (11.30)** |

Table 3: Comparisons with test acc. (%) on Clothing-1M.

| CE | JoCoR (Wei et al., 2020) | Joint Optim (Tanaka et al., 2018) | DivideMix (Li et al., 2020) | ELR+ (Liu et al., 2020) | SFT+ (Wei et al., 2022) | CE + SNSCL | DivideMix + SNSCL |
|---|---|---|---|---|---|---|---|
| 64.54 | 70.30 | 72.23 | 74.76 | 74.81 | 75.08 | **73.49** | **75.31** |

CE, label smooth (Lukasik et al., 2020), confidence penalty (Pereyra et al., 2017), Co-teaching (Han et al., 2018), JoCoR (Wei et al., 2020), DivideMix (Li et al., 2020), SYM (Wang et al., 2019), GCE (Zhang & Sabuncu, 2018), MW-Net (Shu et al., 2019), and MLC (Zheng et al., 2021). Detailed settings about these methods can be found in Appx. A.3.

**Results on four noisy benchmarks**. We compare 10 algorithms and attain significant improvement of top-1 testing accuracy (%) on four fine-grained benchmarks. We show the results in Tab. 1 and Tab. 2, where we test symmetric and asymmetric noise types. To demonstrate the effectiveness of our method, we give experimental comparisons from two aspects. 1) **Improvements on top-1 test accuracy**. Overall, our method SNSCL achieves consistent improvement in all noisy conditions. The average minimal improvement is **5.55%** in Stanford Dogs with 10% asymmetric noise, and the maximum is **21.41%** in Aircraft with 40% symmetric noise. 2) **Mitigating overfitting on fine-grained sets**. In these tables, we report the best accuracy and the last epoch's accuracy. It is noteworthy that the investigated methods mainly overfit on these benchmarks (*i.e.*, the accuracy attaches a peak and then drops gradually, causing a great gap between these two values). However, SNSCL mitigates overfitting and maintains more stable learning curves.

**Results on Clothing-1M**. We evaluate the effectiveness of our algorithm on Clothing-1M, a real-world noisy benchmark set with one million training images, and show the comparison results in Tab. 3. We select cross-entropy and DivideMix as the basic methods and integrate SNSCL with them. Obviously, the combination *DivideMix+SNSCL* outperforms the state-of-the-art method SFT+ by 0.23% top-1 test accuracy. Moreover, in contrast with the bases, SNSCL achieves remarkable improvements by 8.95% and 0.55%, respectively. These results demonstrate the effectiveness of our methods in real-world applications.

### 4.3 More Analysis

**Versatility**. SNSCL is flexible and can also be applied to current LNL methods for improving their performance on **generic noisy sets**. We select several LNL methods and combine them with our proposal. The study results on noisy CIFAR-10 & 100 are reported in Tab. 4. Overall, the performance of these five methods achieve non-trivial improvements by combining with SNSCL. We highlight Top-1 testing accuracy of Peer Loss improve over **18%** on CIFAR-100 by using SNSCL.

Table 4: Comparisons with test acc. (%) on **generic** classification task. The solid results denote the improvement of our method SNSCL. The average results among five times are reported.

| | CIFAR-10 | | CIFAR-100 | |
|---|---|---|---|---|
| | Symm. 40% | Asym. 40% | Symm. 40% | Asym. 40% |
| Peer Loss (Shu et al., 2019) | 84.29 / **92.21** | 85.18 / **91.59** | 50.53 / **69.82** | 50.17 / **68.90** |
| JoCoR (Wei et al., 2020) | 85.44 / **92.70** | 83.91 / **91.41** | 55.97 / **71.44** | 50.97 / **69.89** |
| CDR (Xia et al., 2021) | 86.13 / **93.83** | 85.79 / **92.08** | 60.18 / **71.95** | 59.49 / **71.57** |
| SFT (Wei et al., 2022) | 89.54 / **94.59** | 89.93 / **94.27** | 69.72 / **74.52** | 69.29 / **73.19** |
| DivideMix (Li et al., 2020) | 94.80 / **95.92** | 93.40 / **94.90** | 74.92 / **76.04** | 72.10 / **75.16** |

Table 5: Compared to Sel-CL+ (Li et al., 2022). The backbone is PreAct ResNet-18.

| Dataset | CIFAR-10 | | | | | CIFAR-100 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Noise | S. 20% | S. 50% | S. 80% | A. 20% | A. 40% | S. 20% | S. 50% | S. 80% | A. 20% | A. 40% |
| Sel-CL+ | 95.5 | 93.9 | 89.2 | 95.2 | 93.4 | 76.5 | 72.4 | 59.6 | 77.5 | 74.2 |
| Ours | **96.2** | **95.2** | **91.7** | **95.6** | **94.9** | 76.4 | **74.7** | **64.3** | 77.3 | **75.1** |

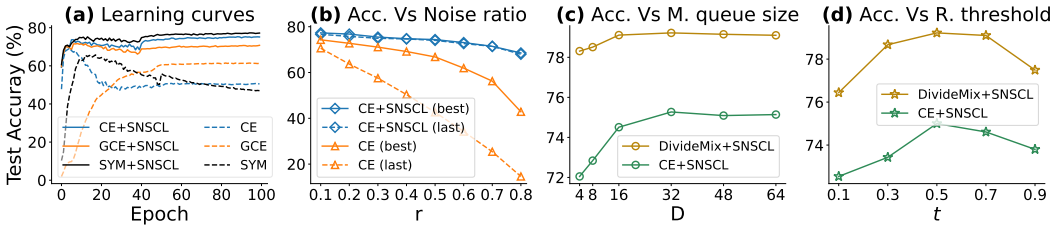



Figure 4: **More analyses** from four perspectives. We select Stanford dogs with 40% symmetric label noise as the test benchmarks and further analyze the effectiveness of our algorithm.

Table 6: **Ablation study** about the effectiveness of each component under 40% *symm.* label noise.

| Component | | | Stanford Dogs | | CUB-200-2011 | |
|---|---|---|---|---|---|---|
| Weight cor. | Weight update | Stoc. Module | CE+SNSCL | DivideMix+SNSCL | CE+SNSCL | DivideMix+SNSCL |
| | | | 69.20 (50.45) | 77.93 (76.28) | 54.14(45.85) | 67.35 (66.96) |
| ✓ | | | 71.44 (68.90) | 78.27 (78.04) | 63.11(60.41) | 69.57 (69.22) |
| ✓ | ✓ | | 74.11 (72.96) | 78.85 (78.61) | 66.44(65.10) | 72.10 (71.84) |
| ✓ | ✓ | ✓ | 75.27 (75.00) | 79.12 (78.91) | 68.83(68.67) | 73.66 (73.28) |

**Compared to Sel-CL+**. We conduct experiments to compare our method (DivideMix + SNSCL) with Sel-CL+ (Li et al., 2022), a LNL method based on contrastive learning. Detailed discussions about these two methods can be found in Related works. Tab. 5 reports the comparison results. Our method outperforms Sel-CL+ in most noisy settings. As the noise ratio arises, the achievements of SNSCL are more remarkable while the performance is improved by 2.5% on CIFAR-10 80% symm. noise, and 4.7% on CIFAR-100 80% symm. noise. Besides, thanks to the stochastic module, we merely adopt two simple augmentation strategies in our framework (see Appx. A.3).

**Effectiveness**. Our algorithm exhibits the superior effectiveness in two aspects. (1) we plot the curve of test accuracy in Fig. 4(a). It is clear that the accuracy of CE rises dramatically to a peak and gradually decreases, indicating overfitting to noise. For SNSCL, the testing curve is relatively stable and result in good generalization performance. (2) we test the noise ratios with a wide range of $r \in \{10\%, \cdots, 80\%\}$ and record the best and the last top-1 testing accuracy. As shown in the scatter plot 4(b), SNSCL can mitigate reasonable discriminability for a high noise ratio (68% acc. for symmetric 80% label noise). More results can be found in Appx. A.3.

**Sensibility**. We explore the effect of two essential hyper-parameters in our method. 1) **The momentum queue size** D. The batch size or momentum queue size is the key point in contrastive learning, and thus we set $\mathrm{D} \in \{4, 8, 16, 32, 48, 64\}$ to explore its influence on our framework. The results are shown in Fig. 4(c). As the size D reaches a certain amount, the performance will not increase. Thus, we set a suitable yet effective value $\mathrm{D} = 32$. 2) **The reliability threshold** $t$. This threshold in the weight-aware mechanism deeply affects the subsequent two weighted strategies. We adjust its value from the space $\{0.1, 0.3, 0.5, 0.7, 0.9\}$ and plot the results in Fig. 4(d). The best performance is attained on two conditions when $t = 0.5$. Therefore, we set the reliability threshold as $0.5$.

**Ablation study**. In our proposed SNSCL, there mainly exist three components, weighted-correction and weighted-update strategy in a weighted-aware mechanism and a stochastic module. We conduct the ablation study on two benchmarks to evaluate the effectiveness of each component and show the results in Tab. 6. Under the settings of Stanford dogs with 40% symmetric noisy labels, the combination of three components improves the performance of CE by more than 6% and the effect of DivideMix by 3% respectively, while all components bring some positive effects. To some extent, these results demonstrate the effectiveness of each part of our method.

**Visualization**. We visualize distributions of feature embeddings via t-SNE in Appx. A.3, verifying that SNSCL faithfully improves discriminability of feature extractors under varied noisy conditions.

## 5 Related Work

**Robust methods in Learning with noisy labels**. The methods in the field of learning with noisy labels can be roughly categorized into robust loss function, sample selection, label correction, and sample reweight. The early works (Zhang & Sabuncu, 2018; Wang et al., 2019; Ma et al., 2020; Liu & Guo, 2020) mainly focus on designing robust loss functions which provide the deep model with greater generalization performance compared with the cross-entropy loss and contain the theoretical guarantee (Ma et al., 2020; Liu & Guo, 2020). Currently, more works turn to explore the application of the other three strategies. In label correction, researchers refurbish the noisy labels by self-prediction of the model's output (Song et al., 2019; Wang et al., 2021b) or an extra meta-corrector (Wu et al., 2021; Zheng et al., 2021). The latter enables admirable results of correction with a small set of meta-data. In sample section, the key point is how effective the preset selection criterion is. Previous literatures leverage the small-loss criterion that selects the examples with small empirical loss as the clean one (Han et al., 2018; Wei et al., 2020). Recently, the works (Nguyen et al., 2020; Bai & Liu, 2021; Wei et al., 2022) represented by SELF (Nguyen et al., 2020) pay more attention to history prediction results, providing selection with more information and thus promoting the selection results. Besides, sample reweight methods (Shu et al., 2019; Ren et al., 2018) give examples with different weights, which can be regarded as a special form of sample selection. For example, Shu et al. (2019) designed a meta-net for learning the mapping from loss to sample weight. The samples with large losses are seen as the noise, and thus meta-net generates small weights.

**Contrastive learning**. As an unsupervised learning strategy, contrastive learning (Chen et al., 2020a;b; He et al., 2020) leverages similarity learning and markedly improves the performance of representation learning. The core idea of these methods is maximizing (minimizing) similarities of positive (negative) pairs at the data points. Further, to deeply explore the supervised information, supervised contrastive learning (Khosla et al., 2020) exploits labels and aims to reduce the distance between the embedding and its congeneric embeddings in the feature space.

CL has also been applied to LNL field for better representation learning. Sel-CL (Li et al., 2022) proposes a pair-wise framework of selecting clean samples and conducts contrastive learning on those samples. Our proposed NTSCL is different in three aspects: 1) a different selection strategy via a novel weight-aware mechanism; 2) a stochastic module avoiding manually-defined augmentations in SCL for LNL. 3) a plug-and-play module for typical LNL methods. NTSCL can be easily integrated into existing methods for improving performance on LNL or LNL-FG, while Sel-CL cannot.

## 6 Conclusion

In this work, we propose a novel task called LNL-FG, posing a more challenging noisy scenario to learning with noisy labels. For this, we design a general framework called SNSCL. SNSCL contains a noise-tolerated contrastive loss and a stochastic module. Compared with typical SCL, our contrastive learning framework incorporates a weight-aware mechanism which corrects noisy labels and selectively update momentum queue lists. Besides, we propose a stochastic module for feature transformation, generating the probabilistic distribution of feature embeddings. We achieve greater representation ability by sampling transformed embedding from this distribution. SNSCL is applicable to prevailing LNL methods and further improves their generalization performance on LNL-FG. Extensive experiments and analysis demonstrate the effectiveness of our method.

## REFERENCES


Yingbin Bai and Tongliang Liu. Me-momentum: Extracting hard confident examples from noisily labeled data. In *ICCV*, 2021.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *ECCV*, 2014.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020a.

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. In *NeurIPS*, 2020b.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. In *NeurIPS*, 2020.

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

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9, 2020.

Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *CVPR workshop*, 2011.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *NeurIPS*, 2020.

Durk P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In *NeurIPS*, 2015.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCV workshops*, 2013.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Citeseer, 2009.

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

Shikun Li, Xiaobo Xia, Shiming Ge, and Tongliang Liu. Selective-supervised contrastive learning with noisy labels. In *CVPR*, 2022.

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

Yang Liu and Hongyi Guo. Peer loss functions: Learning from noisy labels without knowing noise rates. In *ICML*, 2020.

Philip M Long and Rocco A Servedio. Random classification noise defeats all convex potential boosters. In *ICML*, 2008.

Michal Lukasik, Srinadh Bhojanapalli, Aditya Menon, and Sanjiv Kumar. Does label smoothing mitigate label noise? In *ICML*, 2020.

Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah Erfani, and James Bailey. Normalized loss functions for deep learning with noisy labels. In *ICML*, 2020.

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. In *arXiv preprint arXiv:1306.5151*, 2013.

Duc Tam Nguyen, Chaithanya Kumar Mummadi, Thi Phuong Nhung Ngo, Thi Hoai Phuong Nguyen, Laura Beggel, and Thomas Brox. Self: Learning to filter noisy labels with self-ensembling. In *ICLR*, 2020.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Taesung Park, Alexei A Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation. In *ECCV*, 2020.

Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. Regularizing neural networks by penalizing confident output distributions. In *ICLR*, 2017.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *ICML*, 2018.

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

Hwanjun Song, Minseok Kim, and Jae-Gil Lee. Selfie: Refurbishing unclean samples for robust deep learning. In *ICML*, 2019.

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

Ambuj Tewari and Peter L Bartlett. On the consistency of multiclass classification methods. *Journal of Machine Learning Research*, 2007.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. In *California Institute of Technology*, 2011.

Ximei Wang, Jinghan Gao, Mingsheng Long, and Jianmin Wang. Self-tuning for data-efficient deep learning. In *ICML*, 2021a.

Xinshao Wang, Yang Hua, Elyor Kodirov, David A Clifton, and Neil M Robertson. Proselflc: Progressive self label correction for training robust deep neural networks. In *CVPR*, 2021b.

Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *ICCV*, 2019.

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

Qi Wei, Haoliang Sun, Xiankai Lu, and Yilong Yin. Self-filtering: A noise-aware sample selection for label noise with confidence penalization. In *ECCV*, 2022.

Yichen Wu, Jun Shu, Qi Xie, Qian Zhao, and Deyu Meng. Learning to purify noisy labels via meta soft label corrector. In *AAAI*, 2021.

Xiaobo Xia, Tongliang Liu, Bo Han, Chen Gong, Nannan Wang, Zongyuan Ge, and Yi Chang. Robust early-learning: Hindering the memorization of noisy labels. In *ICLR*, 2021.

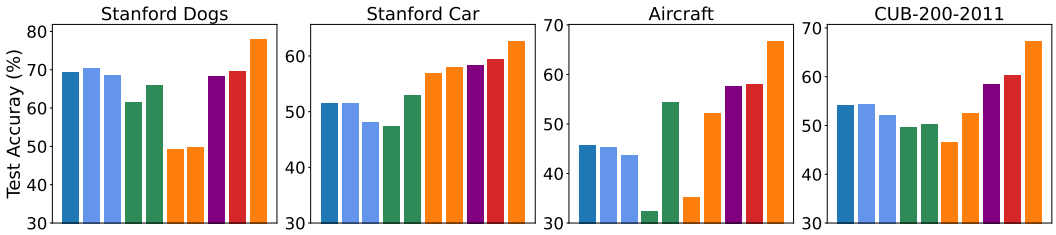
Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *CVPR*, 2015.

Yilun Xu, Peng Cao, Yuqing Kong, and Yizhou Wang. L_dmi: An information-theoretic noise-robust loss function. In *NeurIPS*, 2019.

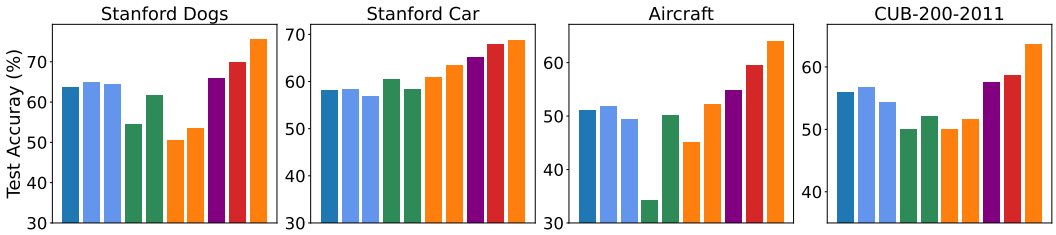
Zhilu Zhang and Mert R Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *NeurIPS*, 2018.

Guoqing Zheng, Ahmed Hassan Awadallah, and Susan Dumais. Meta label correction for noisy label learning. In *AAAI*, 2021.

# A APPENDIX




(a) *40% Symmetric noisy labels*




(b) *30% Asymmetric noisy labels*

Figure 5: Ten tested methods (left → right): cross-entropy, label smooth, confidence penalty, GCE, SYM, Co-teaching, JoCoR, MW-Net, MLC, DivideMix. Methods with same color belong to same LNL robust strategy. The **x-axis** denotes their performance on typical LNL task while the performance increases gradually from left to right.

Table 7: An ablation study on 40% symmetric noisy labels. The performance of Co-teaching can be improved by several robust techniques and gets close to the performance of DivideMix (the SOTA).

| Co-teaching | Mixup | Pseudo-label | Conf. reg. | EMA | Stanford Dogs | CUB-200-2011 |
|---|---|---|---|---|---|---|
| ✓ | | | | | 49.15 (48.92) | 46.57 (46.22) |
| ✓ | ✓ | | | | 62.79 (60.10) | 54.04 (53.09) |
| ✓ | ✓ | ✓ | | | 72.44 (71.97) | 65.77 (63.91) |
| ✓ | ✓ | ✓ | ✓ | | 75.21 (73.94) | 66.47 (65.73) |
| ✓ | ✓ | ✓ | ✓ | ✓ | 77.84 (77.41) | **67.64** (67.20) |
| DivideMix | | | | | **77.93** (76.28) | 67.35 (66.96) |

## A.1 A PRIOR STUDY

In this section, we conduct a preliminary investigation to evaluate the performance of current LNL on LNL-FG. Figure 5 and Table 7 exhibit the qualitative results. Our finds are divided into two parts,

- **Not all investigated algorithms can achieve significant performance for LNL-FG as they achieved in LNL, demonstrating the difficulty of fine-grained noisy settings.** In Stanford Dogs and CUB-200-2011, Cross-entropy, a non-robust method, attains competitive generalization performance while outperforming more than half methods. The insufficient robustness of these methods empirically demonstrates that LNL-FG poses a more challenging noisy condition for model learning and has not attracted much attention.
- **The generalization performance of LNL methods heavily relies on techniques that can mitigate overfitting on noisy labels.** In Table 7, we select co-teaching, a method with poor performance on LNL-FG, and add four robust techniques step by step. Each technique improves the performance of the basic method. Combining with Mixup, pseudo-label, confidence regularization, and EMA seriatim, top-1 testing accuracy of Co-teaching on the clean test set of *Stanford Dogs* is improved by 13.64%, 9.65%, and 2.77%, respectively. **However, integrating these existing techniques into the training process is difficult or requires customized adaptations, inspiring us to design a general method which can be applied to current LNL methods for improving their performance on LNL-FG.**

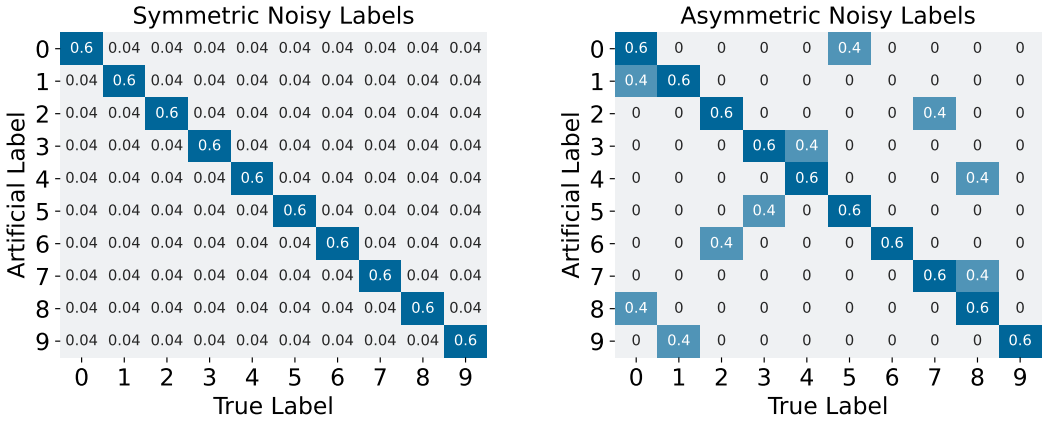



Figure 6: **Illustration** of symmetric and asymmetric noise transition matrix. There is a 10-classes classification task and the noise ratio is set as $r = 0.4$.

### A.2 More implementation details

**Settings of Figure 2.** We conduct an experiment to compare the convergence curves. 50 classes with 100 images per class are selected from ImageNet and Stanford Dogs, respectively, for simulating the generic and fine-grained sets. The backbone is ResNet-18. It is clear that constructing noisy labels on fine-grained sets yields greater negative effects of the noise, causing the model overfits more easily.

**Noise transition matrix**. We give an illustration of two types of the transition matrix in Figure 6.

**Settings of benchmarks**. In this work, we select four fine-grained datasets, two generic datasets, and one open-world noisy set to verify the effectiveness of our method. The detailed information of these benchmarks is shown in Table 8. For validation and hyper-parameter adjustment, we reserve 10% clean training samples and translate the partial labels of the rest to the noisy labels.

Table 8: Extended version of Table 1.

| Datasets | # Train | # Test | # Classes | # Size | # Features | Model | Warmup |
|---|---|---|---|---|---|---|---|
| ***Fine-grained set*** | | | | | | (pre-trained) | |
| Aircraft (Maji et al., 2013) | 6667 | 3333 | 100 | 224 | 512 | ResNet-18 | 10 |
| CUB-200-2011 (Wah et al., 2011) | 5994 | 5794 | 200 | 224 | 512 | ResNet-18 | 10 |
| Stanford-Cars (Krause et al., 2013) | 8114 | 8441 | 196 | 224 | 512 | ResNet-18 | 10 |
| Stanford-Dogs (Khosla et al., 2011) | 12000 | 8580 | 120 | 224 | 512 | ResNet-18 | 5 |
| ***Generic set*** | | | | | | | |
| CIFAR-10 (Krizhevsky et al., 2009) | 50000 | 10000 | 10 | 32 | 512 | PreAct ResNet-18 | 10 |
| CIFAR-100 (Krizhevsky et al., 2009) | 50000 | 10000 | 100 | 32 | 512 | PreAct ResNet-18 | 20 |
| ***Real-world set*** | | | | | | (pre-trained) | |
| Clothing-1M (Xiao et al., 2015) | 1000000 | 10000 | 14 | 224 | 2048 | ResNet-50 | 1 |

**Settings of comparison methods.** We compare our proposal with cross-entropy loss function and the following baselines:

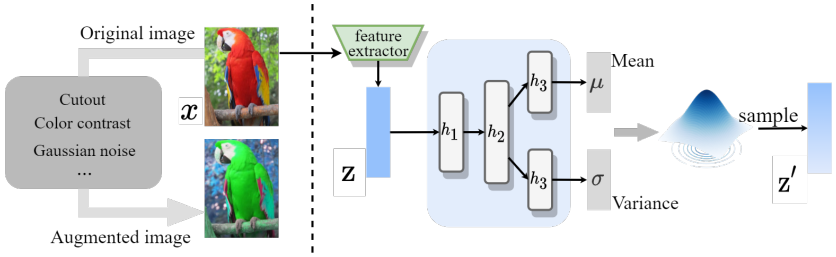
- **Label smooth** (Lukasik et al., 2020), which reassigns the sample label from a hard version to a soft version like $\{0, 0, 1\} \rightarrow \{0.05, 0.05, 0.9\}$. This method confronts the effects of noisy labels by mitigating over-confidence of the model on the given label.
- **Confidence penalty** (Pereyra et al., 2017), which stems from the motivation of penalizing low entropy output distributions. It connects a maximum entropy based confidence penalty to label smoothing through the direction of the KL divergence.
- **GCE** (Zhang & Sabuncu, 2018), which analyzes the robustness of MAE and the poor performance. Then, the author presents a theoretically grounded set of noise-robust loss functions that can be seen as a generalization of MAE and CCE.
- **SYM** (Wang et al., 2019), which obeys the paradigm of the symmetric loss function that ensembles CE and reversed CE. The latter is demonstrated as a robust loss function.

- **Co-teaching** (Han et al., 2018), which ensembles two branches for alternatively selecting samples with small losses and feeds them to another network training. *Co-training* strategy alleviates the error accumulation of the selection to some degree.
- **JoCoR** (Wei et al., 2020), which leverages the framework of Co-teaching and further designs a KL term for consistent output of two networks. It explores the lower bound of small loss and prompts accurate selection.
- **MW-Net** (Shu et al., 2019), which designs a meta-network for generating the sample weight via learning a function from loss to weight. The meta-weight is inserted into the training of the classification network by bi-level strategy.
- **MLC** (Zheng et al., 2021), which also designs a meta-network for label correction. It learns from the original label and feature embeddings and outputs the corrected label.
- **DivideMix** (Li et al., 2020), which belongs to a hybrid approach that bases on *sample selection* and ensembles co-training, pseudo-labeling, and Mixup. It attains state-of-the-art performance on LNL.

For fair comparisons, we keep the same hyper-parameters as they reported in their published versions, where some settings are marginally adjusted, and we report them in table 9. In addition, we adjust the selection process in Co-teaching (Han et al., 2018) and JoCoR (Wei et al., 2020) when combines with our algorithm. Since our algorithm changes the original noise ratio in the training set, we replace the pre-estimation of the noise ratio with the dynamic strategy (i.e., GMM fits the losses among all samples).

Table 9: Detailed settings of compared methods in experiments.

| Method | Settings |
|---|---|
| SYM (Wang et al., 2019) | SYM $= \alpha \times$ CE $+ \beta \times$ RCE, where $\alpha = 0.1, \beta = 1$ |
| Label Smooth (Lukasik et al., 2020) | smooth coefficient $\lambda = 0.1$ |
| MW-Net (Shu et al., 2019) | extra clean sample number $N = 5\times$ category number |
| MLC (Zheng et al., 2021) | extra clean sample number $N = 5\times$ category number |




(**a**) Typical augmentation in CL (**b**) Our stochastic module

Figure 7: **Illustration** of stochastic module. Compared to typical augmentation strategies in contrastive learning, we replace it with a stochastic module. Original feature embedding $\mathrm{z}_i$ is input into a stochastic network. Then, the augmented embedding $\mathrm{z}_i'$ can be sampled from the generated distribution. We consider that the property of stochastic provides more complex feature transformation than typical augmentation in images space, as well as avoiding manually defined augmentation strategies for different datasets.

Table 10: Test accuracy (%) of *CE + SNSCL* with different MLP architecture on 40% symmetric noisy labels. The average best score among three times are reported.

| Architecture $\{h_1, \cdots, h_n\}$ | Stanford Dogs | CUB-200-2011 | Aircraft | Stanford Cars |
|---|---|---|---|---|
| 512 - 1024 - 512 | 74.79 | 68.79 | 70.30 | 76.44 |
| **512 - 2048 - 512** | **75.27** | 68.83 | **70.48** | **76.72** |
| 512 - 4096 - 512 | 75.01 | **69.09** | 70.19 | 76.51 |
| 512 - 1024 - 1024 - 512 | 74.96 | 68.66 | 69.84 | 75.90 |
| 512 - 2048 - 2048 - 512 | 75.04 | 69.00 | 69.07 | 75.61 |

### A.3 More experimental results

**MLP structure of stochastic module.** In this paper, we build a stochastic module for learnable feature transformation, which is constructed as a three-layer MLP structure as shown in Figure 7 (**b**). Besides, we actually have tried different MLP architecture settings in the following experiments. Table 10 exhibits the comparison results with five structures. It can be seen that varying MLP settings do not remarkably affect the final results. Therefore, we prefer to adopt the simple yet effective one, *i.e.*, $\{h_1, h_2, h_3\} = \{512, 2048, 512\}$. Compared to the the backbone whose params is around 11.9 M, the learnable params of this module is only 0.06 M, which do not cause complex computation.

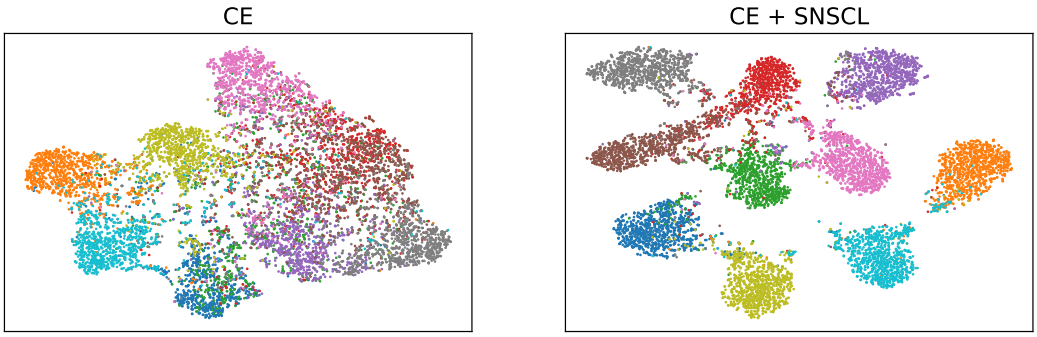
**Stochastic module vs. augmentation strategy**. We compare our proposed stochastic module with several complex augmentation strategies, including random cropping, cutout, color contrast, gaussian noise, and horizontal flipping. In Table 11, we give the comparison results with different strategies for feature transformation. The performance of our stochastic module consistently outperforms traditional augmentation strategies in image space. The average improvement is more than 1%. The results empirically demonstrate the superiority of our proposed stochastic module.

Table 11: Test accuracy (%) of *CE + SNSCL* with different strategies for feature augmentation on 40% symmetric noisy labels. The average best score among three times are reported.

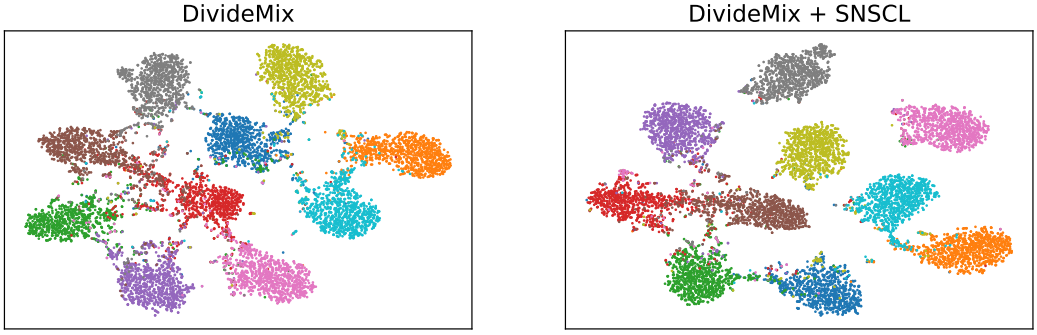
| | Stanford Dogs | CUB-200-2011 | Aircraft | Stanford Cars |
|---|---|---|---|---|
| Strong aug. | $74.02 \pm 0.28$ | $67.26 \pm 0.40$ | $70.19 \pm 0.26$ | $75.16 \pm 0.33$ |
| Ours | $\mathbf{75.27 \pm 0.24}$ | $\mathbf{69.09 \pm 0.41}$ | $\mathbf{70.48 \pm 0.33}$ | $\mathbf{76.72 \pm 0.29}$ |

**More results.** We give more detailed experimental results in the following

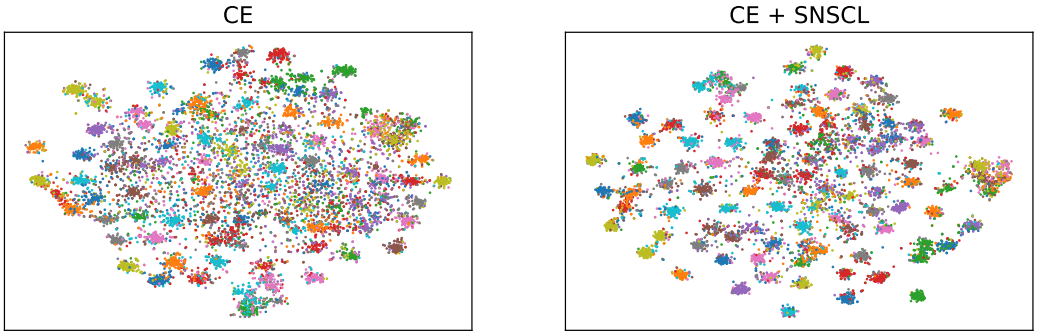
- **Visualization**. Figure 8 demonstrates that SNSCL improves the representation ability of the feature extractor under noisy conditions and achieves more discriminative class representation.
- **Robust learning curves**. Figure 9 shows the robust learning curves of our algorithm under all noise conditions.



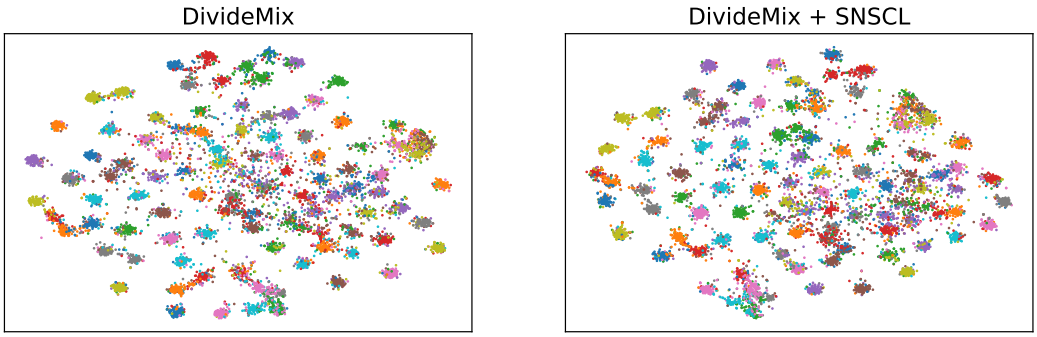

(a) **CE vs. CE+SNSCL**, *CIFAR-10 with 60% **symmetric** noisy labels*




(b) **DivideMix vs. DivideMix+SNSCL**, *CIFAR-10 with 60% **symmetric** noisy labels*

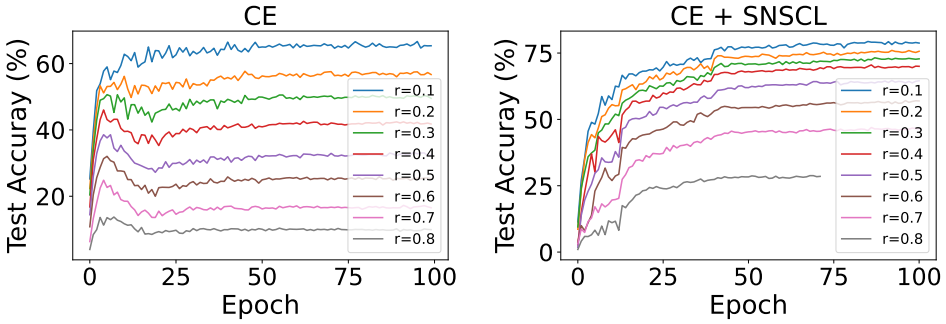



(c) **CE vs. CE+SNSCL**, *CIFAR-100 with 40% **asymmetric** noisy labels*

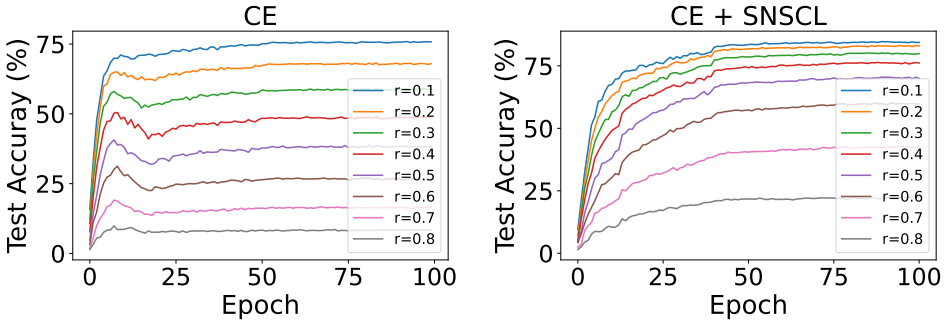



(d) **DivideMix vs. DivideMix+SNSCL**, *CIFAR-100 with 40% **asymmetric** noisy labels*

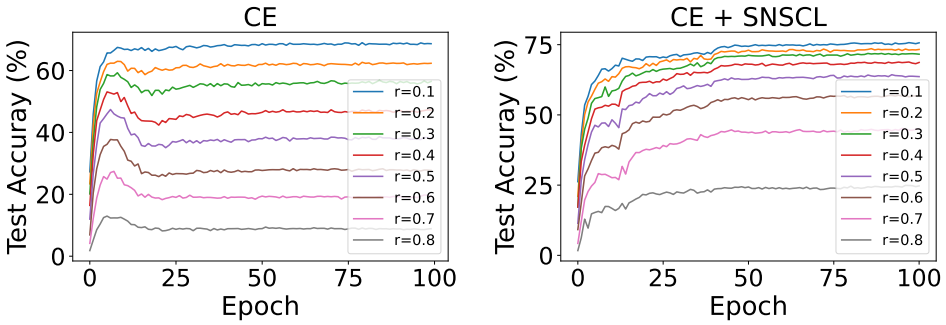
Figure 8: Compared to CE, our improvements in (a)(c) are remarkable. For DivideMix, our method generates more isolated clusters on CIFAR-10, as shown in (b). Since the similar performance between DivideMix and DivideMix+SNSCL (less than 2%), the improvement is not obvious in (d).



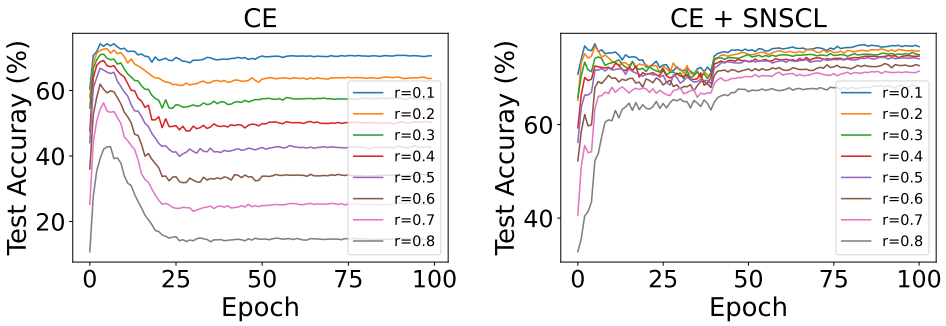

(a) **Stanford Car**, *Symmetric label noise, ResNet-18*




(b) **Aircraft**, *Symmetric label noise, ResNet-18*




(c) **CUB-200-2011**, *Symmetric label noise, ResNet-18*




(d) **Stanford Dogs**, *Symmetric label noise, ResNet-18*

Figure 9: **Comparisons with training curves as noise ratio increases**. We detailedly plot more training results about test accuracy (%) vs. noise ratio $r$, where there is symmetric noise and $r \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$. Under all noise ratio, SNSCL both remarkably improve the performance of the baseline.