Class2Simi: A Noise Reduction Perspective on Learning with Noisy Labels

Songhua Wu [*1] **Xiaobo Xia** [*1] **Tongliang Liu** [1]
Bo Han [2] **Mingming Gong** [3] **Nannan Wang** [4] **Haifeng Liu** [5] **Gang Niu** [6]

Abstract

Learning with noisy labels has attracted a lot of attention in recent years, where the mainstream approaches are in *pointwise* manners. Meanwhile, *pairwise* manners have shown great potential in supervised metric learning and unsupervised contrastive learning. Thus, a natural question is raised: does learning in a pairwise manner *mitigate* label noise? To give an affirmative answer, in this paper, we propose a framework called *Class2Simi*: it transforms data points with noisy *class labels* to data pairs with noisy *similarity labels*, where a similarity label denotes whether a pair shares the class label or not. Through this transformation, the *reduction of the noise rate* is theoretically guaranteed, and hence it is in principle easier to handle noisy similarity labels. Amazingly, DNNs that predict the *clean* class labels can be trained from noisy data pairs if they are first pretrained from noisy data points. Class2Simi is *computationally efficient* because not only this transformation is on-the-fly in mini-batches, but also it just changes loss computation on top of model prediction into a pairwise manner. Its effectiveness is verified by extensive experiments.

1. Introduction

It is difficult to label large-scale data accurately. Therefore, datasets with label noise are ubiquitous in the era of big data. However, label noise will degenerate the performance of deep networks, because deep networks will easily overfit label noise (Zhang et al., 2017). Almost all existing methods deal with the label noise problem in *pointwise* manners. Namely, these methods use pointwise losses (e.g., cross-entropy loss), and pointwise noise corrections (e.g., sample selection, loss correction, label correction, and others) (Xia et al., 2020a; Li et al., 2019; Zhang et al., 2018b; Xia et al., 2020d; Han et al., 2020b).

On the other hand, methods employing *pairwise* manners are very prevailing and have made a great success in machine learning, e.g., supervised metric learning and unsupervised contrastive learning (Qi et al., 2019; Boudiaf et al., 2020; Chen et al., 2020; He et al., 2020), where relationships between data points are exploited. Intuitively, the pairwise manners require less pointwise supervision information, i.e., class labels, and might be robust to label noise. In this paper, we naturally ask a question: does learning in a pairwise manner mitigate label noise? This motivates us to introduce a pairwise manner to deal with label noise.

Here we propose a noise reduction perspective on handling label noise: *Class2Simi*, i.e., transforming training data with noisy class labels into data pairs with noisy similarity labels. A class label shows the class that an instance belongs to, while a similarity label indicates whether or not two instances belong to the same class. We theoretically prove that through this transformation, the noise rate becomes lower (see Theorem 2). This is because, given a data pair, of which if one point has an incorrect class label or even if both points have incorrect class labels, the similarity label could be correct. Moreover, this transformation also reduces a multi-class classification problem into a binary classification problem. In label noise learning, the binary problem is easier to handle and a lower noise rate usually results in higher classification performance (Patrini et al., 2017).

Specifically, we illustrate the transformation and the robustness of similarity labels in Figure 1. In the middle column, we can see the noisy similarity labels of example-pairs (x_2, x_5) and (x_2, x_4) are correct, although there is one mislabeled point in (x_2, x_5), and two mislabeled points in (x_2, x_4). Moreover, if we assume that the noisy class labels in Figure 1 are generated according to the latent clean class labels and the class transition matrix (the ij-th entry of this matrix denotes the probability that the clean class label i flips into the noisy class label j), the noise rate of

[*]Equal contribution [1]Trustworthy Machine Learning Lab, School of Computer Science, The University of Sydney [2]Department of Computer Science, Hong Kong Baptist University [3]School of Mathematics and Statistics, The University of Melbourne [4]ISN State Key Laboratory, School of Telecommunications Engineering, Xidian University [5]Brain-Inspired Technology Co., Ltd. [6]RIKEN Center for Advanced Intelligence Project. Correspondence to: Tongliang Liu <tongliang.liu@sydney.edu.au>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

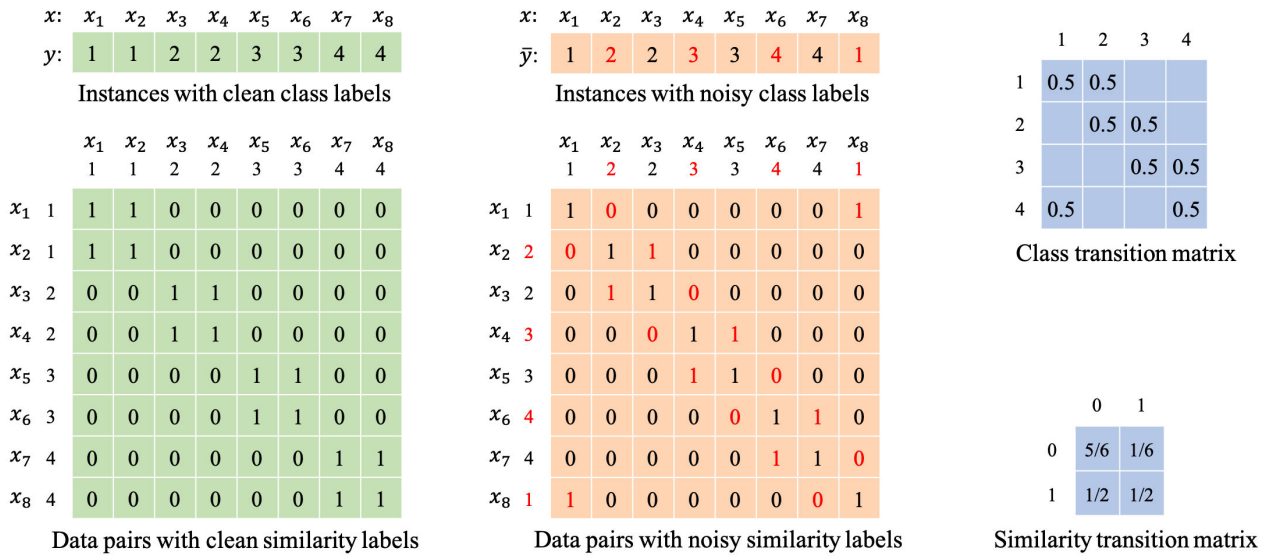


Figure 1. An illustration of the transformation from class labels to similarity labels. Note that $\bar{y}$ stands for the noisy class label and y for the latent clean class label. The labels marked in red are incorrect. If we assume the class label noise is generated according to the transition matrix presented in the upper part of the right column, it can be calculated that the noise rate for the noisy class labels is 0.5 while the noise rate for the noisy similarity labels is 0.25. Note that the transition matrix for similarity labels can be calculated by exploiting the class transition matrix as in Theorem 1.

class labels is 0.5. Meanwhile, the corresponding similarity transition matrix can be derived from the class transition matrix with the class-priors (see Theorem 1). The noise rate of similarity labels is 0.25, which is the proportion of the number of incorrect similarity labels to the number of total similarity labels.

To handle the transformed data pairs with noisy similarity labels, the connection between noisy similarity posterior and clean class posterior should be established. Intuitively, noisy similarity posterior can be linked to clean similarity posterior, and then clean class posterior can be inferred from clean similarity posterior. For the first part, we can draw on the philosophy of dealing with noisy class labels, e.g., selecting reliable data pairs for training, and correcting the similarity loss to learn a robust similarity classifier. For the second part, plenty of similarity metrics can be adopted. As an example, we could adapt the *Forward* (Patrini et al., 2017) to learn clean similarity posterior from data with noisy similarity labels. Then, by using the inner product of the clean class posterior (Hsu et al., 2019) to approximate clean similarity posterior, the clean class posterior (and thus the robust classifier) can thereby be learned.

It is obvious that Class2Simi suffers information loss because we can not recover the class labels from similarity labels, which implies that learning only from similarity labels can only cluster data points but can not identify the semantic classes of clusters. In Hsu et al. (2019), a pointwise cluster can be learned from similarity labels. However, in our case, the pairs with similarity labels are constructed from points with class labels, and we could acquire the semantic class information of clusters by pretraining the model from points with class labels without any additional information. Note that when class labels of points are corrupted, leading to noisy similarity labels, the proposed pretraining still works because the noisy class is assumed to be dominated by its clean class in label noise learning. Thus we do not suffer the major information loss in noisy similarity learning.

It is worthwhile to mention Class2Simi increases the computation cost very slightly, compared with the standard pointwise training. As shown in Figure 2, most computation is still pointwise. Only the computation of the pairwise enumeration layer (Hsu et al., 2018) and the loss are pairwise, while both the forward and backward propagation are pointwise. The pairwise enumeration layer was verified to only introduce a negligible overhead to the training time (Hsu et al., 2019). Moreover, the transformation is on-the-fly in mini-batches, which means the pairs are quadratic on the batch size other than the whole sample size.

The contributions of this paper are summarized as follows:

- We propose a noise reduction perspective on learning with noisy labels, which transforms class labels into similarity labels, reducing the noise rate.
- We provide a way to estimate the similarity transition matrix T_s by theoretically establishing its relation to the class transition matrix T_c. We show even if the T_c is roughly estimated, the induced T_s still works well.

- We design a deep learning method to learn robust classifiers from data with noisy similarity labels and theoretically analyze its generalization ability.
- We empirically demonstrate that the proposed method remarkably surpasses the baselines on many datasets with both synthetic noise and real-world noise.

The rest of this paper is organized as follows: In Section 2, we formalize the noisy multi-class classification problem. In Section 3, we propose the Class2Simi method and practical implementation. Experimental results are discussed in Section 4. We conclude our paper in Section 5.

2. Problem Setup and Related Work

Let $(X, Y) \in \mathcal{X} \times \{1, \ldots, c\}$ be the random variables for instances and clean labels, where $\mathcal{X}$ represents the instance space and c is the number of classes. However, in many real-world applications (Zhang et al., 2017; Zhong et al., 2019; Li et al., 2019; Tanno et al., 2019; Zhang et al., 2018b; Xia et al., 2021; Feng et al., 2020; Chou et al., 2020; Wu et al., 2020b; Zhu et al., 2021; Yu et al., 2020; Berthon et al., 2021), the clean labels cannot be observed. The observed labels are noisy. Let $\bar{Y}$ be the random variable for the noisy labels. What we have is a sample $\{(x_1, \bar{y}_1), \ldots, (x_n, \bar{y}_n)\}$ drawn from the noisy distribution $\mathcal{D}_\rho$ of the random variables $(X, \bar{Y})$. We aim to learn a robust classifier that could assign clean labels to test data by exploiting the sample with noisy labels.

Existing methods for learning with noisy labels can be divided into two categories: algorithms that result in statistically inconsistent or consistent classifiers. Methods in the first category usually employ heuristics to reduce the side-effect of noisy labels, e.g., selecting reliable samples (Han et al., 2018b; Yu et al., 2019; Wei et al., 2020; Wu et al., 2020a; Xia et al., 2020b), reweighting samples (Ren et al., 2018; Jiang et al., 2018; Ma et al., 2018; Kremer et al., 2018; Reed et al., 2015), correcting labels (Tanaka et al., 2018; Zheng et al., 2020), designing robust loss functions (Zhang & Sabuncu, 2018; Xu et al., 2019; Liu & Guo, 2020; Ma et al., 2020), employing side information (Vahdat, 2017; Li et al., 2017), and (implicitly) adding regularization (Li et al., 2021; 2017; Veit et al., 2017; Vahdat, 2017; Han et al., 2018a; Zhang et al., 2018a; Guo et al., 2018; Hu et al., 2020; Zhang et al., 2021; Han et al., 2020a). Those methods empirically work well in many settings. Methods in the second category aim to learn robust classifiers that could converge to the optimal ones defined by using clean data. They utilize the transition matrix, which denotes the probabilities that the clean labels flip into noisy labels, to build consistent algorithms (Natarajan et al., 2013; Scott, 2015; Liu & Tao, 2016; Patrini et al., 2017; Northcutt et al., 2017; Yu et al., 2018; Kremer et al., 2018; Hendrycks et al., 2018; Liu & Guo, 2020; Yao et al., 2020b; Xia et al., 2020c). The idea is that given the noisy class posterior probability and the transition matrix, the clean class posterior probability can be inferred.

Note that the noisy class posterior and the transition matrix can be estimated by exploiting the noisy data, where the transition matrix additionally needs anchor points (Liu & Tao, 2016; Patrini et al., 2017). Some methods assume anchor points have already been given (Yu et al., 2018). There are also methods showing how to identify anchor points from the noisy training data (Liu & Tao, 2016).

3. Class2Simi meets noisy supervision

In this section, we propose a new perspective for learning from noisy data. Our core idea is to transform class labels to similarity labels first, and then handle the noise manifested on similarity labels.

3.1. Transformation on labels and the transition matrix

As in Figure 1, we combine every 2 instances in pairs, and if the two instances have the same class label, we assign this pair a similarity label 1, otherwise 0. If the class labels are corrupted, the generated similarity labels also contain noise.

The definition of the similarity transition matrix is similar to the class one. The elements in a similarity transition matrix denote probabilities that clean similarity labels H flip into noisy similarity labels $\bar{H}$, i.e., $T_{s,mn} \coloneqq P(\bar{H} = n | H = m)$. The dimension of the similarity transition matrix is always 2×2. Since the similarity labels are generated from class labels, the similarity noise is determined and, thus can be calculated, by the class transition matrix.

Theorem 1 *Assume that the dataset is balanced (each class has the same amount of instances, and c classes in total), and the noise is class-dependent. Given a class transition matrix T_c, such that $T_{c,ij} = P(\bar{Y} = j | Y = i)$. The elements of the corresponding similarity transition matrix T_s can be calculated as*

$$T_{s,00} = \frac{c^2 - c - \left(\sum_j (\sum_i T_{c,ij})^2 - ||T_c||^2_{\text{Fro}}\right)}{c^2 - c},$$
$$T_{s,01} = \frac{\sum_j (\sum_i T_{c,ij})^2 - ||T_c||^2_{\text{Fro}}}{c^2 - c},$$
$$T_{s,10} = \frac{c - ||T_c||^2_{\text{Fro}}}{c}, \qquad T_{s,11} = \frac{||T_c||^2_{\text{Fro}}}{c}.$$

A detailed proof is provided in Appendix A.

Remark 1 *Theorem 1 can easily extend to the setting where the dataset is unbalanced in classes by multiplying each $T_{c,ij}$ by a coefficient n_i. n_i is the number of instances from the i-th class.*

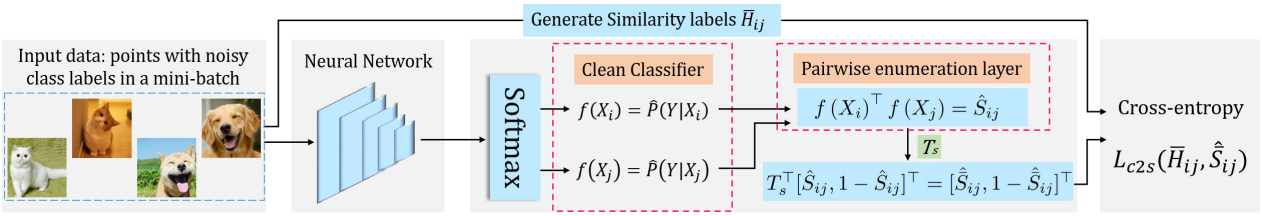


Figure 2. An overview of the proposed method. We add a pairwise enumeration layer and similarity transition matrix to calculate and correct the predicted similarity posterior. By minimizing the proposed loss L_{c2s}, a classifier f can be learned for assigning clean labels. The detailed structures of the Neural Network are provided in Section 4.

Note that the similarity labels are only dependent on class labels. If the class noise is class-dependent, the similarity noise is also 'class-dependent' (class means similar and dissimilar). Under class-dependent label noise, a binary classification is learnable as long as $T_{00} + T_{11} > 1$ (Menon et al., 2015), where T is the corresponding binary transition matrix; a multi-class classification is learnable if the corresponding transition matrix T_c is invertible. For Class2Simi, in the most general sense, i.e., T_c is invertible, $T_{s,00} + T_{s,11} > 1$ holds. Namely, the learnability of the pointwise classification implies the learnability of the reduced pairwise classification. A proof is provided in Appendix B. However, the latter cannot imply the former: As shown in Figure 1, the class transition matrix is not invertible, and thus the pointwise classification is not learnable while the reduced pairwise classification is learnable. Note that this 'learnable' is only for the binary pairwise classification in this case. Technically, two conditions must be met to learn a pointwise classifier from pairwise data: (1) The reduced pairwise classification is learnable; (2) The semantic class information is learnable. Generally, the second condition is equivalent to the learnability of the pointwise classification. Thus the learnability for a pointwise classifier of the two learning manners is consistent.

Theorem 2 *Assume that the dataset is balanced (each class has the same amount of samples), and the noise is class-dependent. When the number of classes $c \geq 8$, the noise rate of noisy similarity labels is lower than that of the noisy class labels.*

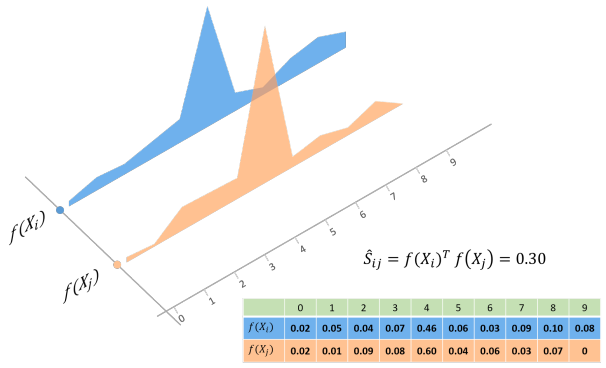
A detailed proof is provided in Appendix C.

In multi-class classification problems, the number of classes is usually larger than 8. As c becomes larger, the range of 'dissimilarity' of data pairs becomes larger, which is conducive to the reduction of the noise rate. Through Class2Simi, the number of *d-pairs* (with similarity label 0) is $(c-1)$ times as much as that of *s-pairs* (with similarity label 1). Meanwhile, compared with the original noise rate of noisy class labels, the noise rate of noisy similarity labels of s-pairs is higher and that of d-pairs is lower, while the overall noise rate of data pairs is lower, which partially reflects that the impact of label noise is less bad. Notably, the flip from 'dissimilar' to 'similar' should be more adversarial and thus more important. In practice, it is common that one class has more than one clusters, while it is rare that two or more classes are in the same cluster. If there is a flip from 'similar' to 'dissimilar' and based on it we split a (latent) cluster into two (latent) clusters, we still have a high chance to label these two clusters correctly later. If there is a flip from 'dissimilar' to 'similar' and based on it we join two clusters belonging to two classes into a single cluster, we nearly have zero chance to label this cluster correctly later. As a consequence, the flip from 'dissimilar' to 'similar' is more adversarial and important, thus deserving a larger weight when calculating the noise rate. Here we assign all data pairs the same weight, otherwise, there would be a more reduction of the noise rate. On balance, considering the reduction of the overall noise rate is meaningful.

When dealing with label noise, a low noise rate has many benefits. The most important one is that the noise-robust algorithms will consistently achieve higher performance when the noise rate is lower (Han et al., 2018b; Xia et al., 2019; Patrini et al., 2017). Another benefit is that, when the noise rate is low, the complex instance-dependent label noise can be well approximated by class-dependent label noise (Cheng et al., 2020), which is easier to handle.

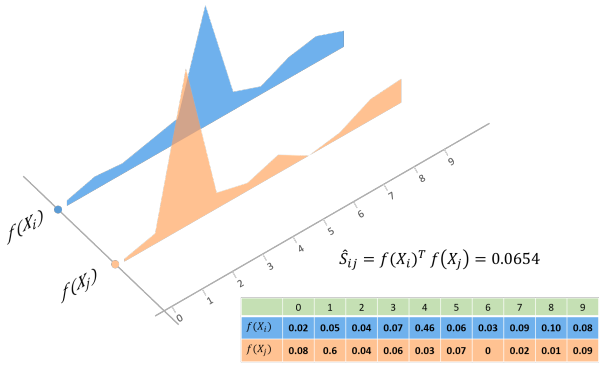
3.2. Learning with noisy similarity labels

In order to learn a multi-class classifier from similarity labeled data, we should establish relationships between class posterior probability and similarity posterior probability. Here we employ the relationship established in (Hsu et al., 2019), which is derived from a likelihood model. As in Figure 2, we denote the predicted clean similarity posterior by the inner product between two categorical distributions: $\hat{S}_{ij} = f(X_i)^\top f(X_j)$. Intuitively, $f(X)$ outputs the predicted categorical distribution of input data X and $f(X_i)^\top f(X_j)$ can measure how similar the two distributions are. For clarity, we visualize the predicted similarity posterior in Figure 3. If X_i and X_j are predicted belonging to the same class, i.e., $\operatorname{argmax}_{m \in c} f_m(X_i) = \operatorname{argmax}_{n \in c} f_n(X_j)$, the predicted



	0	1	2	3	4	5	6	7	8	9
$f(X_i)$	0.02	0.05	0.04	0.07	0.46	0.06	0.03	0.09	0.10	0.08
$f(X_j)$	0.02	0.01	0.09	0.08	0.60	0.04	0.06	0.03	0.07	0

(a) *Similar example*



	0	1	2	3	4	5	6	7	8	9
$f(X_i)$	0.02	0.05	0.04	0.07	0.46	0.06	0.03	0.09	0.10	0.08
$f(X_j)$	0.08	0.6	0.04	0.06	0.03	0.07	0	0.02	0.01	0.09

(b) *Dissimilar example*

Figure 3. Examples of predicted noisy similarity. Assume class number is 10; $f(X_i)$ and $f(X_j)$ are categorical distribution of X_i and X_j respectively, which are shown above in the form of area charts. $\hat{S}_{ij}$ is the predicted similarity posterior between two instances, calculated by the inner product between two categorical distributions.

similarity posterior should be relatively high ($\hat{S}_{ij} = 0.30$ in Figure 3(a)). By contrast, if X_i and X_j are predicted belonging to different classes, the predicted similarity posterior should be relatively low ($\hat{S}_{ij} = 0.0654$ in Figure 3(b)). Note that the noisy similarity posterior $P(\bar{H}_{ij}|X_i, X_j)$ and clean similarity posterior $P(H_{ij}|X_i, X_j)$ satisfy

$$P(\bar{H}_{ij}|X_i, X_j) = T_s^\top P(H_{ij}|X_i, X_j). \tag{1}$$

Therefore, we can infer the predicted noisy similarity posterior $\hat{\bar{S}}_{ij}$ from the predicted clean similarity posterior $\hat{S}_{ij}$ with the similarity transition matrix. To measure the error between the predicted noisy similarity posterior $\hat{\bar{S}}_{ij}$ and noisy similarity label $\bar{H}_{ij}$, we employ a binary cross-entropy loss function. The final optimization function is

$$\begin{aligned} &L_{c2s}(\bar{H}_{ij}, \hat{\bar{S}}_{ij}) \\ &\quad = -\sum_{i,j} \bar{H}_{ij} \log \hat{\bar{S}}_{ij} + (1 - \bar{H}_{ij}) \log(1 - \hat{\bar{S}}_{ij}). \end{aligned}$$

The pipeline of the proposed Class2Simi is summarized in Figure 2. The softmax function outputs an estimation for the clean class posterior, i.e., $f(X) = \hat{P}(Y|X)$, where $\hat{P}(Y|X)$ denotes the estimated class posterior. Then a pairwise enumeration layer is added to calculate the predicted clean similarity posterior $\hat{S}_{ij}$ of every two instances. According to Equation (1), by pre-multiplying the transpose of the noise similarity transition matrix, we can obtain the predicted noisy similarity posterior $\hat{\bar{S}}_{ij}$. Therefore, by minimizing L_{c2s}, we can learn a classifier for predicting noisy similarity labels. Meanwhile, before the transition matrix layer, the pairwise enumeration layer will output a prediction for the clean similarity posterior, which guides $f(X)$ to predict clean class labels.

Remark 2 *For a better understanding, we formulate Class2Simi in the form combined with Forward as an illustration. However, Class2Simi is a meta method that can be applied on top of sample selection, loss correction, label correction, and many other label noise learning methods. We provide another implementation with Reweight in Appendix D.*

3.3. Implementation

The proposed algorithm is summarized in Algorithm 1. Since learning only from similarity labels will lose the semantic class information, we load the model trained on the data with noisy class labels to provide the semantic class information for similarity learning in Stage 2.

3.4. Generalization error

We formulate the above problem in the traditional risk minimization framework (Mohri et al., 2018). The expected and empirical risks of employing estimator f can be defined as

$$R(f) = E_{(X_i, X_j, \bar{Y}_i, \bar{Y}_j, \bar{H}_{ij}, T_s) \sim \mathcal{D}_\rho}[\ell(f(X_i), f(X_j), T_s, \bar{H}_{ij})],$$

and

$$R_n(f) = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \ell(f(X_i), f(X_j), T_s, \bar{H}_{ij}),$$

where n is the training sample size of the noisy data. Assume that the neural network has d layers with parameter matrices $W_1, \ldots, W_d$, and the activation functions $\sigma_1, \ldots, \sigma_{d-1}$ are Lipschitz continuous, satisfying $\sigma_j(0) = 0$. We denote by $H : X \mapsto W_d \sigma_{d-1}(W_{d-1}\sigma_{d-2}(\ldots \sigma_1(W_1 X))) \in \mathbb{R}$ the standard form of the neural network. $H = \operatorname{argmax}_{i \in \{1,\ldots,c\}} h_i$. Then the output of the softmax function is defined as

Algorithm 1 Class2Simi

Input: training data with noisy class labels; validation data with noisy class labels.
Stage 1: Learn $\hat{T}_s$
1: Learn $g(X) = \hat{P}(\bar{Y}|X)$ by training data with noisy class labels, and save the model for Stage 2;
2: Estimate $\hat{T}_c$ following the optimization method in (Patrini et al., 2017);
3: Transform $\hat{T}_c$ to $\hat{T}_s$.
Stage 2: Learn the classifier $f(X) = \hat{P}(Y|X)$
4: Load the model saved in Stage 1, and train the whole pipeline showed in Figure 2.
Output: classifier f.

$f_i(X) = \exp(h_i(X))/\sum_{j=1}^{c}\exp(h_j(X)), i = 1, \ldots, c$. We can then obtain the following generalization error bound.

Theorem 3 *Assume the parameter matrices $W_1, \ldots, W_d$ have Frobenius norm at most $M_1, \ldots, M_d$, and the activation functions are 1-Lipschitz, positive-homogeneous, and applied element-wise (such as the ReLU). Assume the transition matrix is given, and the instances X are upper bounded by B, i.e., $\|X\| \leq B$ for all X, and the loss function ℓ is upper bounded by M. Then, for any $\delta > 0$, with probability at least $1-\delta$,*

$$R(\hat{f}) - R_n(\hat{f}) \leq M\sqrt{\frac{\log 1/\delta}{2n}} + \frac{(T_{s,11} - T_{s,01})2Bc(\sqrt{2d\log 2}+1)\Pi_{i=1}^{d}M_i}{T_{s,11}\sqrt{n}}. \quad (2)$$

A detailed proof are provided in Appendix E.

Theorem 3 implies that if the training error is small and the training sample size is large, the expected risk $R(\hat{f})$ of the representations for noisy similarity posterior will be small. If the transition matrix is well estimated, the clean similarity posterior as well as the classifier for the clean class will also have a small risk according to Equation (1) and the Class2Simi relations. This theoretically justifies why the proposed method works well. In the experiment section, we will show that the transition matrices are well estimated and that the proposed method significantly outperforms the baselines.

In Class2Simi, a multi-class classification is reduced to a pairwise binary classification. For data pairs, if a surrogate loss is classification-calibrated, minimizing it leads to minimizing the zero-one loss on the pointwise random variables in the limit case. Otherwise, we cannot guarantee the worst-case learnability of learning pointwise labels from pairwise labels, but it cannot imply the average-case non-learnability either. Theoretically, (Bao et al., 2020) proved that when the pairwise labels are all correct, for the special case $c = 2$, a good model for predicting s-/d-pairs must also be a good model for predicting the original classes, under mild assumptions. In practice, it seems fine to use non-classification-calibrated losses. According to (Tewari & Bartlett, 2007), the multi-class margin loss (i.e., one-vs-rest loss) and the pairwise comparison loss (i.e., one-vs-one loss) are proved to be non-calibrated, but they are still the main multi-class losses in (Mohri et al., 2018; Shalev-Shwartz & Ben-David, 2014).

4. Experiments

Experiment setup. We employ three widely used image datasets, i.e., *MNIST* (LeCun, 1998), *CIFAR-10*, and *CIFAR-100* (Krizhevsky et al., 2009), one text dataset *News20*, and one real-world noisy dataset *Clothing1M* (Xiao et al., 2015). *News20* is a collection of approximately 20,000 newsgroup documents, partitioned nearly evenly across 20 different newsgroups. *Clothing1M* has 1M images with real-world noisy labels and additional 50k, 14k, 10k images with clean labels for training, validation and test, and we only use noisy training set in the training phase. Note that the similarity learning method of Class2Simi is based on *clustering* because there is no class information. Intuitively, for a noisy class, if most instances in it belong to another specific class, we can hardly identify it. For example, assume that a class with noisy labels $\bar{i}$ contains n_i instances with ground-truth labels i and n_j instances with ground-truth labels j. If n_j is bigger than n_i, the model will cluster class i into j. Unfortunately, in *Clothing1M*, most instances with label '5' belong to class '3' actually. Therefore, we merge the two classes and denote the modified dataset by *Clothing1M** which contains 13 classes. For all the datasets, we leave out 10% of the training data as a validation set, which is for model selection.

For *MNIST*, *CIFAR-10*, and *CIFAR-100*, we use LeNet (LeCun et al., 1998), ResNet-26 with shake-shake regularization (Gastaldi, 2017), and ResNet-56 with pre-activation (He et al., 2016b), respectively. For *News20*, we first use GloVe (Pennington et al., 2014) to obtain vector representations for the raw text data, and employ a 3-layer MLP with the Softsign active function. For *Clothing1M**, we use pre-trained ResNet-50 (He et al., 2016a). Further details for the experiments are provided in Appendix F.1.

Noisy labels generation. For clean datasets, we artificially corrupt the class labels of training and validation sets according to the class transition matrix. Specifically, for each instance with clean label i, we replace its label by j with a probability of $T_{c,ij}$. In this paper, we consider both symmetric and asymmetric noise settings which are defined in Appendix F.2. *Sym-0.2* means symmetric noise type with a 0.2 noise rate and *Asym-0.2* means asymmetric noise type with a 0.2 noise rate.

Table 1. Means and Standard Deviations of Classification Accuracy over 5 trials on image datasets.

MNIST	Sym-0.2	Sym-0.4	Sym-0.6	Asym-0.2	Asym-0.4	Asym-0.6
Co-teaching	97.34±0.26	94.68±0.52	93.36±0.47	97.37±0.20	96.63±0.41	91.33±0.38
JoCor	97.48±0.12	96.31±0.20	93.18±0.27	97.31±0.09	95.73±0.29	91.43±0.28
PHuber-CE	98.65±0.18	98.17±0.15	97.63±0.36	98.73±0.09	98.36±0.25	97.37±0.41
APL	98.77±0.21	97.06±0.37	97.67±0.35	98.72±0.10	98.45±0.29	97.58±0.25
S2E	98.96±0.27	93.27±2.18	89.37±0.70	99.19±0.05	94.47±1.08	92.36±2.40
Revision	98.92±0.09	98.42±0.50	98.10±0.37	98.97±0.06	98.58±0.19	98.21±0.19
Reweight	98.78±0.16	98.26±0.22	97.02±0.58	98.62±0.19	98.12±0.31	96.98±0.29
Forward	98.76±0.03	98.37±0.25	96.89±0.49	98.61±0.22	98.08±0.33	97.43±0.25
R-Class2Simi	99.04±0.06	98.87±0.06	98.40±0.17	99.06±0.05	98.75±0.08	98.23±0.20
F-Class2Simi	**99.26±0.07**	**99.18±0.06**	**98.91±0.09**	**99.26±0.05**	**99.08±0.07**	**98.91±0.07**
CIFAR10	Sym-0.2	Sym-0.4	Sym-0.6	Asym-0.2	Asym-0.4	Asym-0.6
Co-teaching	88.92±0.45	85.97±1.02	75.97±1.33	89.14±0.36	84.77±1.08	76.07±1.27
JoCor	88.46±0.25	85.19±0.75	77.03±0.92	88.96±0.70	85.19±0.58	75.76±1.31
PHuber-CE	90.37±0.26	86.05±0.37	74.06±0.92	90.73±0.22	86.06±0.53	73.25±1.04
APL	89.07±0.92	85.77±0.84	70.06±1.06	89.97±0.19	85.60±0.91	72.33±1.68
S2E	90.04±1.22	82.05±1.95	57.96±4.70	90.12±0.97	83.16±1.58	64.77±3.06
Revision	90.02±0.48	85.47±0.71	73.92±2.02	89.77±0.28	85.32±1.36	75.24±1.87
Reweight	89.05±0.32	84.60±0.45	74.87±1.18	89.28±0.26	84.61±0.62	72.77±1.91
Forward	89.63±0.20	87.08±0.31	73.24±1.33	90.03±0.41	86.64±0.71	77.41±0.43
R-Class2Simi	90.91±0.26	87.80±0.23	79.19±1.65	91.07±0.21	87.78±0.33	78.56±0.63
F-Class2Simi	**91.38±0.19**	**88.22±0.19**	**79.45±0.53**	**91.24±0.27**	**87.79±0.36**	**79.05±0.56**
CIFAR100	Sym-0.2	Sym-0.4	Sym-0.6	Asym-0.2	Asym-0.4	Asym-0.6
Co-teaching	57.14±0.49	52.62±1.03	37.32±1.67	57.82±0.37	51.32±0.83	35.32±1.68
JoCoR	58.32±0.71	51.76±1.07	37.02±1.33	58.61±0.30	49.18±1.05	37.09±1.82
PHuber-CE	57.90±0.31	52.36±0.77	37.93±0.86	57.33±0.71	51.29±0.96	36.03±1.34
APL	54.03±0.92	49.06±0.93	36.06±2.02	55.62±0.92	48.37±0.94	35.02±1.72
S2E	59.37±1.09	43.29±1.94	30.08±3.91	58.92±1.21	42.88±2.16	29.93±4.05
Revision	59.62±0.97	53.26±0.84	35.82±2.06	58.77±0.93	52.72±1.38	37.72±1.75
Reweight	49.59±0.74	39.72±0.57	22.79±1.35	48.87±0.96	36.65±0.90	17.24±1.97
Forward	48.68±0.57	39.78±1.23	27.01±0.89	47.90±0.23	37.89±0.57	21.71±1.53
R-Class2Simi	55.45±0.55	50.38±0.49	35.57±0.75	54.95±0.65	47.56±0.72	34.82±0.58
F-Class2Simi	**60.26±0.18**	**54.85±0.60**	**40.38±0.58**	**59.10±0.13**	**52.99±0.78**	**38.69±2.84**

Baselines. In this paper, we compare our method with the following baselines: *Reweight* (Liu & Tao, 2016), *Forward* (Patrini et al., 2017), and *Revision* (Xia et al., 2019), which utilize a class-dependent transition matrix to model the noise, and learn a robust classifier. Besides, we externally conduct experiments on *Co-teaching* (Han et al., 2018b), which is a representative algorithm of selecting reliable samples for training; *JoCoR* (Wei et al., 2020), which employs a joint loss function to select small-loss samples; *PHuber-CE* (Menon et al., 2020), which introduces gra-

Table 2. Means and Standard Deviations of Classification Accuracy over 5 trials on text datasets.

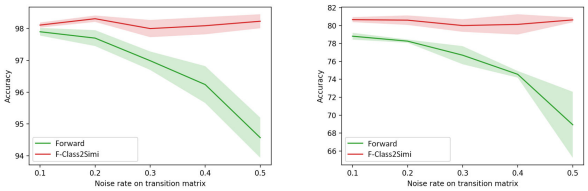
NEWS20	Sym-0.2	Sym-0.4	Sym-0.6	Asym-0.2	Asym-0.4	Asym-0.6
Co-teaching	55.32±0.28	51.09±1.06	47.07±0.83	55.29±0.41	53.08±0.26	45.63±0.75
JoCor	52.21±0.70	49.84±0.92	48.83±0.43	55.58±0.27	49.35±0.62	46.21±0.73
PHuber-CE	55.73±0.38	54.33±0.92	45.05±0.49	56.76±0.26	51.15±0.65	41.59±1.05
APL	56.91±0.21	53.12±1.21	43.60±1.28	56.11±0.23	50.93±1.05	43.60±1.28
S2E	57.93±0.37	47.16±1.32	28.53±5.04	54.89±1.92	50.42±1.71	30.67±3.12
Revision	58.06±0.19	52.30±1.73	46.84±1.09	56.41±0.77	53.44±0.83	43.77±1.08
Reweight	53.34±1.08	50.15±1.33	44.73±0.79	53.37±0.66	49.82±0.44	39.46±1.27
Forward	57.30±0.32	53.94±0.42	46.91±1.48	53.58±0.54	49.90±1.44	42.55±3.81
R-Class2Simi	**58.67±0.38**	56.59±0.74	50.48±0.97	58.44±0.66	**55.03±1.55**	**47.75±2.17**
F-Class2Simi	58.27±0.47	**56.70±1.13**	**50.18±0.89**	**58.46±0.68**	54.92±1.66	46.07±3.54

dient clipping to mitigate the effects of noise; *APL* (Ma et al., 2020), which applies simple normalization on loss functions and makes them robust to noisy labels; *S2E* (Yao et al., 2020a), which properly controls the sample selection process so that deep networks can benefit from the memorization effect. Besides, we conduct experiments on another implementation of the proposed method, which employs *Reweight* (More details are provided in Appendix D). To distinguish these two methods, we call them 'F-Class2Simi' and 'R-Class2Simi'.

Results on noisy image datasets. The results in Table 1 and Figure 4 demonstrate that Class2Simi achieves distinguished classification accuracy and is robust against the estimation errors on the transition matrix.

From Table 1, overall, we can see that after the transformation, better performance are achieved due to a lower noise rate and the similarity transition matrix being robust to noise. Even for challenging noise rates of 0.6, Class2Simi achieves good accuracy, uplifting about 5 and 10 points on *CIFAR-10* and *CIFAR-100* respectively, compared with the corresponding pointwise methods.

In Figure 4, we show that the similarity transition matrix is robust against estimation errors. To verify this, we add some random noise to the ground-truth T_c through multiplying every element in class T_c by a random variable α_{ij}. We control the noise rate on the T_c by sampling α_{ij} in different intervals, i.e., 0.1 noise means that α_{ij} is uniformly sampled from $\pm[1.1, 1.2]$. Then we normalize T_c to make its row sums equal to 1. From Figure 4, we can see that the accuracy of Forward drops dramatically with the increase of the noise on T_c. By contrast, there is only a slight fluctuation of F-Class2Simi, indicating Class2Simi is robust against the estimation errors on the transition matrix.



(a) *MNIST* (b) *CIFAR10*

Figure 4. Means and Standard Deviations of Classification Accuracy over 5 trials on *MNIST* and *CIFAR10* with perturbational ground-truth $\hat{T}_c$.

Table 3. Classification Accuracy on *Clothing1M**.

Co-teaching	74.70	JoCoR	74.98
PHuber-CE	73.16	APL	58.93
S2E	72.30	Revision	74.65
Forward	73.88	F-Class2Simi	75.41
Reweight	74.44	R-Class2Simi	**75.76**

Table 4. Classification Accuracy on clean datasets. CE uses class labels and the cross-entropy loss function. C2S refers to Class2Simi.

Dataset	***MNIST***	***CIFAR10***	***CIFAR100***	***News20***
CE	**99.30±0.02**	94.03±0.14	58.74±0.51	**59.86±0.39**
C2S	99.24±0.05	**94.05±0.27**	**60.36±0.89**	59.74±0.20

Results on the noisy text dataset. Results in Table 2 show that the proposed method works well on the text dataset under both symmetric and asymmetric noise settings.

Results on the real-world noisy dataset. Results in Table

3 show that the proposed method also performs well against agnostic noise.

Ablation study. To investigate how the similarity loss function influences the classification accuracy, we conduct experiments with the cross-entropy loss function and the similarity loss function on clean datasets over 3 trials, where the T_c is set to an identity matrix. All other settings are kept the same. As shown in Table 4, on *MNIST*, *CIFAR10*, and *News20*, the similarity loss function does not improve the classification accuracy on clean data, and on *CIFAR100*, the improvement is marginal. However, in Table 1 and 2, the improvements are significant, which reflects the improvements are mainly benefited from the lower noise rate and the reduced noisy binary paradigm.

5. Conclusion

This paper proposes a noise reduction perspective on dealing with class label noise by transforming training data with noisy class labels into data pairs with noisy similarity labels. We establish the connection between noisy similarity posterior and clean class posterior and propose a deep learning framework to learn classifiers from the transformed noisy similarity labels. The core idea is to transform pointwise information into pairwise information, which makes the noise rate lower. We also prove that not only the similarity labels but the similarity transition matrix is robust to noise. Experiments are conducted on benchmark datasets, demonstrating the effectiveness of our method. In future work, investigating different types of noise for diverse real-life scenarios might prove important.

Acknowledgments

SHW, XBX, and TLL were supported by Australian Research Council Project DE-190101473. BH was supported by the RGC Early Career Scheme No. 22200720, NSFC Young Scientists Fund No. 62006202 and HKBU CSD Departmental Incentive Grant. NNW was supported by National Natural Science Foundation of China Grant 61922066, Grant 61876142. GN was supported by JST AIP Acceleration Research Grant Number JPMJCR20U3, Japan. We thank the anonymous reviewers for their constructive comments.

References

Bao, H., Shimada, T., Xu, L., Sato, I., and Sugiyama, M. Similarity-based classification: Connecting similarity learning to binary classification. *arXiv preprint arXiv:2006.06207*, 2020.

Berthon, A., Han, B., Niu, G., Liu, T., and Sugiyama, M. Confidence scores make instance-dependent label-noise learning possible. *ICML*, 2021.

Boudiaf, M., Rony, J., Ziko, I. M., Granger, E., Pedersoli, M., Piantanida, P., and Ayed, I. B. A unifying mutual information view of metric learning: cross-entropy vs. pairwise losses. In *ECCV*, pp. 548–564. Springer, 2020.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *AAAI*, pp. 1597–1607. PMLR, 2020.

Cheng, J., Liu, T., Ramamohanarao, K., and Tao, D. Learning with bounded instance-and label-dependent label noise. In *ICML*, 2020.

Chou, Y.-T., Niu, G., Lin, H.-T., and Sugiyama, M. Unbiased risk estimators can mislead: A case study of learning with complementary labels. In *ICML*, pp. 1929–1938. PMLR, 2020.

Feng, L., Kaneko, T., Han, B., Niu, G., An, B., and Sugiyama, M. Learning with multiple complementary labels. In *ICML*, pp. 3072–3081. PMLR, 2020.

Gastaldi, X. Shake-shake regularization. *arXiv preprint arXiv:1705.07485*, 2017.

Guo, S., Huang, W., Zhang, H., Zhuang, C., Dong, D., Scott, M. R., and Huang, D. Curriculumnet: Weakly supervised learning from large-scale web images. In *ECCV*, pp. 135–150, 2018.

Han, B., Yao, J., Niu, G., Zhou, M., Tsang, I., Zhang, Y., and Sugiyama, M. Masking: A new perspective of noisy supervision. In *NeurIPS*, pp. 5836–5846, 2018a.

Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I., and Sugiyama, M. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, 2018b.

Han, B., Niu, G., Yu, X., Yao, Q., Xu, M., Tsang, I., and Sugiyama, M. Sigua: Forgetting may make learning with noisy labels more robust. In *ICML*, pp. 4006–4016. PMLR, 2020a.

Han, B., Yao, Q., Liu, T., Niu, G., Tsang, I. W., Kwok, J. T., and Sugiyama, M. A survey of label-noise representation learning: Past, present and future. *arXiv preprint arXiv:2011.04406*, 2020b.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, pp. 770–778, 2016a.

He, K., Zhang, X., Ren, S., and Sun, J. Identity mappings in deep residual networks. In *ECCV*, pp. 630–645. Springer, 2016b.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pp. 9729–9738, 2020.

Hendrycks, D., Mazeika, M., Wilson, D., and Gimpel, K. Using trusted data to train deep networks on labels corrupted by severe noise. In *NeurIPS*, 2018.

Hsu, Y.-C., Lv, Z., and Kira, Z. Learning to cluster in order to transfer across domains and tasks. In *ICLR*, 2018.

Hsu, Y.-C., Lv, Z., Schlosser, J., Odom, P., and Kira, Z. Multi-class classification without multi-class labels. In *ICLR*, 2019.

Hu, W., Li, Z., and Yu, D. Simple and effective regularization methods for training on noisily labeled data with generalization guarantee. In *ICLR*, 2020.

Jiang, L., Zhou, Z., Leung, T., Li, L.-J., and Fei-Fei, L. MentorNet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *ICML*, pp. 2309–2318, 2018.

Kremer, J., Sha, F., and Igel, C. Robust active label correction. In *AISTATS*, pp. 308–316, 2018.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

LeCun, Y. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Li, J., Wong, Y., Zhao, Q., and Kankanhalli, M. S. Learning to learn from noisy labeled data. In *CVPR*, pp. 5051–5059, 2019.

Li, X., Liu, T., Han, B., Niu, G., and Sugiyama, M. Provably end-to-end label-noise learning without anchor points. *ICML*, 2021.

Li, Y., Yang, J., Song, Y., Cao, L., Luo, J., and Li, L.-J. Learning from noisy labels with distillation. In *ICCV*, pp. 1910–1918, 2017.

Liu, T. and Tao, D. Classification with noisy labels by importance reweighting. *IEEE Transactions on pattern analysis and machine intelligence*, 38(3):447–461, 2016.

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

Ma, X., Wang, Y., Houle, M. E., Zhou, S., Erfani, S. M., Xia, S.-T., Wijewickrema, S., and Bailey, J. Dimensionality-driven learning with noisy labels. In *ICML*, pp. 3361–3370, 2018.

Ma, X., Huang, H., Wang, Y., Romano, S., Erfani, S., and Bailey, J. Normalized loss functions for deep learning with noisy labels. In *ICML*, 2020.

Menon, A., Van Rooyen, B., Ong, C. S., and Williamson, B. Learning from corrupted binary labels via class-probability estimation. In *ICML*, pp. 125–134, 2015.

Menon, A. K., Rawat, A. S., Reddi, S. J., and Kumar, S. Can gradient clipping mitigate label noise? *ICLR*, 2020.

Mohri, M., Rostamizadeh, A., and Talwalkar, A. *Foundations of Machine Learning*. MIT Press, 2018.

Natarajan, N., Dhillon, I. S., Ravikumar, P. K., and Tewari, A. Learning with noisy labels. In *NeurIPS*, pp. 1196–1204, 2013.

Northcutt, C. G., Wu, T., and Chuang, I. L. Learning with confident examples: Rank pruning for robust classification with noisy labels. In *UAI*, 2017.

Patrini, G., Rozza, A., Krishna Menon, A., Nock, R., and Qu, L. Making deep neural networks robust to label noise: A loss correction approach. In *CVPR*, pp. 1944–1952, 2017.

Pennington, J., Socher, R., and Manning, C. D. Glove: Global vectors for word representation. In *EMNLP*, pp. 1532–1543, 2014.

Qi, Q., Yan, Y., Wu, Z., Wang, X., and Yang, T. A simple and effective framework for pairwise deep metric learning. *ECCV*, 2019.

Reed, S. E., Lee, H., Anguelov, D., Szegedy, C., Erhan, D., and Rabinovich, A. Training deep neural networks on noisy labels with bootstrapping. In *ICLR*, 2015.

Ren, M., Zeng, W., Yang, B., and Urtasun, R. Learning to reweight examples for robust deep learning. In *ICML*, pp. 4331–4340, 2018.

Scott, C. A rate of convergence for mixture proportion estimation, with application to learning from noisy labels. In *AISTATS*, pp. 838–846, 2015.

Shalev-Shwartz, S. and Ben-David, S. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Tanaka, D., Ikami, D., Yamasaki, T., and Aizawa, K. Joint optimization framework for learning with noisy labels. In *CVPR*, pp. 5552–5560, 2018.

Tanno, R., Saeedi, A., Sankaranarayanan, S., Alexander, D. C., and Silberman, N. Learning from noisy labels by regularized estimation of annotator confusion. In *CVPR*, pp. 11236–11245, 2019.

Tewari, A. and Bartlett, P. L. On the consistency of multiclass classification methods. *JMLR*, 8(May):1007–1025, 2007.

Vahdat, A. Toward robustness against label noise in training deep discriminative neural networks. In *NeurIPS*, pp. 5596–5605, 2017.

Veit, A., Alldrin, N., Chechik, G., Krasin, I., Gupta, A., and Belongie, S. Learning from noisy large-scale datasets with minimal supervision. In *CVPR*, pp. 839–847, 2017.

Wei, H., Feng, L., Chen, X., and An, B. Combating noisy labels by agreement: A joint training method with co-regularization. In *CVPR*, June 2020.

Wu, P., Zheng, S., Goswami, M., Metaxas, D., and Chen, C. A topological filter for learning with label noise. In *NeurIPS*, 2020a.

Wu, S., Xia, X., Liu, T., Han, B., Gong, M., Wang, N., Liu, H., and Niu, G. Multi-class classification from noisy-similarity-labeled data. *arXiv preprint arXiv:2002.06508*, 2020b.

Xia, X., Liu, T., Wang, N., Han, B., Gong, C., Niu, G., and Sugiyama, M. Are anchor points really indispensable in label-noise learning? In *NeurIPS*, pp. 6835–6846, 2019.

Xia, X., Liu, T., Han, B., Gong, M., Yu, J., Niu, G., and Sugiyama, M. Instance correction for learning with open-set noisy labels. *arXiv preprint arXiv:2106.00455*, 2020a.

Xia, X., Liu, T., Han, B., Gong, M., Yu, J., Niu, G., and Sugiyama, M. Sample selection with uncertainty of losses for learning with noisy labels. *arXiv preprint arXiv:2106.00445*, 2020b.

Xia, X., Liu, T., Han, B., Wang, N., Deng, J., Li, J., and Mao, Y. Extended t: Learning with mixed closed-set and open-set noisy labels. *arXiv preprint arXiv:2012.00932*, 2020c.

Xia, X., Liu, T., Han, B., Wang, N., Gong, M., Liu, H., Niu, G., Tao, D., and Sugiyama, M. Part-dependent label noise: Towards instance-dependent label noise. In *NeurIPS*, 2020d.

Xia, X., Liu, T., Han, B., Gong, C., Wang, N., Ge, Z., and Chang, Y. Robust early-learning: Hindering the memorization of noisy labels. In *ICLR*, 2021.

Xiao, T., Xia, T., Yang, Y., Huang, C., and Wang, X. Learning from massive noisy labeled data for image classification. In *CVPR*, pp. 2691–2699, 2015.

Xu, Y., Cao, P., Kong, Y., and Wang, Y. L_dmi: An information-theoretic noise-robust loss function. In *NeurIPS*, 2019.

Yao, Q., Yang, H., Han, B., Niu, G., and Kwok, J. Searching to exploit memorization effect in learning with noisy labels. In *ICML*, 2020a.

Yao, Y., Liu, T., Han, B., Gong, M., Deng, J., Niu, G., and Sugiyama, M. Dual t: Reducing estimation error for transition matrix in label-noise learning. In *NeurIPS*, 2020b.

Yu, X., Liu, T., Gong, M., and Tao, D. Learning with biased complementary labels. In *ECCV*, pp. 68–83, 2018.

Yu, X., Han, B., Yao, J., Niu, G., Tsang, I. W., and Sugiyama, M. How does disagreement benefit co-teaching? In *ICML*, 2019.

Yu, X., Liu, T., Gong, M., Zhang, K., Batmanghelich, K., and Tao, D. Label-noise robust domain adaptation. In *ICML*, pp. 10913–10924. PMLR, 2020.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. In *ICLR*, 2017.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond empirical risk minimization. *ICLR*, 2018a.

Zhang, J., Zhang, T., Dai, Y., Harandi, M., and Hartley, R. Deep unsupervised saliency detection: A multiple noisy labeling perspective. In *CVPR*, pp. 9029–9038, 2018b.

Zhang, Y., Niu, G., and Sugiyama, M. Learning noise transition matrix from only noisy labels via total variation regularization. *ICML*, 2021.

Zhang, Z. and Sabuncu, M. Generalized cross entropy loss for training deep neural networks with noisy labels. In *NeurIPS*, pp. 8778–8788, 2018.

Zheng, S., Wu, P., Goswami, A., Goswami, M., Metaxas, D., and Chen, C. Error-bounded correction of noisy labels. In *ICML*, pp. 11447–11457, 2020.

Zhong, J.-X., Li, N., Kong, W., Liu, S., Li, T. H., and Li, G. Graph convolutional label noise cleaner: Train a plug-and-play action classifier for anomaly detection. In *CVPR*, pp. 1237–1246, 2019.

Zhu, Z., Liu, T., and Liu, Y. A second-order approach to learning with instance-dependent label noise. *CVPR*, 2021.

Appendices

A Proof of Theorem 1

Theorem 1. *Assume that the dataset is balanced (each class has the same amount of instances, and c classes in total), and the noise is class-dependent. Given a class transition matrix T_c, such that $T_{c,ij} = P(\bar{Y} = j|Y = i)$. The elements of the corresponding similarity transition matrix T_s can be calculated as*

$$T_{s,00} = \frac{c^2 - c - \left(\sum_j (\sum_i T_{c,ij})^2 - ||T_c||^2_{\text{Fro}}\right)}{c^2 - c}, \qquad T_{s,01} = \frac{\sum_j (\sum_i T_{c,ij})^2 - ||T_c||^2_{\text{Fro}}}{c^2 - c},$$
$$T_{s,10} = \frac{c - ||T_c||^2_{\text{Fro}}}{c}, \qquad T_{s,11} = \frac{||T_c||^2_{\text{Fro}}}{c}.$$

Proof. Assume each class has n samples. $n^2 T_{c,ij} T_{c,i'j'}$ represents the number the kind of data pairs composed by points of $(\bar{Y} = j|Y = i)$ and $(\bar{Y} = j'|Y = i')$. For the first element $T_{s,00}$, $n^2 \sum_{i \neq i'} T_{c,ij} T_{c,i'j'}$ is the number of data pairs with clean similarity labels $H = 0$, while $n^2 \sum_{i \neq i', j \neq j'} T_{c,ij} T_{c,i'j'}$ is the number of data pairs with clean similarity labels $H = 0$ and noisy similarity labels $\bar{H} = 0$. Thus the proportion of these two terms is exact the $T_{s,00} = P(\bar{H} = 0|H = 0)$. The remaining three elements can be represented in the same way. The primal representations are as follows,

$$T_{s,00} = \frac{\sum_{i \neq i', j \neq j'} T_{c,ij} T_{c,i'j'}}{\sum_{i \neq i'} T_{c,ij} T_{c,i'j'}}, \qquad T_{s,01} = \frac{\sum_{i \neq i', j = j'} T_{c,ij} T_{c,i'j'}}{\sum_{i \neq i'} T_{c,ij} T_{c,i'j'}},$$
$$T_{s,10} = \frac{\sum_{i = i', j \neq j'} T_{c,ij} T_{c,i'j'}}{\sum_{i = i'} T_{c,ij} T_{c,i'j'}}, \qquad T_{s,11} = \frac{\sum_{i = i', j = j'} T_{c,ij} T_{c,i'j'}}{\sum_{i = i'} T_{c,ij} T_{c,i'j'}}.$$

Further, note that

$$\sum_{i = i'} T_{c,i,j} T_{c,i',j'} = \sum_{i,j,j'} T_{c,i,j} T_{c,i,j'} = \sum_i (\sum_j T_{c,i,j})(\sum_{j'} T_{c,i,j'}) = c,$$
$$\sum_{i \neq i'} T_{c,i,j} T_{c,i',j'} = \sum_{i \neq i', j, j'} T_{c,i,j} T_{c,i',j'} = \sum_{i \neq i'} (\sum_j T_{c,i,j})(\sum_{j'} T_{c,i,j'}) = (c-1)c,$$
$$\sum_{i = i', j = j'} T_{c,ij} T_{c,i'j'} = ||T_c||^2_{\text{Fro}},$$
$$\sum_{i \neq i', j = j'} T_{c,ij} T_{c,i'j'} = \sum_j (\sum_i T_{c,ij})^2 - ||T_c||^2_{\text{Fro}}.$$

Substituting above equations to the primal representations, we have the Theorem 1 proved. □

B *Pointwise* implies *pairwise*

For an invertible T_c, denote by $\mathbf{v}_j$ the j-th column of T_c and $\mathbf{1}$ the all-one vector. Then,

$$\sum_j (\sum_i T_{c,ij})^2 = \sum_j \langle \mathbf{v}_j, \mathbf{1} \rangle^2 \leq \sum_j ||\mathbf{v}_j||^2 ||\mathbf{1}||^2 = c||T_c||_{\text{Fro}}^2,$$

where we use the Cauchy–Schwarz inequality [Steele, 2004] in the second step. Further, we have

$$\begin{aligned}
T_{s,11} + T_{s,00} &= \frac{||T_c||_{\text{Fro}}^2}{c} + \frac{c^2 - c - \left(\sum_j (\sum_i T_{c,ij})^2 - ||T_c||_{\text{Fro}}^2\right)}{c^2 - c} \\
&= \frac{(c-1)||T_c||_{\text{Fro}}^2 + c^2 - c - \left(\sum_j (\sum_i T_{c,ij})^2 - ||T_c||_{\text{Fro}}^2\right)}{c^2 - c} \\
&= \frac{(c-1)||T_c||_{\text{Fro}}^2 + c^2 - c - \left(\sum_j \langle \mathbf{v}_j, \mathbf{1} \rangle^2 - ||T_c||_{\text{Fro}}^2\right)}{c^2 - c} \\
&\geq \frac{(c-1)||T_c||_{\text{Fro}}^2 + c^2 - c - \left(c||T_c||_{\text{Fro}}^2 - ||T_c||_{\text{Fro}}^2\right)}{c^2 - c} \\
&= 1.
\end{aligned}$$

Thus the learnability of the pointwise classification implies the learnability of the reduced pairwise classification.

C Proof of Theorem 2

Theorem 2. *Assume that the dataset is balanced (each class has the same amount of samples), and the noise is class-dependent. When the number of classes $c \geq 8$, the noise rate of noisy similarity labels is lower than that of the noisy class labels.*

Proof. Assume each class has n points. As we state in the proof of Theorem 1, the number of data pairs with clean similarity labels $H = 0$ and noisy similarity labels $\bar{H} = 0$ is $n^2 \sum_{i \neq i', j \neq j'} T_{c,ij} T_{c,i'j'}$. We denote it by N_{00}. Similarly, we have,

$$\begin{aligned}
N_{00} &= n^2 \sum_{i \neq i', j \neq j'} T_{c,ij} T_{c,i'j'}, \qquad & N_{01} &= n^2 \sum_{i \neq i', j = j'} T_{c,ij} T_{c,i'j'}, \\
N_{10} &= n^2 \sum_{i = i', j \neq j'} T_{c,ij} T_{c,i'j'}, \qquad & N_{11} &= n^2 \sum_{i = i', j = j'} T_{c,ij} T_{c,i'j'}.
\end{aligned}$$

The noise rate is the proportion of the number of noisy labels to the number of total labels.

Assume that the number of classes is c. We have

$$S_{noise} = \frac{N_{01} + N_{10}}{N_{00} + N_{01} + N_{10} + N_{11}} = \frac{N_{01} + N_{10}}{c^2 n^2},$$
$$C_{noise} = \frac{n \sum_{i \neq j} T_{c,ij}}{cn}.$$

Let S_{noise} minus C_{noise}, we have

$$\begin{aligned} S_{noise} - C_{noise} &= \frac{n^2 \sum_{i \neq i', j = j'} T_{c,ij} T_{c,i'j'} + n^2 \sum_{i = i', j \neq j'} T_{c,ij} T_{c,i'j'}}{c^2 n^2} - \frac{n \sum_{i \neq j} T_{c,ij}}{cn} \\ &= \frac{\sum_{i \neq i', j = j'} T_{c,ij} T_{c,i'j'} + \sum_{i = i', j \neq j'} T_{c,ij} T_{c,i'j'} - c \sum_{i \neq j} T_{c,ij}}{c^2}. \end{aligned}$$

Let $A = \sum_{i \neq i', j = j'} T_{c,ij} T_{c,i'j'} + \sum_{i = i', j \neq j'} T_{c,ij} T_{c,i'j'} - c \sum_{i \neq j} T_{c,ij}$, we have

$$\begin{aligned} A &= \sum_{i \neq i', j = j'} T_{c,ij} T_{c,i'j'} + \sum_{i = i', j \neq j'} T_{c,ij} T_{c,i'j'} - c \sum_{i \neq j} T_{c,ij} \\ &= \sum_{i \neq i', j = j'} T_{c,ij} T_{c,i'j'} + \sum_{i = i', j \neq j'} T_{c,ij} T_{c,i'j'} - c(\sum_{i,j} T_{c,ij} - \sum_{i=j} T_{c,ij}) \\ &= \sum_{i \neq i', j = j'} T_{c,ij} T_{c,i'j'} + \sum_{i = i', j \neq j'} T_{c,ij} T_{c,i'j'} - c^2 + c \sum_{i=j} T_{c,ij}. \end{aligned}$$

The second equation holds because the row sum of T_c is 1.

For the first term $\sum_{i \neq i', j = j'} T_{c,ij} T_{c,i'j'}$, notice that:

$$\begin{aligned} \sum_{i \neq i', j = j'} T_{c,ij} T_{c,i'j'} &= \sum_j \sum_i T_{c,ij} (\sum_{i' \neq i} T_{c,i'j}) \\ &= \sum_j \sum_i T_{c,ij} (\sum_{i' \neq i} T_{c,i'j} + T_{c,ij} - T_{c,ij}) \\ &= \sum_j \sum_i T_{c,ij} (\sum_{i'} T_{c,i'j} - T_{c,ij}) \\ &= \sum_j \sum_i T_{c,ij} (S_j - T_{c,ij}) \qquad (S_j \text{ is the column sum of the } j-th \text{ column}) \\ &= \sum_j \sum_i T_{c,ij} S_j - T_{c,ij}^2 \\ &= \sum_j S_j \sum_i T_{c,ij} - \sum_j \sum_i T_{c,ij}^2 \\ &= \sum_j S_j^2 - \sum_j \sum_i T_{c,ij}^2. \end{aligned} \tag{1}$$

Due to the symmetry of i and j, for the second term $\sum_{i=i',j\neq j'} T_{c,ij}T_{c,i'j'}$, we have

$$\begin{aligned}\sum_{i=i',j\neq j'} T_{c,ij}T_{c,i'j'} &= \sum_j \sum_i T_{c,ij}(R_i - T_{c,ij}) \qquad (R_i \text{ is the row sum of the } i-th \text{ row, and } R_i = 1)\\ &= \sum_j \sum_i T_{c,ij} - T_{c,ij}^2 \\ &= c - \sum_j \sum_i T_{c,ij}^2. \end{aligned} \tag{2}$$

Therefore, substituting Equation (1) and (2) into A, we have

$$A = \sum_j S_j^2 - \sum_j \sum_i T_{c,ij}^2 + c - \sum_j \sum_i T_{c,ij}^2 - c^2 + c\sum_{i=j} T_{c,ij}.$$

To prove $S_{noise} - C_{noise} \leq 0$ is equivalent to prove $A \leq 0$.

Let $M = c^2 - c$, $N = \sum_j S_j^2 - 2\sum_j \sum_i T_{ij}^2 + c\sum_{i=j} T_{ij}$ (we drop the subscript c in $T_{c,ij}$), and $A = N - M$. Now we utilize the Adjustment method [Su and Xiong, 2015] to scale N. For every iteration, we denote the original N by N_o, and the adjusted N by N_a.

Since $c \geq 8$, there can not exist three columns with column sum bigger than $c/2 - 1$. Otherwise, the sum of the three columns will be bigger than c, which is impossible because the sum of the whole matrix is c.

Therefore, first, we assume that the $j, k-th$ columns have column sum bigger than $c/2 - 1$. Then, for the row i, we add the elements l, which are not in $j, k-th$ columns, to the diagonal element. We have

$$\begin{aligned} N_a - N_o &= (S_i + T_{il})^2 + (S_l + T_{il})^2 + cT_{il} - 2(T_{ii} + T_{il})^2 - S_i^2 - S_l^2 + 2(T_{ii}^2 + T_{il}^2) \\ &= T_{il}(2T_{il} + 2S_i - 2S_l + c - 4T_{ii}) \\ &\geq T_{il}(2T_{il} - 2S_l + c - 2T_{ii}) \qquad (\because S_i \geq T_{ii}) \\ &> T_{il}(2T_{il} - c + 2 + c - 2T_{ii}) \qquad (\because S_l < c/2 - 1) \\ &\geq 0. \qquad (\because T_{ii} \leq 1) \end{aligned}$$

We do such adjustment to every rows, then N_a is getting bigger and the adjusted matrix will only have values on diagonal elements and the $j, k-th$ columns. Since the diagonal elements are dominant in the row, $S_j + S_k < 2c/3 + 2/3$ (because for $i \neq j, k, T_{ij} + T_{ik} < 2/3$).

Assume that the column sum of $k-th$ column is no bigger than that of the $j-th$ column, and thus $S_k < c/3 + 1/3$. Then, for a row i, we add the T_{ik} to T_{ii}. We have

$$\begin{aligned} N_a - N_o &= (S_i + T_{ik})^2 + (S_k + T_{ik})^2 + cT_{ik} - 2(T_{ii} + T_{ik})^2 - S_i^2 - S_k^2 + 2(T_{ii}^2 + T_{ik}^2) \\ &= T_{ik}(2T_{ik} + 2S_i - 2S_k + c - 4T_{ii}) \\ &\geq T_{ik}(2T_{ik} - 2S_k + c - 2T_{ii}) \qquad (\because S_i \geq T_{ii}) \\ &> T_{ik}(2T_{ik} + c/3 - 2/3 - 2T_{ii}) \qquad (\because S_k < c/3 + 1/3) \\ &\geq 0. \qquad (\because c \geq 8, \text{ and } T_{ii} \leq 1) \end{aligned}$$

We do such adjustment to every rows, then N_a is getting bigger and the adjusted matrix will only have values on diagonal elements and the $j-th$ column, which is called final matrix.

Note that if there is only one column with a column sum bigger than $c/2-1$, we can adjust the rest $c-1$ columns as above and then obtain the final matrix as well. If there is no column with a column sum bigger than $c/2-1$, we can adjust all the elements as above and then obtain a *unit matrix*. For the unit matrix, $A = N - M < N_a - M = 0$, the Theorem 2 is proved.

Now we process the final matrix. For simplification, we assume $j = 0$ in the final matrix. We denote the T_{ij} by b_i and T_{ii} by a_i, for $i = \{1, \ldots, c-1\}$. We have

$$\begin{aligned}
N_a &= \sum_i a_i^2 + (1 + \sum_i b_i)^2 + c(\sum_i a_i + 1) - 2(\sum_i a_i^2 + \sum_i b_i^2 + 1) \\
&= (1 + \sum_i b_i)^2 + c\sum_i a_i + c - \sum_i a_i^2 - 2\sum_i b_i^2 - 2 \\
&= 1 + (\sum_i b_i)^2 + 2\sum_i b_i + c\sum_i a_i + c - \sum_i a_i^2 - 2\sum_i b_i^2 - 2 \\
&= (\sum_i b_i)^2 + 2\sum_i b_i - 2\sum_i b_i^2 + c\sum_i a_i - \sum_i a_i^2 + c - 1 \\
&= (\sum_i b_i)^2 + 2\sum_i b_i - 2\sum_i b_i^2 + c\sum_i (1 - b_i) - \sum_i (1 - b_i)^2 + c - 1 \\
&= (\sum_i b_i)^2 + 2\sum_i b_i - 2\sum_i b_i^2 + c^2 - c - c\sum_i b_i - \sum_i (1 - 2b_i + b_i^2) + c - 1 \\
&= (\sum_i b_i)^2 + 4\sum_i b_i - 3\sum_i b_i^2 - c\sum_i b_i + c^2 - c.
\end{aligned}$$

Now we prove $A = N - M \leq N_a - M \leq 0$. Note that

$$\begin{aligned}
N_a - M &= (\sum_i b_i)^2 + 4\sum_i b_i - 3\sum_i b_i^2 - c\sum_i b_i \\
&= (\sum_i b_i)^2 + 3\sum_i b_i - 3\sum_i b_i^2 - (c-1)\sum_i b_i \\
&= (\sum_i b_i)^2 + 3\sum_i b_i - 3\sum_i b_i^2 - (\sum_i (1 - b_i) + \sum_i b_i)\sum_i b_i \\
&= 3\sum_i b_i - 3\sum_i b_i^2 - \sum_i (1 - b_i)\sum_i b_i \\
&= 3\sum_i b_i(1 - b_i) - \sum_i (1 - b_i)\sum_i b_i.
\end{aligned}$$

According to the rearrangement inequality[Hardy et al., 1952], we have

$$\sum_i (1 - b_i)\sum_i b_i \geq (c-1)\sum_i b_i(1 - b_i).$$

Note that $c \geq 8$, thus $3\sum_i b_i(1-b_i) - \sum_i(1-b_i)\sum_i b_i \leq 0$, and $A \leq 0$. Therefore $S_{noise} - C_{noise} \leq 0$, and the equation holds if and only if the noise rate is 0 or every instances have the same noisy class label (i.e., there is one column in the T_c, of which every elements are 1, and the rest elements of the T_c are 0). Above two extreme situations are not considered in this paper. Namely, the noise rate of the noisy similarity labels is lower than that of the noisy class labels. Theorem 2 is proved. □

D Implementation of Class2Simi with *Reweight*

The expected risk for clean pairwise data is

$$R(f) = E_{(X_i,X_j,H_{ij})\sim\mathcal{D}}[\ell(\langle f(X_i), f(X_j)\rangle, H_{ij})],$$

where

$$\begin{aligned}\ell(\langle f(X_i), f(X_j)\rangle, H_{ij}) = &-\sum_{i,j} H_{ij}\log(\langle f(X_i), f(X_j)\rangle) + (1-H_{ij})\log(1-\langle f(X_i), f(X_j)\rangle),\\ &-\sum_{i,j} H_{ij}\log\hat{S}_{ij} + (1-H_{ij})\log(1-\hat{S}_{ij}).\end{aligned}$$

Here, we employ the *importance reweighting* technique to build a *risk-consistent* algorithms. Specifically,

$$\begin{aligned}R(f) &= E_{(X_i,X_j,H_{ij})\sim\mathcal{D}}[\ell(\langle f(X_i), f(X_j)\rangle, H_{ij})]\\ &= \int_{(x_i,x_j)}\sum_k P_{\mathcal{D}}(X_i=x_i, X_j=x_j, H_{ij}=k)\ell(\langle f(X_i), f(X_j)\rangle, H_{ij})d(x_i,x_j)\\ &= \int_{(x_i,x_j)}\sum_k P_{\mathcal{D}_\rho}(X_i, X_j, \bar{H}_{ij}=k)\frac{P_{\mathcal{D}}(X_i, X_j, H_{ij}=k)}{P_{\mathcal{D}_\rho}(X_i, X_j, \bar{H}_{ij}=k)}\ell(\langle f(X_i), f(X_j)\rangle, H_{ij}=k)d(x_i,x_j)\\ &= \int_{(x_i,x_j)}\sum_k P_{\mathcal{D}_\rho}(X_i, X_j, \bar{H}_{ij}=k)\frac{P_{\mathcal{D}}(H_{ij}=k|X_i, X_j)}{P_{\mathcal{D}_\rho}(\bar{H}_{ij}=k|X_i, X_j)}\ell(\langle f(X_i), f(X_j)\rangle, H_{ij}=k)d(x_i,x_j)\\ &= E_{(X_i,X_j,\bar{H}_{ij})\sim\mathcal{D}_\rho}[\bar{\ell}(\langle f(X_i), f(X_j)\rangle, \bar{H}_{ij})],\end{aligned}$$

where $\mathcal{D}$ denotes the distribution of clean data; $\mathcal{D}$ denotes the distribution of noisy data, and

$$\bar{\ell}(\langle f(X_i), f(X_j)\rangle, \bar{H}_{ij}) = \frac{P_{\mathcal{D}}(H_{ij}=\bar{H}_{ij}|X_i, X_j)}{P_{\mathcal{D}_\rho}(\bar{H}_{ij}|X_i, X_j)}\ell(\langle f(X_i), f(X_j)\rangle, \bar{H}_{ij}).$$

Empirically, as shown in Figrue 1, we use $\hat{S}_{ij} = f(X_i)^\top f(X_j)$ to measure the similarity of two points in a pair. $P(H_{ij}=1|X_i, X_j)$ and $P(H_{ij}=0|X_i, X_j)$ are approximated by

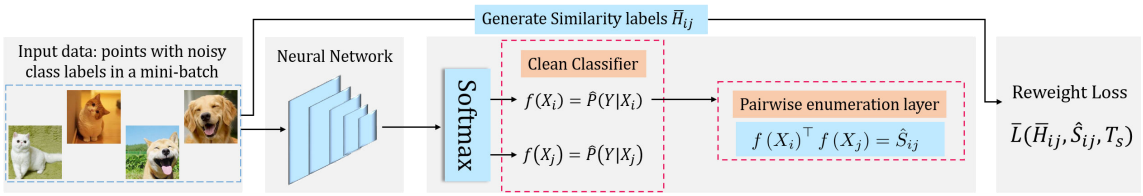


Figure 1: Pipeline of Class2Simi with *Reweight*.

$\hat{S}_{ij}$ and $1 - \hat{S}_{ij}$, respectively. Then $P\left(\bar{H}_{ij} \mid X_i, X_j\right)$ can be approximated according to $P\left(\bar{H}_{ij} \mid X_i, X_j\right) = T_s^\top P\left(H_{ij} \mid X_i, X_j\right)$. Thus a risk-consistent estimator can be built:

$$R_n(f) = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha \ell(\langle f(X_i), f(X_j) \rangle, \bar{H}_{ij}),$$

where

$$\alpha = \left\{ \bar{H}_{ij} \frac{\hat{S}_{ij}}{T_{s,11}\hat{S}_{ij} + T_{s,01}(1 - \hat{S}_{ij})} + (1 - \bar{H}_{ij}) \frac{1 - \hat{S}_{ij}}{T_{s,10}\hat{S}_{ij} + T_{s,00}(1 - \hat{S}_{ij})} \right\}.$$

E Proof of Theorem 3

Theorem 3. *Assume the parameter matrices $W_1, \ldots, W_d$ have Frobenius norm at most $M_1, \ldots, M_d$, and the activation functions are 1-Lipschitz, positive-homogeneous, and applied element-wise (such as the ReLU). Assume the transition matrix is given, and the instances X are upper bounded by B, i.e., $\|X\| \leq B$ for all X, and the loss function ℓ is upper bounded by M. Then, for any $\delta > 0$, with probability at least $1 - \delta$,*

$$R(\hat{f}) - R_n(\hat{f}) \leq \frac{(T_{s,11} - T_{s,01}) 2Bc(\sqrt{2d \log 2} + 1) \Pi_{i=1}^{d} M_i}{T_{s,11}\sqrt{n}} + M\sqrt{\frac{\log 1/\delta}{2n}}. \tag{3}$$

Proof. We have defined

$$R(f) = E_{(X_i, X_j, \bar{Y}_i, \bar{Y}_j, \bar{H}_{ij}, T_s) \sim \mathcal{D}_\rho}[\ell(f(X_i), f(X_j), T_s, \bar{H}_{ij})], \tag{4}$$

and

$$R_n(f) = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \ell(f(X_i), f(X_j), T_s, \bar{H}_{ij}), \tag{5}$$

where n is training sample size of the noisy data.

First, we bound the generalization error with Rademacher complexity [Bartlett and Mendelson, 2002].

Theorem 4 (Bartlett and Mendelson [2002]). *Let the loss function be upper bounded by M. Then, for any $\delta > 0$, with the probability $1 - \delta$, we have*

$$\sup_{f \in \mathcal{F}} |R(f) - R_n(f)| \leq 2\mathfrak{R}_n(\ell \circ \mathcal{F}) + M\sqrt{\frac{\log 1/\delta}{2n}}, \tag{6}$$

where $\mathfrak{R}_n(\ell \circ \mathcal{F})$ is the Rademacher complexity defined by

$$\mathfrak{R}_n(\ell \circ \mathcal{F}) = E\left[\sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i \ell(f(X_i), f(X_j), T_s, \bar{H}_{ij})\right], \tag{7}$$

and $\{\sigma_1, \cdots, \sigma_n\}$ are Rademacher variables uniformly distributed from $\{-1, 1\}$.

Before further upper bound the Rademacher complexity $\mathfrak{R}_n(\ell \circ \mathcal{F})$, we discuss the special loss function and its *Lipschitz continuity* w.r.t $h_k(X_i), k = \{1, \ldots, c\}$.

Lemma 1. *Given similarity transition matrix T_s, loss function $\ell(f(X_i), f(X_j), T_s, \bar{H}_{ij})$ is μ-Lipschitz with respect to $h_k(X_i), k = \{1, \ldots, c\}$, and $\mu = (T_{s,11} - T_{s,01})/T_{s,11}$*

$$\left|\frac{\partial \ell(f(X_i), f(X_j), T_s, \bar{H}_{ij})}{\partial h_k(X_i)}\right| < \frac{T_{s,11} - T_{s,01}}{T_{s,11}}. \tag{8}$$

Detailed proof of Lemma 1 can be found in Section E.1.

Lemma 1 shows that the loss function is μ-Lipschitz with respect to $h_k(X_i), k = \{1, \ldots, c\}$.

Based on Lemma 1, we can further upper bound the Rademacher complexity $\mathfrak{R}_n(\ell \circ \mathcal{F})$ by the following lemma.

Lemma 2. *Given similarity transition matrix T_s and assume that loss function $\ell(f(X_i), f(X_j), T_s, \bar{H}_{ij})$ is μ-Lipschitz with respect to $h_k(X_i), k = \{1, \ldots, c\}$, we have*

$$\begin{aligned}\mathfrak{R}_n(\ell \circ \mathcal{F}) &= E\left[\sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i \ell(f(X_i), f(X_j), T_s, \bar{H}_{ij})\right] \\ &\leq \mu c E\left[\sup_{h \in H} \frac{1}{n} \sum_{i=1}^{n} \sigma_i h(X_i)\right],\end{aligned} \tag{9}$$

where H is the function class induced by the deep neural network.

Detailed proof of Lemma 2 can be found in Section E.2.

The right-hand side of the above inequality, indicating the hypothesis complexity of deep neural networks and bounding the Rademacher complexity, can be bounded by the following theorem.

Theorem 5. *[Golowich et al., 2018] Assume the Frobenius norm of the weight matrices $W_1, \ldots, W_d$ are at most $M_1, \ldots, M_d$. Let the activation functions be 1-Lipschitz, positive-homogeneous, and applied element-wise (such as the ReLU). Let X is upper bounded by B, i.e., for any X, $\|X\| \leq B$. Then,*

$$E\left[\sup_{h\in H}\frac{1}{n}\sum_{i=1}^{n}\sigma_i h(X_i)\right] \leq \frac{B(\sqrt{2d\log 2}+1)\Pi_{i=1}^{d}M_i}{\sqrt{n}}. \tag{10}$$

Combining Lemma 1,2, and Theorem 4, 5, Theorem 3 is proved. □

E.1 Proof of Lemma 1

Recall that

$$\begin{aligned}\ell(f(X_i), f(X_j), T_s, \bar{H}_{ij}=1) &= -\log(\hat{\bar{S}}_{ij}) \\ &= -\log(\hat{S}_{ij}\times T_{s,11} + (1-\hat{S}_{ij})\times T_{s,01}) \\ &= -\log(f(X_i)^\top f(X_j)\times T_{s,11} + (1-f(X_i)^\top f(X_j))\times T_{s,01}),\end{aligned} \tag{11}$$

where

$$\begin{aligned}f(X_i) &= [f_1(X_i), \ldots, f_c(X_i)]^\top \\ &= \left[\left(\frac{\exp(h_1(X))}{\sum_{k=1}^{c}\exp(h_k(X))}\right), \ldots, \left(\frac{\exp(h_c(X))}{\sum_{k=1}^{c}\exp(h_k(X))}\right)\right]^\top.\end{aligned} \tag{12}$$

Take the derivative of $\ell(f(X_i), f(X_j), T_s, \bar{H}_{ij}=1)$ w.r.t. $h_k(X_i)$, we have

$$\frac{\partial\ell(f(X_i), f(X_j), T_s, \bar{H}_{ij}=1)}{\partial h_k(X_i)} = \frac{\partial\ell(f(X_i), f(X_j), T_s, \bar{H}_{ij}=1)}{\partial\hat{\bar{S}}_{ij}}\left[\frac{\partial f(X_i)}{\partial h_k(X_i)}\right]^\top\frac{\partial\hat{\bar{S}}_{ij}}{\partial f(X_i)}, \tag{13}$$

where

$$\begin{aligned}&\frac{\partial\ell(f(X_i), f(X_j), T_s, \bar{H}_{ij}=1)}{\partial\hat{\bar{S}}_{ij}} = -\frac{1}{f(X_i)^\top f(X_j)\times T_{s,11} + (1-f(X_i)^\top f(X_j))\times T_{s,01}}, \\ &\frac{\partial\hat{\bar{S}}_{ij}}{\partial f(X_i)} = f(X_j)\times T_{s,11} - f(X_j)\times T_{s,01}, \\ &\frac{\partial f(X_i)}{\partial h_k(X_i)} = f'(X_i) = [f_1'(X_i), \ldots, f_c'(X_i)]^\top.\end{aligned}$$

Note that the derivative of the softmax function has some properties, i.e., if $m \neq k$, $f_m'(X_i) = -f_m(X_i)f_k(X_i)$ and if $m = k$, $f_k'(X_i) = (1-f_k(X_i))f_k(X_i)$.

We denote by $Vector_m$ the $m-th$ element in $Vector$ for those complex vectors. Because $0 < f_m(X_i) < 1, \forall m \in \{1, \ldots, c\}$, we have

$$f'_m(X_i) \leq |f'_m(X_i)| < f_m(X_i), \qquad \forall m \in \{1, \ldots, c\}; \tag{14}$$

$$f'(X_i)^\top f(X_j) < f(X_i)^\top f(X_j). \tag{15}$$

Therefore,

$$\begin{aligned}
&\left|\frac{\partial \ell(f(X_i), f(X_j), T_s, \bar{H}_{ij}=1)}{\partial h_k(X_i)}\right| = \left|\frac{\partial \ell(f(X_i), f(X_j), T_s, \bar{H}_{ij}=1)}{\partial \hat{\bar{S}}_{ij}} \left[\frac{\partial f(X_i)}{\partial h_k(X_i)}\right]^\top \frac{\partial \hat{\bar{S}}_{ij}}{\partial f(X_i)}\right| \\
&= \left|\frac{f'(X_i)^\top f(X_j) \times T_{s,11} - f'(X_i)^\top f(X_j) \times T_{s,01}}{f(X_i)^\top f(X_j) \times T_{s,11} + (1 - f(X_i)^\top f(X_j)) \times T_{s,01}}\right| \\
&< \left|\frac{f(X_i)^\top f(X_j) \times T_{s,11} - f(X_i)^\top f(X_j) \times T_{s,01}}{f(X_i)^\top f(X_j) \times T_{s,11} + (1 - f(X_i)^\top f(X_j)) \times T_{s,01}}\right| \\
&< \left|\frac{T_{s,11} - T_{s,01}}{T_{s,11}}\right| \\
&= \frac{T_{s,11} - T_{s,01}}{T_{s,11}}.
\end{aligned} \tag{16}$$

The second inequality holds because of $T_{s,11} > T_{s,01}$ (Detailed proof can be found in Section E.1.1) and Equation (20). The third inequality holds because of $f(X_i)^\top f(X_j) < 1$.

Similarly, we can prove

$$\left|\frac{\partial \ell(f(X_i), f(X_j), T_s, \bar{H}_{ij}=0)}{\partial h_k(X_i)}\right| < \frac{T_{s,11} - T_{s,01}}{T_{s,11}}. \tag{17}$$

Combining Equation (16) and Equation (17), we obtain

$$\left|\frac{\partial \ell(f(X_i), f(X_j), T_s, \bar{H}_{ij})}{\partial h_k(X_i)}\right| < \frac{T_{s,11} - T_{s,01}}{T_{s,11}}. \tag{18}$$

E.1.1 Proof of $T_{s,11} > T_{s,01}$

As we mentioned in Section C, we have,

$$\begin{aligned}
N_{00} &= n^2 \sum_{i \neq i', j \neq j'} T_{c,ij} T_{c,i'j'}, \qquad N_{01} = n^2 \sum_{i \neq i', j = j'} T_{c,ij} T_{c,i'j'}, \\
N_{10} &= n^2 \sum_{i = i', j \neq j'} T_{c,ij} T_{c,i'j'}, \qquad N_{11} = n^2 \sum_{i = i', j = j'} T_{c,ij} T_{c,i'j'}, \\
T_{s,01} &= \frac{N_{01}}{N_{00} + N_{01}}, \qquad T_{s,11} = \frac{N_{11}}{N_{10} + N_{11}}, \\
T_{s,11} - T_{s,01} &= \frac{N_{11}N_{00} + N_{11}N_{01} - N_{01}N_{10} - N_{01}N_{11}}{(N_{00} + N_{01})(N_{10} + N_{11})}.
\end{aligned}$$

Let us review the definition of similarity labels: if two instances belong to the same class, they will have similarity label $S = 1$, otherwise $S = 0$. That is to say, for a k-class dataset, only $\frac{1}{k}$ of similarity data has similarity labels $S = 1$, and the rest $1 - \frac{1}{k}$ has similarity labels $S = 0$. We denote the number of data with similarity labels $S = 1$ by N_1, otherwise N_0. Therefore, for the balanced dataset with n samples of each class, $N_1 = cn^2$, and $N_0 = c(c-1)n^2$. Let $A = T_{s,11} - T_{s,01}$, we have

$$\begin{aligned} A &= N_{11}N_{00} - N_{01}N_{10} \\ &= N_{11}N_{00} - (N_0 - N_{00})(N_1 - N_{11}) \\ &= N_{11}N_{00} - N_0N_1 - N_{11}N_{00} + N_{11}N_0 + N_1N_{00} \\ &= N_{11}N_0 - N_{01}N_1 \\ &= c(c-1)n^2N_{11} - cn^2N_{01} \\ &> 0. \end{aligned}$$

The last equation holds because of $(c-1)N_{11} - N_{01} > 0$ according to the rearrangement inequality [Hardy et al., 1952].

E.2 Proof of Lemma 2

$$\begin{aligned} & E\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i \ell(f(X_i), f(X_j), T_s, \bar{H}_{ij})\right] \\ &= E\left[\sup_{g} \frac{1}{n}\sum_{i=1}^{n} \sigma_i \ell(f(X_i), f(X_j), T_s, \bar{H}_{ij})\right] \\ &= E\left[\sup_{\text{argmax}\{h_1,\ldots,h_c\}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i \ell(f(X_i), f(X_j), T_s, \bar{H}_{ij})\right] \\ &= E\left[\sup_{\max\{h_1,\ldots,h_c\}} \frac{1}{n}\sum_{i=1}^{n} \sigma_i \ell(f(X_i), f(X_j), T_s, \bar{H}_{ij})\right] \\ &\leq E\left[\sum_{k=1}^{c} \sup_{h_k\in H} \frac{1}{n}\sum_{i=1}^{n} \sigma_i \ell(f(X_i), f(X_j), T_s, \bar{H}_{ij})\right] \\ &= \sum_{k=1}^{c} E\left[\sup_{h_k\in H} \frac{1}{n}\sum_{i=1}^{n} \sigma_i \ell(f(X_i), f(X_j), T_s, \bar{H}_{ij})\right] \\ &\leq \mu c E\left[\sup_{h_k\in H} \frac{1}{n}\sum_{i=1}^{n} \sigma_i h_k(X_i)\right] \\ &= \mu c E\left[\sup_{h\in H} \frac{1}{n}\sum_{i=1}^{n} \sigma_i h(X_i)\right], \end{aligned}$$

where the first three equations hold because given T_s, f and $\max\{h_1, \ldots, h_c\}$ give the same constraint on $h_j(X_i), j = \{1, \ldots, c\}$; the sixth inequality holds because of the Talagrand Contraction Lemma [Ledoux and Talagrand, 2013].

F Further details on experiments

F.1 Network structure and optimization

Note that for *CIFAR-10*, we use ResNet-26 with shake-shake regularization [Gastaldi, 2017] **except** the experiment on noisy T_c in Figure 4, where we use ResNet-32 with pre-activation [He et al., 2016] for shorter training time. In stage 1, We use the same optimization method as *Forward* to learn the transition matrix $\hat{T}_c$. In stage 2, we use Adam optimizer with an initial learning rate 0.001. On *MNIST*, the batch size is 128 and the learning rate decays every 5 epochs by a factor of 0.1 with 30 epochs in total. On *CIFAR-10*, the batch size is 512 and the learning rate decays every 40 epochs by a factor of 0.1 with 200 epochs in total. On *CIFAR-100*, the batch size is 512 and the learning rate decays every 40 epochs by a factor of 0.1 with 120 epochs in total. On *News20*, the batch size is 128 and the learning rate decays every 5 epochs by a factor of 0.1 with 30 epochs in total. On *Clothing1M**, the batch size is 32 and the learning rate drops every 5 epochs by a factor of 0.1 with 10 epochs in total.

F.2 Symmetric and asymmetric noise settings

Symmetric noise setting is defined as follow, where c is the number of classes.

$$\text{Sym-}\rho\text{:} \quad T = \begin{bmatrix} 1-\rho & \frac{\rho}{C-1} & \cdots & \frac{\rho}{C-1} & \frac{\rho}{C-1} \\ \frac{\rho}{C-1} & 1-\rho & \frac{\rho}{C-1} & \cdots & \frac{\rho}{C-1} \\ \vdots & & \ddots & & \vdots \\ \frac{\rho}{C-1} & \cdots & \frac{\rho}{C-1} & 1-\rho & \frac{\rho}{C-1} \\ \frac{\rho}{C-1} & \frac{\rho}{C-1} & \cdots & \frac{\rho}{C-1} & 1-\rho \end{bmatrix}. \tag{19}$$

The asymmetric noise setting is set as follow,

Listing 1: Asymmetric noise (transition matrix) generation.

```
def AsymTransitionMatrixGenerate(NoiseRate=0.3,
    NumClasses=10, seed=1):
    np.random.seed(seed)
    t = np.random.rand(NumClasses, NumClasses)
    i = np.eye(NumClasses)
    t = t + Coef * NumClasses * i
    for a in range(NumClasses):
    t[a] = t[a] / t[a].sum()
return t
```

Coef is set to 1.70, 1.20, 0.60, 0.24 at the rate 0.2, 0.3, 0.4, 0.6, respectively.

References

Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002. 7, 8

Xavier Gastaldi. Shake-shake regularization. *arXiv preprint arXiv:1705.07485*, 2017. 12

Noah Golowich, Alexander Rakhlin, and Ohad Shamir. Size-independent sample complexity of neural networks. In *COLT*, 2018. 9

Godfrey Harold Hardy, John Edensor Littlewood, George Pólya, György Pólya, DE Littlewood, et al. *Inequalities*. Cambridge university press, 1952. 5, 11

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV*, pages 630–645. Springer, 2016. 12

Michel Ledoux and Michel Talagrand. *Probability in Banach Spaces: isoperimetry and processes*. Springer Science & Business Media, 2013. 12

J Michael Steele. *The Cauchy-Schwarz master class: an introduction to the art of mathematical inequalities*. Cambridge University Press, 2004. 2

Yong Su and Bin Xiong. *Methods and Techniques for Proving Inequalities: In Mathematical Olympiad and Competitions*, volume 11. World Scientific Publishing Company, 2015. 4