Clusterability as an Alternative to Anchor Points When Learning with Noisy Labels

Zhaowei Zhu [1] **Yiwen Song** [2] **Yang Liu** [1]

Abstract

The label noise transition matrix, characterizing the probabilities of a training instance being wrongly annotated, is crucial to designing popular solutions to learning with noisy labels. Existing works heavily rely on finding "anchor points" or their approximates, defined as instances belonging to a particular class almost surely. Nonetheless, finding anchor points remains a non-trivial task, and the estimation accuracy is also often throttled by the number of available anchor points. In this paper, we propose an alternative option to the above task. Our main contribution is the discovery of an efficient estimation procedure based on a clusterability condition. We prove that with clusterable representations of features, using up to third-order consensuses of noisy labels among neighbor representations is sufficient to estimate a unique transition matrix. Compared with methods using anchor points, our approach uses substantially more instances and benefits from a much better sample complexity. We demonstrate the estimation accuracy and advantages of our estimates using both synthetic noisy labels (on CIFAR-10/100) and real human-level noisy labels (on Clothing1M and our self-collected human-annotated CIFAR-10). Our code and human-level noisy CIFAR-10 labels are available at `https://github.com/UCSC-REAL/HOC`.

1. Introduction

Training deep neural networks (DNNs) relies on the large-scale labeled datasets while they often include a non-negligible fraction of wrongly annotated instances. The corrupted patterns tend to be memorized by the over-parameterized DNNs (Xia et al., 2021; Han et al., 2020), and lead to unexpected and disparate impacts (Liu, 2021).

A variety of approaches were proposed to address the problem of learning with noisy labels. The implementations of a major line of them, e.g., Patrini et al. (2017); Xiao et al. (2015); Xia et al. (2020b); Berthon et al. (2021); Xia et al. (2019); Yao et al. (2020b); Li et al. (2021), depend on accurate knowledge of the *noise transition matrix* $\boldsymbol{T}$, which characterizes the probabilities of a training example being wrongly annotated. It has been show that (Liu & Tao, 2015; Patrini et al., 2017), with perfect knowledge of $\boldsymbol{T}$, the minimizer of a corrected or reweighted expected risk (loss) defined on the noisy distribution is the same as the minimizer of the true expected risk (loss) of the clean distribution. These results clearly established the power and benefits of knowing $\boldsymbol{T}$.

Estimating $\boldsymbol{T}$ is challenging without accessing clean labels. Existing works on estimating $\boldsymbol{T}$ often rely on finding a number of high-quality anchor points (Scott, 2015; Liu & Tao, 2015; Patrini et al., 2017), or approximate anchor points (Xia et al., 2019), which are defined as the training examples that belong to a particular class almost surely. To find the anchor point, a model needs to be trained to accurately characterize the noisy label distribution. This model will help inform the selection of anchor points. Again relying on this model, $\boldsymbol{T}$ is then estimated using posterior noisy label distributions of the anchor points.

While the anchor point approach observes a significant amount of successes, it suffers from several limitations: 1) accurately fitting noisy distributions is challenging when the number of label classes is high; 2) the number of anchor points restricts the estimation accuracy; and 3) it lacks the flexibility to extend to more complicated noise settings. Other methods such as confident learning (Northcutt et al., 2017; 2021) may not explicitly identify anchor points, but they still need to fit the noisy distributions and find some "confident points", thus suffer from the above limitations.

In this paper, we provide an alternative to estimate $\boldsymbol{T}$ without resolving to anchor points. The only requirement we need is clusterability, i.e., the **two** nearest-neighbor representations of a training example and the example itself

[1]Department of Computer Science and Engineering, University of California, Santa Cruz, CA, USA [2]Beijing University of Posts and Telecommunications, Beijing, China. Correspondence to: Yang Liu <yangliu@ucsc.edu>.

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

belong to the same true label class. Our main contributions summarize as follows:

- Based on the clusterability condition, we propose a novel $\boldsymbol{T}$ estimator by exploiting a set of high-order consensuses information among neighbor representations' noisy labels. Compared with the methods using anchor points, our estimator uses a much larger set of training examples and benefits from a much better sample complexity.
- We prove that using up to third-order consensuses is sufficient to identify the true noise transition matrix uniquely.
- Extensive empirical studies on CIFAR-10/100 datasets with synthetic noisy labels, the Clothing1M dataset with real-world human noise, and the CIFAR-10 dataset with our self-collected human annotations, demonstrate the advantage of our estimator.
- Open-source contribution and flexible extension: we will contribute to the community 1) a generically applicable and light tool for fast estimation of the noise transition matrix. This flexible tool has the potential to be applied to more sophisticated noise settings, including instance-dependent ones (Section 3.4). 2) A noisy version of the CIFAR-10 dataset with human-level label noise.

1.1. Related Works

In the literature of learning with label noise, a major set of works focus on designing *risk-consistent* methods, i.e., performing empirical risk minimization (ERM) with specially designed loss functions on noisy distributions leads to the same minimizer as if performing ERM over the corresponding unobservable clean distribution. The *noise transition matrix* is a crucial component for implementing risk-consistent methods, e.g., loss correction (Patrini et al., 2017), loss reweighting (Liu & Tao, 2015), label correction (Xiao et al., 2015) and unbiased loss (Natarajan et al., 2013). To a certain degree, the knowledge of it also helps tune hyperparameters in other approaches, e.g., label smoothing (Lukasik et al., 2020). As introduced previously, anchor points are critical for estimating the transition matrix in above mentioned existing methods - we further elaborate this in Section 2.2.

Some recently proposed risk-consistent approaches do not require the knowledge of transition matrix, including: L_{DMI} (Xu et al., 2019) based on an information theoretical measure, peer loss (Liu & Guo, 2020) by punishing over-agreements with noisy labels, robust f-divergence (Wei & Liu, 2021), and CORES^2 (Cheng et al., 2020) built on a confidence-regularizer. However, to principally handle a more complicated case when the noise transition matrix depends on each feature locally, i.e., instance-dependent noise, the ability to estimate *local transition matrices* remains a significant and favorable property. Examples include the potential of applying local transition matrices to different groups of data (Xia et al., 2020b), using confidence scores to revise transition matrices (Berthon et al., 2021), and estimating the second-order information of local transition matrices (Zhu et al., 2021). Thus we need an estimation approach that scales and generalizes well to these situations.

As a growing literature, we are aware of other promising approaches that do not rely on the estimation of $\boldsymbol{T}$, e.g., focusing on the numerical property of loss functions and designing bounded loss functions (Amid et al., 2019a;b; Zhang & Sabuncu, 2018; Wang et al., 2019; Gong et al., 2018; Ghosh et al., 2017; Shu et al., 2020), using sample selection to pick up reliable instances from the dataset (Jiang et al., 2018; Han et al., 2018; Yu et al., 2019; Yao et al., 2020a; Wei et al., 2020), among many more. We compare to some of the popular ones using experiments.

2. Preliminaries

This section introduces the preliminaries, including problem formulation, anchor points, and the clusterability condition.

2.1. Our Setup

We summarize the important definitions as follows.

Clean/Noisy distribution The traditional classification problem with clean labels often builds on a set of N training examples denoted by $D := \{(x_n, y_n)\}_{n\in[N]}$, where $[N] := \{1, 2, \cdots, N\}$. Each example (x_n, y_n) could be seen as a snapshot of random variable (X, Y) drawn from a clean distribution $\mathcal{D}$. Let $\mathcal{X}$ and $\mathcal{Y}$ denote the space of feature X and label Y, respectively. In our considered weakly-supervised classification problem, instead of having access to the clean dataset D, the learner could only obtain a noisy dataset $\widetilde{D} := \{(x_n, \tilde{y}_n)\}_{n\in[N]}$, where the noisy label $\tilde{y}_n$ may or may not be the same as y_n. Noisy examples $(x_n, \tilde{y}_n)$ are generated according to random variables $(X, \widetilde{Y})$ drawn from a distribution $\widetilde{\mathcal{D}}$.

Noise transition matrix We model the relationship between (X, Y) and $(X, \widetilde{Y})$ using a noise transition matrix $\boldsymbol{T}(X)$, where each element $T_{ij}(X)$ represents the probability of mislabeling a clean label $Y = i$ to the noisy label $\widetilde{Y} = j$, i.e. $T_{ij}(X) := \mathbb{P}(\widetilde{Y} = j|Y = i, X)$. We call $\boldsymbol{T}(X)$ the *local* transition matrix in this paper since it is defined for a particular feature X. Most of the literature would focus on the case where the noise is independent of feature X: $\boldsymbol{T}(X) \equiv \boldsymbol{T}$. The knowledge of $\boldsymbol{T}$ enables a variety of learning with noisy label solutions. Below we illustrate solutions with the celebrated loss correction approach (Natarajan et al., 2013; Patrini et al., 2017).

The learning task The classification task aims to identify a classifier $\boldsymbol{f} : \mathcal{X} \to \mathcal{Y}$ that maps X to Y accurately. We focus on minimizing the empirical risk using DNNs with respect to the cross-entropy (CE) loss defined

as $\ell(\boldsymbol{f}(X), Y) = -\ln(f_X[Y]),\ Y \in [K]$, where $f_X[Y]$ denotes the Y-th component of column vector $\boldsymbol{f}(X)$ and K is the number of classes.

2.2. Loss Correction and Estimating $\boldsymbol{T}$

In the popular loss correction approach (Patrini et al., 2017), when the noise transition matrix is known, forward or backward loss correction can be applied to design a corrected loss. For example, the forward loss correction function can be designed as: $\ell^{\rightarrow}(\boldsymbol{f}(X), \widetilde{Y}) := \ell(\boldsymbol{T}^\top \boldsymbol{f}(X), \widetilde{Y})$, where $\boldsymbol{T}^\top$ denotes the transpose of matrix $\boldsymbol{T}$. If $\boldsymbol{T}$ is perfectly known in advance, it can be shown that the minimizer of the corrected loss under the noisy distribution is the same as the minimizer of the original loss ℓ under the clean distribution (Patrini et al., 2017).

We would like to emphasize that in addition to loss correction, the knowledge of noise transition matrices is potentially useful in other approaches, especially when dealing with the challenging instance-dependent label noise where $\boldsymbol{T}(X)$ differs for different X. For example, it was shown that knowing $\boldsymbol{T}(X)$ helps improve the robustness of peer loss when the noise transition matrix differs across instances (Zhu et al., 2021), and can help improve fairness guarantees when label noise is group-dependent (Wang et al., 2021). Knowing $\boldsymbol{T}$ also tends to be helpful in setting hyperparameters in sample selection (Han et al., 2018) and label smoothing (Lukasik et al., 2020; Wei et al., 2021).

Estimating $\boldsymbol{T}$ with anchor points The traditional approach for estimating $\boldsymbol{T}$ relies on anchor points (Scott, 2015; Liu & Tao, 2015), which are defined as the training examples (Xs) that belong to a specific class almost surely. Formally, an x is an anchor point for the class i if $\mathbb{P}(Y = i|X = x)$ is equal to one or close to one (Xia et al., 2019). Further, if $\mathbb{P}(Y = i|X = x) = 1$, we have $\mathbb{P}(\widetilde{Y} = j|X = x) = \sum_{k\in[K]} T_{kj}\mathbb{P}(Y = k|X = x) = T_{ij}$. The matrix $\boldsymbol{T}$ can be obtained via estimating the noisy class posterior probabilities for anchor points heuristically (Patrini et al., 2017) or theoretically (Liu & Tao, 2015).

While the anchor point approach observes a significant amount of successes, this method suffers from three major limitations:

- The implementation of it requires that the trained model can perfectly predict the probability of the noisy labels, which is challenging when the number of classes is high, and when the number of training instances is limited.
- The number of available and identifiable anchor points can become a bottleneck even if the posterior distribution can be perfectly learned.
- The lack of flexibility to zoom into a subset of training data also limits its potential to be applied to estimate local transition matrices for more challenging instance-dependent settings (Xia et al., 2019).

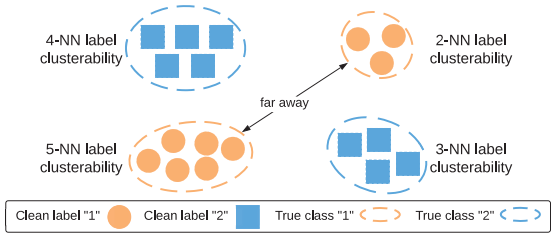

Figure 1. Illustration of k-NN label clusterability.

2.3. Clusterability

The alternative we are seeking builds on the notion of clusterability. Intuitively, clusterability implies that two instances are likely to have the same labels if they are close to each other (Gao et al., 2016). To facilitate the discovery of close-by instances, our solution will resolve to representation learning (Bengio et al., 2013). Recent literature shows, even though label noise makes the model generalizes poorly, it still induces good representations (Li et al., 2020). Formally, for a neural network with both convolutional layers and linear layers, e.g., ResNet (He et al., 2016), we denote the convolution layers by function $\boldsymbol{f}_{\text{conv}}$ and the representations by $\bar{X} := \boldsymbol{f}_{\text{conv}}(X)$. With the above, we define k-Nearest-Neighbor (k-NN) label clusterability[1] as:

Definition 1 (k-NN label clusterability)**.** *We call a dataset D satisfies k-NN label clusterability if $\forall n \in [N]$, the representation $\bar{x}_n$ and its k-Nearest-Neighbor $\bar{x}_{n_1}, \cdots, \bar{x}_{n_k}$ belong to the same true class.*

See Figure 1 for an illustration of the k-NN clusterability. There are three primary properties of the definition:

- The k_1-NN label clusterability condition is harder to satisfy than k_2-NN label clusterability when $k_1 > k_2$;
- The cluster containing the same clean labels is not required to be a continuum, e.g., in Figure 1, two clusters of class "1" can be far away;
- Our k-NN label clusterability only requires the existence of these feasible points, i.e., specifying the true class is not necessary.

The k-NN label clusterability likely holds in many tasks, such as image classification when features are well-extracted by convolutional layers (Han et al., 2019; Ji et al., 2019; Kolesnikov et al., 2019) and each feature belongs to a unique true class. The high-level intuition is that similar representations should belong to the same label class. One can consider a label generation process (Feldman, 2020; Liu, 2021) where the feature distribution is modeled as a mixture

[1]Distances are measured between representations. Feature x_n and its representation $\bar{x}_n$ refer to the same data point in different views.

of many disjoint sub-distributions, and the labeling function maps each sub-distribution to a unique label class. Therefore, samples from the same sub-distribution have the same true label. In this paper, instead of requiring identical labels for a big cluster defined by a large k, we will only require the **2** nearest neighbors to have the same clean labels with the example itself, i.e., 2-*NN label clusterability*. Its feasibility will be demonstrated in Section 5.3.

Comparison to anchor points The anchor point approach relies on training a classifier to identify anchor points and the corresponding true class. Our label clusterability definition *does not require the knowledge of true label class* as claimed in the third property. Moreover, if good representations are available apriori, our method is *model-free*.

Next, we will elaborate our proposed $\boldsymbol{T}$ estimator leveraging 2-NN label clusterability. Relaxation of 2-NN label clusterability is discussed in Appendix C.1.

3. The Power of High-Order Consensuses

We now present our alternative to estimate $\boldsymbol{T}$. Our idea builds around the concept of using high-order consensuses of the noisy labels $\widetilde{Y}$s among each training instance and its 2-NN. In this section, we consider the case when $\boldsymbol{T}(X)$ is the same for different X, i.e., $\boldsymbol{T}(X) \equiv \boldsymbol{T}$.

3.1. Warm-up: A Binary Example

For a gentle start, consider binary cases ($K = 2$) with classes $\{1, 2\}$. Short-hand error rates $e_1 := T_{12} := \mathbb{P}(\widetilde{Y} = 2|Y = 1)$, $e_2 := T_{21} := \mathbb{P}(\widetilde{Y} = 1|Y = 2)$. $p_1 := \mathbb{P}(Y = 1)$ denotes the clean prior probability of class-1.

We are inspired by the matching mechanism for binary error rates estimation (Liu & Chen, 2017; Liu et al., 2020). Intuitively, with 1-NN label clusterability, for two representations in the same dataset with minimal distance, their labels should be identical. Otherwise, we know there must be exactly one example with the corrupted label. Similarly, if k-NN label clusterability holds, by comparing the noisy label of one representation with its k-NN, we can write down the probability of the $k + 1$ noisy label consensuses (including agreements and disagreements) as a function of e_1, e_2, p_1.

Going beyond votes from k-NN noisy labels To infer whether the label of an instance is clean or corrupted, one could use the 2-NN of this instance and take a majority vote. For example, if the considered instance has the label "1" and the other two neighbors have the label "2", it can be inferred that the label of the considered instance is corrupted since "2" is in the majority. Nonetheless, this inference would be wrong when the 2-NN are corrupted. Increasing accuracy of the naive majority vote (Liu & Liu, 2015) or other inference approaches (Liu et al., 2012) requires stronger clusterability that more neighbor representations should belong to the same clean class. Our approach goes beyond simply using the votes among k-NNs. Instead, we will rely on the statistics of high-order consensuses among the k-NN noisy labels. As a result, our method enjoys a robust implementation with only requiring 2-NN label clusterability.

Consensuses in binary cases We now derive our approach for the binary case to deliver our main idea. We present the general form of our estimator in the next subsection. Let $\widetilde{Y}_1$ be the noisy label of one particular instance, $\widetilde{Y}_2$ and $\widetilde{Y}_3$ be the noisy labels of its nearest neighbor and second nearest neighbor. With 2-NN label clusterability, their clean labels are identical, i.e. $Y_1 = Y_2 = Y_3$. For $\widetilde{Y}_1$, noting $\mathbb{P}(\widetilde{Y}_1 = j) = \sum_{i\in[K]} \mathbb{P}(\widetilde{Y}_1 = j|Y_1 = i) \cdot \mathbb{P}(Y_1 = i)$, we have the following **two** first-order equations:

$$\mathbb{P}(\widetilde{Y}_1 = 1) = p_1(1 - e_1) + (1 - p_1)e_2,$$
$$\mathbb{P}(\widetilde{Y}_1 = 2) = p_1e_1 + (1 - p_1)(1 - e_2).$$

For the second-order consensuses, we have

$$\begin{aligned}
&\mathbb{P}(\widetilde{Y}_1 = j_1, \widetilde{Y}_2 = j_2)\\
&\overset{(a)}{=} \sum_{i\in[K]} \mathbb{P}(\widetilde{Y}_1 = j_1, \widetilde{Y}_2 = j_2|Y_1 = i, Y_2 = i) \cdot \mathbb{P}(Y_1 = i)\\
&\overset{(b)}{=} \sum_{i\in[K]} \mathbb{P}(\widetilde{Y}_1 = j_1|Y_1 = i) \cdot \mathbb{P}(\widetilde{Y}_2 = j_2|Y_2 = i) \cdot \mathbb{P}(Y_1 = i),
\end{aligned}$$

where equality (a) holds due to the 2-NN label clusterability, i.e., $Y_1 = Y_2 (= Y_3)$ w.p. 1, and equality (b) holds due to the conditional independency between $\widetilde{Y}_1$ and $\widetilde{Y}_2$ given their clean labels. In total, there are **four** second-order equations for different combinations of $\widetilde{Y}_1, \widetilde{Y}_2$, e.g.,

$$\mathbb{P}(\widetilde{Y}_1 = 1, \widetilde{Y}_2 = 1) = p_1(1 - e_1)^2 + (1 - p_1)e_2^2,$$
$$\mathbb{P}(\widetilde{Y}_1 = 1, \widetilde{Y}_2 = 2) = p_1(1 - e_1)e_1 + (1 - p_1)e_2(1 - e_2).$$

Similarly, given $Y_1 = Y_2 = Y_3$, there are **eight** third-order equations defined for consensuses among $\widetilde{Y}_1, \widetilde{Y}_2, \widetilde{Y}_3$, e.g.,

$$\mathbb{P}(\widetilde{Y}_1 = 1, \widetilde{Y}_2 = 1, \widetilde{Y}_3 = 1) = p_1(1 - e_1)^3 + (1 - p_1)e_2^3.$$

Figure 2 illustrates the above consensus checking process. We leave more details and full derivations to Appendix A. The left-hand side of each above equation is the probability of a particular first-, second-, or third-order consensus pattern of $\widetilde{Y}$, which could be estimated given the noisy dataset $\widetilde{D}$. These consensus patterns encode the high-order information of $\boldsymbol{T}$. Later in Section 4.1, we will prove that given the consensus probability (LHS), the first three order consensus equations we presented above are sufficient to jointly identify a unique solution to $\boldsymbol{T}$, which indeed corresponds to the true $\boldsymbol{T}$.

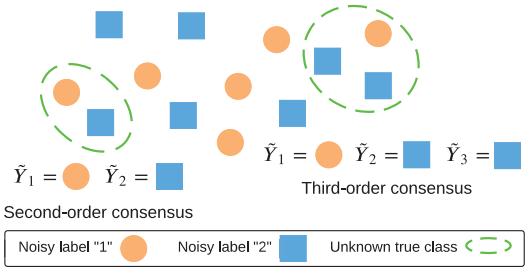

Figure 2. Illustration of high-order consensuses.

3.2. Estimating T: The General Form

We generalize this idea to classifications with multiple classes. For a K-class classification problem, define $\boldsymbol{p} := [\mathbb{P}(Y=i), i \in [K]]^\top$ and

$$\boldsymbol{T}_r := \boldsymbol{T} \cdot \boldsymbol{S}_r, \ \forall r \in [K], \tag{1}$$

where $\boldsymbol{S}_r := [\boldsymbol{e}_{r+1}, \boldsymbol{e}_{r+2}, \cdots, \boldsymbol{e}_K, \boldsymbol{e}_1, \boldsymbol{e}_2, \cdots \boldsymbol{e}_r]$ is a cyclic permutation matrix, and $\boldsymbol{e}_r$ is the $K \times 1$ column vector of which the r-th element is 1 and 0 otherwise. The matrix $\boldsymbol{S}_r$ cyclically shifts each column of $\boldsymbol{T}$ to its left side by r units. Similar to the previous binary example, the LHS of the equation is the probability of different distributions of $\widetilde{Y}$s among each instance and its 2-NN. Let $(i+r)_K := [(i+r-1) \mod K] + 1$. For the first-, second-, and third-order consensuses, we can respectively denote them in vector forms as follows ($\forall r \in [K], s \in [K]$).

$$\boldsymbol{c}^{[1]} = [\mathbb{P}(\widetilde{Y}_1 = i), i \in [K]]^\top,$$
$$\boldsymbol{c}^{[2]}_r = [\mathbb{P}(\widetilde{Y}_1 = i, \widetilde{Y}_2 = (i+r)_K), i \in [K]]^\top,$$
$$\boldsymbol{c}^{[3]}_{r,s} = [\mathbb{P}(\widetilde{Y}_1 = i, \widetilde{Y}_2 = (i+r)_K, \widetilde{Y}_3 = (i+s)_K), i \in [K]]^\top.$$

Denote by $\circ$ the Hadamard product of two matrices. We now present the system of consensus equations for estimating $\boldsymbol{T}$ and $\boldsymbol{p}$ in the general form:

Consensus Equations

- First-order (K equations):

$$\boldsymbol{c}^{[1]} := \boldsymbol{T}^\top \boldsymbol{p}, \tag{2}$$

- Second-order (K^2 equations):

$$\boldsymbol{c}^{[2]}_r := (\boldsymbol{T} \circ \boldsymbol{T}_r)^\top \boldsymbol{p}, \ r \in [K], \tag{3}$$

- Third-order (K^3 equations):

$$\boldsymbol{c}^{[3]}_{r,s} := (\boldsymbol{T} \circ \boldsymbol{T}_r \circ \boldsymbol{T}_s)^\top \boldsymbol{p}, \ r, s \in [K]. \tag{4}$$

While we leave the full details of derivation to Appendix A, we show one second-order consensus below for an example:

$$\begin{aligned}
&\boldsymbol{e}_j^\top \boldsymbol{c}^{[2]}_r = \mathbb{P}(\widetilde{Y}_1 = j, \widetilde{Y}_2 = (j+r)_K) \\
&\overset{(a)}{=} \sum_{i \in [K]} \mathbb{P}(\widetilde{Y}_1 = j | Y_1 = i) \mathbb{P}(\widetilde{Y}_2 = (j+r)_K | Y_2 = i) \mathbb{P}(Y_1 = i) \\
&= \sum_{i \in [K]} T_{i,j} \cdot T_{i,(j+r)_K} \cdot p_i \overset{(b)}{=} \boldsymbol{e}_j^\top (\boldsymbol{T} \circ \boldsymbol{T}_r)^\top \boldsymbol{p},
\end{aligned}$$

where equality (a) holds again due to the 2-NN label clusterability and the conditional independency (similar to binary cases), and equality (b) holds due to $\boldsymbol{T}_r[i,j] = T_{i,(j+r)_K}$.

We note that although there are higher-order consensuses according to this rule, we only consider up to third-order consensuses of $\widetilde{Y}$ as shown in Eqns. (2)–(4). For ease of notation, we define two stacked vector-forms for $\boldsymbol{c}^{[2]}_{r,s}, \boldsymbol{c}^{[3]}_{r,s}$:

$$\boldsymbol{c}^{[2]} := [(\boldsymbol{c}^{[2]}_r)^\top, \forall r \in [K]]^\top, \tag{5}$$
$$\boldsymbol{c}^{[3]} := [(\boldsymbol{c}^{[3]}_{r,s})^\top, \forall r, s \in [K]]^\top. \tag{6}$$

3.3. The HOC Estimator

Solving the consensus equations requires estimating the consensus probabilities $\boldsymbol{c}^{[1]}$, $\boldsymbol{c}^{[2]}$, and $\boldsymbol{c}^{[3]}$. In this subsection, we will first show the procedures for estimating these probabilities and then formulate an efficient optimization problem for $\boldsymbol{T}$ and $\boldsymbol{p}$. To summarize, there are three steps:

- **Step 1:** Find 2-NN for each $\bar{x}_n$ from the noisy dataset $\widetilde{D}$.
- **Step 2:** Compute each $\hat{\boldsymbol{c}}^{[\nu]}$ using $\bar{x}_n$ and their 2-NN.
- **Step 3:** Formulate the optimization problem in (10).

Denote by $E \subseteq [N]$. We elaborate on each step as follows.

Step 1: Find 2-NN Given the noisy dataset $\{(x_n, \tilde{y}_n), n \in E\}$, for each representation $\bar{x}_n = \boldsymbol{f}_{\mathsf{conv}}(x_n)$, we can find its 2-NN $\bar{x}_{n_1}, \bar{x}_{n_2}$ as:

$$n_1 = \underset{n' \in E, n' \neq n}{\arg\min} \ \mathsf{Dist}(\bar{x}_n, \bar{x}_{n'}), \ n_2 = \underset{n' \in E, n' \neq n \neq n_1}{\arg\min} \ \mathsf{Dist}(\bar{x}_n, \bar{x}_{n'}),$$

and the corresponding noisy labels $\tilde{y}_{n_1}, \tilde{y}_{n_2}$. $\mathsf{Dist}(A, B)$ measures the distance between A and B - we will use Dist as the negative cosine similarity in our experiment.

Step 2: Empirical mean Denote by $\mathbb{1}\{\cdot\}$ the indicator function taking value 1 when the specified condition is met and 0 otherwise. Let E be a set of indices and $|E|$ be the number of them. The probability of each high-order consensus could be estimated by the empirical mean using a particular set of sampled examples in E: $\{(\tilde{y}_n, \tilde{y}_{n_1}, \tilde{y}_{n_2}), n \in E\}$ as follows ($\forall i$).

$$\begin{aligned}
\hat{\boldsymbol{c}}^{[1]}[i] &= \frac{1}{|E|} \sum_{n \in E} \mathbb{1}\{\tilde{y}_n = i\}, \\
\hat{\boldsymbol{c}}^{[2]}_r[i] &= \frac{1}{|E|} \sum_{n \in E} \mathbb{1}\{\tilde{y}_n = i, \tilde{y}_{n_1} = (i+r)_K\}, \\
\hat{\boldsymbol{c}}^{[3]}_{r,s}[i] &= \frac{1}{|E|} \sum_{n \in E} \mathbb{1}\{\tilde{y}_n = i, \tilde{y}_{n_1} = (i+r)_K, \tilde{y}_{n_2} = (i+s)_K\}.
\end{aligned} \tag{7}$$

The motivation of identifying a subset E for the estimators is due to the desired provable convergence to the expectation. Each 3-tuple in the sample should be independent and identically distributed (i.i.d.) so that each $\hat{\boldsymbol{c}}^{[\nu]}$ is consistent. However, the existence of nearest neighbors, e.g., when both n and n_1 belong to E and n is a 2-NN of n_1, may violate the i.i.d. property of these 3-tuples. Denote by

$$E_3^* = \underset{E \subseteq [N]}{\arg\max} |E|, \ \ \text{s.t.} \ \ |\{n, n_1, n_2, \forall n \in E\}| = 3|E|.$$

Then any subset $E \subseteq E_3^*$ guarantees the i.i.d. property. Note it is generally time-consuming to find the best E. For an efficient solution (with empirical approximation), we randomly sample $|E|$ center indices from $[N]$ and repeat Step 1 and Step 2 multiple times with different E (as Line 3 – Line 8 in Algorithm 2). We will further discuss the magnitude of $|E|$ in Section 4.2 and Appendix B.3.

Step 3: Optimization With $\hat{\boldsymbol{c}}^{[1]}$, $\hat{\boldsymbol{c}}^{[2]}$, and $\hat{\boldsymbol{c}}^{[3]}$, we formulate the optimization problem in (8) to jointly solve for $\boldsymbol{T}, \boldsymbol{p}$.

$$\underset{\boldsymbol{T},\boldsymbol{p}}{\text{minimize}} \quad \sum_{\nu=1}^{3} \|\hat{\boldsymbol{c}}^{[\nu]} - \boldsymbol{c}^{[\nu]}\|_2 \tag{8a}$$

$$\text{subject to} \quad \text{Eqns. (1) – (6)} \tag{8b}$$

$$p_i \geq 0, T_{ij} \geq 0, i, j \in [K] \tag{8c}$$

$$\sum_{i\in[K]} p_i = 1, \sum_{j\in[K]} T_{ij} = 1, i \in [K]. \tag{8d}$$

The crucial components in (8) are:

- Objective (8a): the sum of errors from each order of consensus, where the error is defined in ℓ_2-norm.
- Variable definitions (8b): the closed-form relationship between intermediate variables (such as $\boldsymbol{c}^{[\nu]}$ and $\boldsymbol{T}_r$) and the optimized variables ($\boldsymbol{T}$ and $\boldsymbol{p}$).
- Constraints (8c) and (8d): feasibility of a solution.

Challenges for solving the constrained optimization problem The problem in (8) is a constrained optimization problem with $K(K+1)$ variables, $K(K+1)$ inequality constraints, and $(K+1)$ equality constraints, and it is generally hard to guarantee its convexity. Directly solving this problem using the Lagrangian-dual method may take a long time to converge (Boyd et al., 2004).

Unconstrained soft approximation Notice that both $\boldsymbol{p}$ and each row of $\boldsymbol{T}$ are probability measures. Instead of directly solving for $\boldsymbol{T}$ and $\boldsymbol{p}$, we seek to relax the constraints by introducing auxiliary and unconstrained variables to represent $\boldsymbol{T}$ and $\boldsymbol{p}$. Particularly, we turn to optimizing variables $\bar{\boldsymbol{T}} \in \mathbb{R}^{K\times K}$ and $\bar{\boldsymbol{p}} \in \mathbb{R}^K$ that are associated with $\boldsymbol{T}$ and $\boldsymbol{p}$ by $\boldsymbol{T} := \sigma_T(\bar{\boldsymbol{T}})$, $\boldsymbol{p} := \sigma_p(\bar{\boldsymbol{p}})$, where $\sigma_T(\cdot)$ and $\sigma_p(\cdot)$ are softmax functions such that

$$T_{ij} := \frac{\exp(\bar{T}_{ij})}{\sum_{k\in[K]}\exp(\bar{T}_{ik})}, \; p_i := \frac{\exp(\bar{p}_i)}{\sum_{k\in[K]}\exp(\bar{p}_k)}. \tag{9}$$

Therefore, we can drop all the constraints in (8) and focus on solving the unconstrained optimization problem with $K(K+1)$ variables. Our new optimization problem is given as follows:

$$\underset{\bar{\boldsymbol{T}},\bar{\boldsymbol{p}}}{\text{minimize}} \quad \sum_{\nu=1}^{3} \|\hat{\boldsymbol{c}}^{[\nu]} - \boldsymbol{c}^{[\nu]}\|_2 \tag{10a}$$

$$\text{subject to} \quad \text{Eqns. (1) – (6), Eqn. (9).} \tag{10b}$$

Algorithm 1 The HOC Estimator

Input: Rounds: G. Sample size: $|E|$. Noisy dataset: $\widetilde{D} = \{(x_n, \tilde{y}_n)\}_{n\in[N]}$. Representation extractor: $\boldsymbol{f}_{\text{conv}}$.
Initialization: Set $\hat{\boldsymbol{c}}^{[1]}$, $\hat{\boldsymbol{c}}^{[2]}$, $\hat{\boldsymbol{c}}^{[3]}$ to 0. Extract representations $x_n \leftarrow \boldsymbol{f}_{\text{conv}}(x_n), \forall n \in [N]$. $\bar{\boldsymbol{T}} = K\boldsymbol{I} - \mathbf{1}\mathbf{1}^\top$. $\bar{\boldsymbol{p}} = \mathbf{1}/K$. *// $\boldsymbol{I}$: identity matrix, $\mathbf{1}$: all-ones column vector.*
repeat
  $E \leftarrow$ `RndSmp`$([N], |E|)$; *// sample $|E|$ center indices*
  $\{(\tilde{y}_n, \tilde{y}_{n_1}, \tilde{y}_{n_2}), n \in [E]\} \leftarrow$ `Get2NN`$(\widetilde{D}, E)$; *// find the noisy labels of the 2-NN of $x_n, n \in [E]$*
  $(\hat{\boldsymbol{c}}^{[1]}_{\text{tmp}}, \hat{\boldsymbol{c}}^{[2]}_{\text{tmp}}, \hat{\boldsymbol{c}}^{[3]}_{\text{tmp}}) \leftarrow$ `CountFreq`(E) *// as Eqn. (7)*
  $\hat{\boldsymbol{c}}^{[\nu]} \leftarrow \hat{\boldsymbol{c}}^{[\nu]} + \hat{\boldsymbol{c}}^{[\nu]}_{\text{tmp}}, \nu \in \{1,2,3\}$;
until G times
$\hat{\boldsymbol{c}}^{[\nu]} \leftarrow \hat{\boldsymbol{c}}^{[\nu]}/G, \nu \in \{1,2,3\}$; *// estimate $\boldsymbol{c}^{[\nu]}$ G times*
Solve the unconstrained problem in (10) with $(\hat{\boldsymbol{c}}^{[1]}, \hat{\boldsymbol{c}}^{[2]}, \hat{\boldsymbol{c}}^{[3]})$ by gradient decent, get $\bar{\boldsymbol{T}}$ and $\bar{\boldsymbol{p}}$
Output: Estimates $\hat{\boldsymbol{T}} \leftarrow \sigma_T(\bar{\boldsymbol{T}})$, $\hat{\boldsymbol{p}} \leftarrow \sigma_p(\bar{\boldsymbol{p}})$.

Equations in (10b) are presented only for a clear objective function. Given the solution of problem (10), we can calculate $\boldsymbol{T}$ and $\boldsymbol{p}$ according to Eqn. (9). Note the search space of $\boldsymbol{T}$ before and after soft approximation differs only in corner cases (before: $T_{ij} \geq 0$, after: $T_{ij} > 0$). For each original and non-corner $\boldsymbol{T}$, there exists a soft approximated $\boldsymbol{T}$ that leads to the same transition probabilities. Thus the soft approximation preserves the property of $\boldsymbol{T}$, e.g. the uniqueness in Theorem 1. Algorithm 1 summarizes our High-Order-Consensus (HOC) estimator.

3.4. Flexible Extensions to Instance-Dependent Noise

Algorithm 1 provides a generically applicable and light tool for fast estimation of $\boldsymbol{T}$. The flexibility makes it possible to be applied to more sophisticated instance-dependent label noise. We briefly discuss possible applications to estimating the local noise transition matrix $\boldsymbol{T}(X)$.

Locally homogeneous label noise Intuitively, by considering a local dataset in which every representation shares the same $\boldsymbol{T}(X)$, the method in Section 3.2 can then be applied locally to estimate the local $\boldsymbol{T}(X)$. Specially, using a "waypoint" $\bar{x}_n$, we build a local dataset $\widetilde{D}_n$ that includes the M-NN of $\bar{x}_n$, i.e., $\widetilde{D}_n = \{(x_n, \tilde{y}_n)\} \cup \{(x_{n_i}, \tilde{y}_{n_i}), \forall i \in [M]\}$, where $\{n_i, i \in [M]\}$ are the indices of the M-NN of $\bar{x}_n$. We introduce the following definitions:

Definition 2 (M-NN noise clusterability)**.** *We call $\widetilde{D}_n$ satisfies M-NN noise clusterability if the M-NN of $\bar{x}_n$ have the same noise transition matrix as x_n, i.e., $\boldsymbol{T}(x_n) = \boldsymbol{T}(x_{n_i}), \forall i \in [M]$.*

Definition 3 ((H,M)-coverage)**.** *We call $\widetilde{D}$ satisfies (H,M)-coverage if there exist H instances $\bar{x}_{h(n)}, n \in [H]$ such that $\widetilde{D} = \cup_{n=1}^{H} \widetilde{D}_{h(n)}$, where each $\widetilde{D}_{h(n)}$ satisfies M-NN noise clusterability.*

Note Dentition 2 focuses on the clusterability of noise transition matrices, which is different from the clusterability of the true classes of labels. When M-NN noise clusterability holds for $\bar{x}_n$, the label noise in local dataset $\widetilde{D}_n$ is effectively homogeneous. If $\widetilde{D}$ further satisfies (H, M)-coverage, we can divide the training data $\widetilde{D}$ to H local sub-datasets $\widetilde{D}_{h(n)}, n \in [H]$ and separately apply Algorithm 1 on each of them. The local estimates allow us to apply loss correction separately using different $\boldsymbol{T}(X)$ at different parts of the training data. Besides, when there is no M-NN noise clusterability, we may require knowing properly constructed sub-spaces to separate the data, with each part of them sharing similar noise rates (Xia et al., 2020a;b). We leave more detailed discussions in Appendix C.2.

4. Theoretical Guarantees

We will prove that our consensus equations are sufficient for estimating a unique $\boldsymbol{T}$, and show the advantage of our approach in terms of a better sample complexity than the anchor point approach.

4.1. Uniqueness of Solution

Before formally presenting the uniqueness guarantee, we introduce two assumptions as we will need.

Assumption 1 (Nonsingular $\boldsymbol{T}$)**.** *The noise transition matrix is non-singular, i.e.,* $\mathsf{Rank}(\boldsymbol{T}) = K$.

Assumption 2 (Informative $\boldsymbol{T}$)**.** *The diagonal elements of* $\boldsymbol{T}$ *are dominant, i.e.,* $T_{ii} > T_{ij}, \forall i \in [K], j \in [K], j \neq i$.

Assumption 1 is commonly made in the literature and ensures the effect of label noise is invertible (Van Rooyen & Williamson, 2017). Assumption 2 characterizes a particular permutation of row vectors in $\boldsymbol{T}$ (Liu et al., 2020). See more discussions on their feasibility in Appendix C.3. The uniqueness is formally stated in Theorem 1. The proof is sketched at the end of main paper and detailed in Appendix B.1.

Theorem 1. *When* $\widetilde{D}$ *satisfies the* 2*-NN label clusterability and* $\boldsymbol{T}$ *is nonsingular and informative, with a perfect knowledge of* $\boldsymbol{c}^{[\nu]}, \nu = 1, 2, 3$*, the solution of consensus equations (2) – (4) returns the true* $\boldsymbol{T}$ *uniquely.*

Challenges Proving Theorem 1 is challenging due to: 1) The coupling effect between $\boldsymbol{T}$ and $\boldsymbol{p}$ makes the structure of solution $\boldsymbol{T}$ unclear; 2) Naively replacing $\boldsymbol{p}$, e.g., using $\boldsymbol{p} = (\boldsymbol{T}^\top)^{-1}\boldsymbol{c}^{[1]}$, will introduce matrix inverse, which cannot be canceled with the Hadamard product; 3) A system of third-order equations with K^2 variables will have up to 3^{K^2} solutions and the closed-form is not explicit.

Local estimates Our next corollary 1 extends Theorem 1 to local datasets, when $\boldsymbol{T}$ can be heterogeneous.

Corollary 1. *When* $\widetilde{D}$ *satisfies* (H, M)*-coverage, each* $\widetilde{D}_{h(n)}$ *satisfies 2-NN label clusterability, and* $\boldsymbol{T}(x_{h(n)})$ *is nonsingular and informative, with a perfect knowledge of the local* $\boldsymbol{c}^{[\nu]}, \nu = 1, 2, 3$*, the solution of consensus equations (2) – (4) is unique and recovers* $\boldsymbol{T}(x_{h(n)})$.

4.2. Sample Complexity

We next show that with the estimates $\hat{\boldsymbol{c}}^{[1]}$, $\hat{\boldsymbol{c}}^{[2]}$, and $\hat{\boldsymbol{c}}^{[3]}$, HOC returns a reasonably well solution.

Recall that, in Section 3.3, Step 2 requires a particular $E \subseteq E_3^*$ to guarantee the i.i.d. property of the sample $\{(\tilde{y}_n, \tilde{y}_{n_1}, \tilde{y}_{n_2}), n \in E\}$. For a tractable sample complexity, we focus on a particular dataset $\widetilde{D}$ and feature extractor $\boldsymbol{f}_{\mathsf{conv}}$ such that 1) $|E_3^*| = \Theta(N)$ and 2) $T_{ij} = \frac{1-T_{ii}}{K-1}, \forall j \neq i, i \in [N], j \in [N]$. Supposing each tuple is drawn from non-overlapping balls, condition 1) is satisfied when the number of these non-overlapping balls covering the representation space is $\Theta(N)$. See Appendix B.2 for a detailed example when the representations are uniformly distributed. Lemma 1 shows the error upper bound of our estimates $\hat{\boldsymbol{c}}^{[\nu]}, \nu = 1, 2, 3$. See Appendix B.3 for the proof.

Lemma 1. *With probability* $1-\delta$*,* $\forall \nu, l$*, the estimation error* $|\hat{\boldsymbol{c}}^{[\nu]}[l] - \boldsymbol{c}^{[\nu]}[l]|$ *is bounded at the order of* $O(\sqrt{\ln(1/\delta)/N})$.

Lemma 1 is effectively the sample complexity of estimating $|E_3^*|$ i.i.d. random variables by the sample mean. Due to assuming a uniform diagonal $\boldsymbol{T}$, we only need to consider the estimation error of $\hat{T}_{ii}$. For each $i \in [K]$, see the result in Theorem 2 and the proof in Appendix B.4.

Theorem 2. *When* $T_{ii} > \frac{1-\mathbb{P}(Y=i)+(K-1)\mathbb{P}(\widetilde{Y}=i)}{K(K-1)\mathbb{P}(Y=i)}$*, w.p.* $1 - 2\delta$*,* $|\hat{T}_{ii} - T_{ii}|$ *is bounded at the order of* $O(\sqrt{\ln(1/\delta)/N})$.

Theorem 2 indicates the sample complexity of our solution has the same order in terms of N compared to a standard empirical mean estimation in Lemma 1. Remark 1 shows our approach is better than using a set of anchor points in the sample complexity.

Remark 1 (Comparison)**.** *The methods based on anchor points estimate* $\boldsymbol{T}$ *with* $N_{AC} < N$ *(*$N_{AC} \ll N$ *in many cases) anchor points. Thus w.p.* $1 - \delta$*, the estimation error is at the order of* $O(\sqrt{\ln(1/\delta)/N_{AC}})$.

5. Experiments

We present experiment settings as follows.

Datasets and models HOC is evaluated on three benchmark datasets: CIFAR-10, CIFAR-100 (Krizhevsky et al., 2009) and Clothing1M (Xiao et al., 2015). For the standard training step, we use ResNet34 for CIFAR-10 and CIFAR-100, and ResNet50 for Clothing1M. The representations come from the outputs before the final fully-connected layer of ResNet34/50. The distance between different representations is measured by the negative cosine similarity.

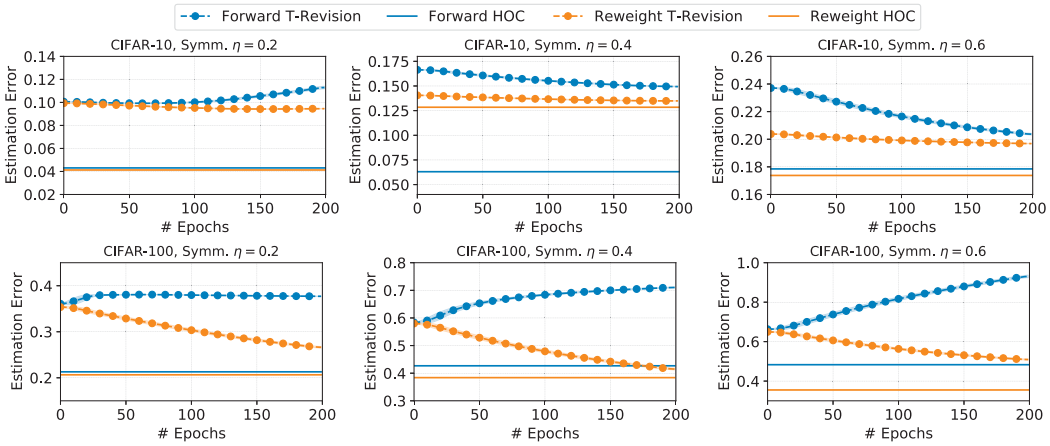

Figure 3. Comparison of estimation errors of $\boldsymbol{T}$ given by T-Revision (Xia et al., 2019) and our HOC estimator. The error is measured by the matrix $L_{1,1}$-norm with a normalization factor K, i.e. $\|\hat{\boldsymbol{T}} - \boldsymbol{T}\|_{1,1}/K$. Forward: Using the forward corrected loss (Patrini et al., 2017) Reweight: Using the reweighted loss (Liu & Tao, 2015). Symmetric noise is applied.

Table 1. The best epoch (clean) test accuracy (%) with synthetic label noise.

Method	*Inst. CIFAR-10*			*Inst. CIFAR-100*		
	$\eta = 0.2$	$\eta = 0.4$	$\eta = 0.6$	$\eta = 0.2$	$\eta = 0.4$	$\eta = 0.6$
CE (Standard)	85.66±0.62	76.89±0.93	60.29±1.17	57.26±1.33	41.33±0.89	25.08±1.85
Peer Loss (Liu & Guo, 2020)	89.52±0.22	83.44±0.30	75.15±0.82	61.13±0.48	48.01±0.12	33.00±1.47
L_{DMI} (Xu et al., 2019)	88.67±0.70	83.65±1.13	69.82±1.72	57.36±1.18	43.06±0.97	26.13±2.39
L_q (Zhang & Sabuncu, 2018)	85.66±1.09	75.24±1.07	61.30±3.35	56.92±0.24	40.17±1.52	25.58±3.12
Co-teaching (Han et al., 2018)	88.84±0.20	72.61±1.35	63.76±1.11	43.37±0.47	23.20±0.44	12.43±0.50
Co-teaching+ (Yu et al., 2019)	89.82±0.39	73.44±0.38	63.61±1.78	41.62±1.05	24.73±0.85	12.25±0.35
JoCoR (Wei et al., 2020)	88.82±0.20	71.13±1.94	63.88±2.05	44.55±0.62	23.92±0.32	13.05±1.10
Forward (Patrini et al., 2017)	87.87±0.96	79.81±2.58	68.32±1.68	57.69±1.55	42.62±0.92	27.35±3.42
T-Revision (Xia et al., 2019)	**90.31±0.37**	84.99±0.81	72.06±3.40	58.00±0.20	40.01±0.32	40.88±7.57
HOC Global	89.71±0.51	84.62±1.02	70.67±3.38	**68.82±0.26**	**62.29±1.11**	**52.96±1.85**
HOC Local	**90.03±0.15**	**85.49±0.80**	**77.40±0.47**	67.47±0.85	61.20±1.04	49.84±1.81

Noise type HOC is tested on both synthetic label noise and real-world human label noise. The synthetic label noise includes two regimes: *symmetric* noise and *instance-dependent* noise. For both regimes, the noise rate η is the overall ratio of instances with a corrupted label in the whole dataset. The symmetric noise is generated by randomly flipping a clean label to the other possible classes w.p. η (Xia et al., 2019). The basic idea of generating instance-dependent noise is to randomly generate one vector for each class (K vectors in total) and project each incoming feature onto these K vectors (Xia et al., 2020b). The label noise is added by jointly considering the clean label and the projection results. See Appendix D.1 for more details. The *real-world human noise* comes from human annotations. Particularly, for the $50,000$ training images in CIFAR-10, we *re-collect* human annotations[2] from Amazon Mechanical Turk (MTurk) in February 2020. For the Clothing1M dataset, we train on 1 million noisy training instances reflecting the real-world human noise.

[2]We only collect one annotation for each image with a cost of ¢10 per image.

5.1. Performance of Estimating $\boldsymbol{T}$

We compare HOC with T-revision (Xia et al., 2019) following the flow: 1) Estimation → 2) Training → 3) Revision. For a fair comparison, we follow their training framework and parameter settings to get representations. Particularly, we obtain the same model as the one that T-revision adopts before revision. As illustrated in Figure 3, compared with the dynamical revision adopted in T-revision, HOC does not need to change or adapt in different epochs and still achieves lower estimation errors no matter the model is trained with forward corrected loss or reweighted loss.

5.2. Performance of Classification Accuracy

To test the classification performance, we adopt the flow: 1) Pre-training → 2) Global Training → 3) Local Training. Our HOC estimator is applied once at the beginning of each above step. In Stage-1, we load the standard ResNet50 model pre-trained on ImageNet to obtain basic representations. At the beginning of Stage-2 and Stage-3, we use the representations given by the current model. All experiments are repeated three times. *HOC Global* only employs

Table 2. The best epoch test accuracy (%) with human noise.

Method	Clothing1M	Human CIFAR-10
CE (standard)	68.94	83.50
CORES^2 (Cheng et al., 2020)	73.24	89.98
L_{DMI} (Xu et al., 2019)	72.46	86.33
Co-teaching (Han et al., 2018)	69.21	90.39
JoCoR (Wei et al., 2020)	70.30	90.10
Forward (Patrini et al., 2017)	70.83	86.82
PTD-R-V(Xia et al., 2020b)	71.67	85.92
HOC	**73.39**	**90.62**

one global $\boldsymbol{T}$ with $G = 50$ and $|E| = 15k$ as inputs of Algorithm 2. *HOC Local* uses 300 local matrices (250-NN noise clusterability, $G = 30$, $|E| = 100$) for CIFAR-10 and 5 local matrices ($10k$-NN noise clusterability, $G = 30$, $|E| = 5k$) for CIFAR-100.[3] See more details in Appendix D. Without sophisticated learning techniques, we simply feed the estimated transition matrices given by HOC into *forward loss correction* (Patrini et al., 2017). We report the performance on synthetic instance-dependent label noise in Table 1 and real-world human-level label noise in Table 2. Comparing with these baselines (with similar data augmentations), both global estimates and local estimates given by HOC achieve satisfying performance, and the local estimates indeed provide sufficient performance improvement on CIFAR-10. When there are 100 classes, $\boldsymbol{T}$ contains $10k$ variables thus local estimates with only $10k$ instances may not be accurate, which leads to a slight performance drop in HOC Local on CIFAR-100 (but it still outperforms other methods).

Real human-level noise On CIFAR-10 with our self-collected human-level noisy labels, HOC achieves a 0.097 estimation error in the global $\boldsymbol{T}$ and a 0.110 ± 0.027 error in estimating 300 local transition matrices. See more details in Appendix D.3.

5.3. Feasibility of 2-NN label clusterability

We show the ratio of feasible 2-NN tuples in Table 3. One 2-NN tuple is called feasible if $\bar{x}_n$ and its 2-NN belong to the same true class. The feature extractors are obtained from overfitting CIFAR-10/100 with different noise levels. For example, *CIFAR-10 Inst.* $\eta = 0.2$ indicates that we use the standard CE loss to train ResNet34 on CIFAR-10 with 20% instance-dependent label noise. The convolution layers when the model approaches nearly 100% training accuracy are selected as the feature extractor $\boldsymbol{f}_{\text{conv}}(X)$. Table 3 shows, with a standard feature extractor, there are more than $2/3$ of the feasible 2-NN tuples in most cases. Besides, reducing the sample size from $50k$ to $5k$ will not substantially reduce the ratio of feasible 2-NN tuples.

[3]Our unconstrained transformation provides much better convergence such that running HOC Local on CIFAR will at most double the running time of a standard training with CE.

Table 3. The ratio of feasible 2-NN tuples with different feature extractors. $|E| = 5k$: Sample $5k$ examples from the whole dataset in each round, and average over 10 rounds. $|E| = 50k$: Check the feasibility of all 2-NN tuples.

Feature Extractor	*CIFAR-10*		*CIFAR-100*	
	$\lvert E\rvert = 5k$	$\lvert E\rvert = 50k$	$\lvert E\rvert = 5k$	$\lvert E\rvert = 50k$
Clean	99.99	99.99	99.88	99.90
Inst. $\eta = 0.2$	87.88	89.06	82.82	84.33
Inst. $\eta = 0.4$	78.15	79.85	64.88	68.31

6. Conclusions

This paper has proposed a new and flexible estimator of the noise transition matrix relying on the first-, second-, and third-order consensuses checking among an example and its' 2-NN's noisy labels. Future directions of this work include extending our estimator to collaborate with other learning with noisy label techniques. We are also interested in developing algorithms to identify critical masses of instances that share similar noise rates such that our estimator can be applied to local estimation more efficiently.

Proof Sketch for Theorem 1

The high-level idea of the proof is to connect the Hadamard products to matrix products, and prove that any linear combination of two or more rows of $\boldsymbol{T}$ does not exist in $\boldsymbol{T}$.

Step I: Transform the second-order equations. By exploiting the relation between Hadamard products and matrix products, the second-order equations can be transformed to $\boldsymbol{T}^\top \boldsymbol{D}_{\boldsymbol{p}} \boldsymbol{T} = \boldsymbol{T}_\dagger$, where $\boldsymbol{T}_\dagger$ is fixed given $\boldsymbol{c}_r^{[2]}, \forall r \in [K]$, and $\boldsymbol{D}_{\boldsymbol{p}}$ is a diagonal matrix with $\boldsymbol{p}$ as its main diagonals,

Step II: Transform the third-order equations. Following the idea in Step I, we can also transform the third-order equations to $(\boldsymbol{T} \circ \boldsymbol{T}_s) = \boldsymbol{T}\boldsymbol{T}_\dagger^{-1}\boldsymbol{T}_{\ddagger,s}^\top, \forall s \in [K]$, where $\boldsymbol{T}_{\ddagger,s}$ is fixed given $\boldsymbol{c}_{r,s}^{[3]}, \forall r, s$.

Step III: From matrices to vectors We analyze the rows $\boldsymbol{u}^\top$ of $\boldsymbol{T}$ and transform the equations in Step II to (e.g. $s = 0$) $\boldsymbol{A}\boldsymbol{u} = \boldsymbol{u} \circ \boldsymbol{u}$, where $\boldsymbol{A} = \boldsymbol{T}_\ddagger(\boldsymbol{T}_\dagger^{-1})^\top$. Then we need to find the number of feasible vectors $\boldsymbol{u}$.

Step IV: Construct the $(K+1)$-th vector When $\boldsymbol{T}$ is non-singular, we prove the $(K+1)$-th solution $\boldsymbol{u}_{K+1}$ must be identical $\boldsymbol{u}_k, k \in [K]$.

Wrapping-up: Unique $\boldsymbol{T}$ Step IV shows $\boldsymbol{T}$ only contains K different feasible rows. The informativeness of $\boldsymbol{T}$ ensures the unique order of these K rows. Thus $\boldsymbol{T}$ is unique.

Acknowledgment This work is partially supported by the National Science Foundation (NSF) under grant IIS-2007951 and the Office of Naval Research under grant N00014-20-1-22.

References

Amid, E., Warmuth, M. K., Anil, R., and Koren, T. Robust bi-tempered logistic loss based on bregman divergences. In *Advances in Neural Information Processing Systems*, pp. 14987–14996, 2019a.

Amid, E., Warmuth, M. K., and Srinivasan, S. Two-temperature logistic regression based on the tsallis divergence. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2388–2396. PMLR, 2019b.

Bengio, Y., Courville, A., and Vincent, P. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8): 1798–1828, 2013.

Berthon, A., Han, B., Niu, G., Liu, T., and Sugiyama, M. Confidence scores make instance-dependent label-noise learning possible. In *Proceedings of the 38th International Conference on Machine Learning*, ICML, 2021.

Boyd, S., Boyd, S. P., and Vandenberghe, L. *Convex optimization*. Cambridge university press, 2004.

Cheng, H., Zhu, Z., Li, X., Gong, Y., Sun, X., and Liu, Y. Learning with instance-dependent label noise: A sample sieve approach, 2020.

Feldman, V. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 954–959, 2020.

Gao, W., Yang, B.-B., and Zhou, Z.-H. On the resistance of nearest neighbor to random noisy labels. *arXiv preprint arXiv:1607.07526*, 2016.

Ghosh, A., Kumar, H., and Sastry, P. Robust loss functions under label noise for deep neural networks. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Gong, M., Li, H., Meng, D., Miao, Q., and Liu, J. Decomposition-based evolutionary multiobjective optimization to self-paced learning. *IEEE Transactions on Evolutionary Computation*, 23(2):288–302, 2018.

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 *Advances in neural information processing systems*, pp. 8527–8537, 2018.

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*, 2020.

Han, J., Luo, P., and Wang, X. Deep self-learning from noisy labels. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5138–5147, 2019.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hoeffding, W. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. ISSN 01621459.

Horn, R. A. and Johnson, C. R. *Matrix analysis*. Cambridge university press, 2012.

Ji, X., Henriques, J. F., and Vedaldi, A. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9865–9874, 2019.

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 *International Conference on Machine Learning*, pp. 2304–2313. PMLR, 2018.

Kolesnikov, A., Zhai, X., and Beyer, L. Revisiting self-supervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1920–1929, 2019.

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

Li, J., Zhang, M., Xu, K., Dickerson, J. P., and Ba, J. Noisy labels can induce good representations. *arXiv preprint arXiv:2012.12896*, 2020.

Li, X., Liu, T., Han, B., Niu, G., and Sugiyama, M. Provably end-to-end label-noise learning without anchor points. *arXiv preprint arXiv:2102.02400*, 2021.

Liu, Q., Peng, J., and Ihler, A. Variational inference for crowdsourcing. In *Proceedings of the 25th International Conference on Neural Information Processing Systems-Volume 1*, pp. 692–700, 2012.

Liu, T. and Tao, D. Classification with noisy labels by importance reweighting. *IEEE Transactions on pattern analysis and machine intelligence*, 38(3):447–461, 2015.

Liu, Y. The importance of understanding instance-level noisy labels. In *Proceedings of the 38th International Conference on Machine Learning*, ICML '21, 2021.

Liu, Y. and Chen, Y. Machine-learning aided peer prediction. In *Proceedings of the 2017 ACM Conference on Economics and Computation*, pp. 63–80, 2017.

Liu, Y. and Guo, H. Peer loss functions: Learning from noisy labels without knowing noise rates. In *Proceedings of the 37th International Conference on Machine Learning*, ICML '20, 2020.

Liu, Y. and Liu, M. An online learning approach to improving the quality of crowd-sourcing. *ACM SIGMETRICS Performance Evaluation Review*, 43(1):217–230, 2015.

Liu, Y., Wang, J., and Chen, Y. Surrogate scoring rules. In *Proceedings of the 21st ACM Conference on Economics and Computation*, pp. 853–871, 2020.

Lukasik, M., Bhojanapalli, S., Menon, A., and Kumar, S. Does label smoothing mitigate label noise? In *International Conference on Machine Learning*, pp. 6448–6458. PMLR, 2020.

Natarajan, N., Dhillon, I. S., Ravikumar, P. K., and Tewari, A. Learning with noisy labels. In *Advances in neural information processing systems*, pp. 1196–1204, 2013.

Northcutt, C., Jiang, L., and Chuang, I. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, 70:1373–1411, 2021.

Northcutt, C. G., Wu, T., and Chuang, I. L. Learning with confident examples: Rank pruning for robust classification with noisy labels. *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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1944–1952, 2017.

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

Shu, J., Zhao, Q., Chen, K., Xu, Z., and Meng, D. Learning adaptive loss for robust learning with noisy labels. *arXiv preprint arXiv:2002.06482*, 2020.

Van Rooyen, B. and Williamson, R. C. A theory of learning with corrupted labels. *J. Mach. Learn. Res.*, 18(1):8501–8550, 2017.

Wang, J., Liu, Y., and Levy, C. Fair classification with group-dependent label noise. FAccT, pp. 526–536, New York, NY, USA, 2021.

Wang, Y., Ma, X., Chen, Z., Luo, Y., Yi, J., and Bailey, J. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 322–330, 2019.

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

Wei, J. and Liu, Y. When optimizing f-divergence is robust with label noise. In *International Conference on Learning Representations*, 2021.

Wei, J., Liu, H., Liu, T., Niu, G., and Liu, Y. Understanding (generalized) label smoothing when learning with noisy labels. 2021.

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 *Advances in Neural Information Processing Systems*, pp. 6838–6849, 2019.

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*, 2020a.

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 *Advances in Neural Information Processing Systems*, volume 33, pp. 7597–7610, 2020b.

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 *International Conference on Learning Representations*, 2021.

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

Xu, Y., Cao, P., Kong, Y., and Wang, Y. L_dmi: A novel information-theoretic loss function for training deep nets robust to label noise. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Yao, Q., Yang, H., Han, B., Niu, G., and Kwok, J. T. Searching to exploit memorization effect in learning with noisy labels. In *Proceedings of the 37th International Conference on Machine Learning*, ICML '20, 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 *Advances in Neural Information Processing Systems*, volume 33, pp. 7260–7271, 2020b.

Yu, X., Han, B., Yao, J., Niu, G., Tsang, I., and Sugiyama, M. How does disagreement help generalization against

label corruption? In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 7164–7173. PMLR, 09–15 Jun 2019.

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

Zhu, Z., Liu, T., and Liu, Y. A second-order approach to learning with instance-dependent label noise. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.