Beyond Images: Label Noise Transition Matrix Estimation for Tasks with Lower-Quality Features

Zhaowei Zhu [1] **Jialu Wang** [1] **Yang Liu** [1]

Abstract

The label noise transition matrix, denoting the transition probabilities from clean labels to noisy labels, is crucial for designing statistically robust solutions. Existing estimators for noise transition matrices, e.g., using either anchor points or clusterability, focus on computer vision tasks that are relatively easier to obtain high-quality representations. We observe that tasks with lower-quality features fail to meet the anchor-point or clusterability condition, due to the coexistence of both uninformative and informative representations. To handle this issue, we propose a generic and practical information-theoretic approach to down-weight the less informative parts of the lower-quality features. This improvement is crucial to identifying and estimating the label noise transition matrix. The salient technical challenge is to compute the relevant information-theoretical metrics using only noisy labels instead of clean ones. We prove that the celebrated f-mutual information measure can often preserve the order when calculated using noisy labels. We then build our transition matrix estimator using this distilled version of features. The necessity and effectiveness of the proposed method are also demonstrated by evaluating the estimation error on a varied set of tabular data and text classification tasks with lower-quality features. Code is available at `github.com/UCSC-REAL/BeyondImages`.

1. Introduction

When a feature is not properly annotated, the returned noisy label may differ from the ground-truth one (Wei et al., 2022d). One popular and useful statistical information about noisy labels is the *label noise transition matrix* (Liu, 2022), which characterizes the transition probability from the ground-truth label to a particular noisy label. The noise transition matrix has been demonstrated to be crucial in various tasks, e.g., learning noise-tolerant classifiers (Natarajan et al., 2013), recovering unbiased estimates of fairness constraints (Lamy et al., 2019; Wang et al., 2021b), identifying mislabeled data (Northcutt et al., 2021b), and aggregating multiple labels in crowdsourcing (Liu & Liu, 2015). These applications often postulate the true knowledge of the noise transition matrix, which is generally unattainable in real-world data. Problems may arise when the noise transition matrix, acquired through an estimation process, incurs misspecification. It has been shown that the blind application of such a noise transition matrix may in fact lead to higher errors (Liu & Wang, 2021). In consequence, it is important to estimate the label noise transition matrix accurately.

Most existing approaches to estimating the label noise transition matrix presuppose *high-quality* features. For example, a series of research relies on finding the *anchor points* (Scott, 2015; Menon et al., 2015; Patrini et al., 2017; Liu & Tao, 2015; Xia et al., 2019; Yao et al., 2020; Zhang et al., 2021a; Yang et al., 2021), defined as the instances that belong to a true label class almost surely. Without high-quality features, instances belonging to different true label classes may not be well-separated in any hidden space such that anchor points rarely exist. Likewise, estimating the noise transition matrix with confident learning (Northcutt et al., 2021a) also requires high-quality features to estimate the confident points accurately.

Recent *training-free* estimator requires 2-Nearest-Neighbor (2-NN) clusterability (Zhu et al., 2021c) to estimate the noise transition matrix, where one feature and its 2-NN should have the same true label. This method inevitably depends on the quality of features since the likelihood that a data instance satisfies the clusterability condition will naturally decrease with lower-quality features.

Despite the above approaches achieving huge success on image benchmark datasets, such as CIFAR-10 (Krizhevsky et al., 2009), it remains questionable how these noise estimation approaches perform when it is hard to obtain high-

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

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

quality features. In particular, for some tabular data with categorical features like UCI datasets (Dua & Graff, 2017), popular feature learning or representation learning techniques seem unusable to improve the quality due to the sparseness of features. In more sophisticated natural language processing (NLP) tasks, such as text classification, some stop words (e.g., "a/an", "the", "is", and "are") are less informative but may be encoded in textual representations. For the learning-based methods, these less informative components may be overrated on noisy labels and harmful to finding anchor points or converging to the global optimum. For the training-free methods, these uninformative variables are likely to obscure the useful counterpart and result in mistakes in finding the nearest neighbors.

Unfortunately, we have observed consistent performance drops for the existing estimators, as shown in Figure 1. We evaluate several baselines (Xia et al., 2019; Northcutt et al., 2021a; Zhu et al., 2021c) and find the estimation errors of most approaches are around or larger than 0.1 for binary classifications with lower-quality features. As a rough comparison, the estimation errors on CIFAR-10 reported by these methods are around 0.05. Note the average noise rates of binary classification and a 10-class classification are within the range $[0, 0.5)$ and $[0, 0.9)$, respectively, showing the "same" error for binary classifications is effectively worse than a 10-class one, not to mention 0.1 for binary versus an even lower 0.05 for 10-class. Thus there is a non-negligible performance drop. We defer more detailed observations and discussions to Section 2.3.

We propose a generally practical information-theoretic approach to address the label noise matrix estimation for tasks with lower-quality features in response to the failures. Our approach builds on HOC (Zhu et al., 2021c), since we seek to avoid heavy task-specific hyperparameter fine-tuning and make it a generically applicable and light estimator. In particular, we divide the input features into several exclusively uncorrelated parts and down-weight the less informative parts by the f-mutual information between each part and the noisy labels. This operation allows us to improve the clusterability of features which proves to be a crucial property for identifying label noise transition matrix (Zhu et al., 2021c; Liu, 2022). Our main contributions are:

- Based on the f-mutual information, we propose a novel reweighting mechanism to firstly decouple features by projecting to its eigenspace and then down-weight the less-informative parts with only access to noisy labels. The mechanism intuitively disentangles features but it does not require training and can be applied efficiently.
- The salient technical challenge is to compute f-mutual information using only noisy labels instead of clean ones. We prove that calculating the total-variation-based mutual information with noisy labels preserves the same order as using clean labels (Theorem 4.4), and the traditional mutual information preserves the order when the absolute gap between two noisy measurements is larger than a guaranteed threshold (Theorem 4.5).
- We empirically demonstrate that our approach helps return a more accurate estimate of the transition matrix and reduce the classification errors of downstream learning tasks on datasets with lower-quality or more sophisticated features, including UCI datasets with tabular data and text classification benchmarks.

1.1. Related Works

The noise transition matrix $\boldsymbol{T}$ is important in several communities (Han et al., 2020). For example, it helps build noise-consistent classifiers when learning with noisy labels, such as loss correction (Natarajan et al., 2013; Patrini et al., 2017; Wei et al., 2022e), loss reweighting (Liu & Tao, 2015), and capturing the imbalance caused by heterogeneous noise rates (Zhu et al., 2021b). It also helps tune hyperparameters for label smoothing (Lukasik et al., 2020; Wei et al., 2021b), set thresholds for sample selection (Han et al., 2018; Wei et al., 2020; Zhang et al., 2021b) and detect label mistakes (Zhu et al., 2021a; Northcutt et al., 2021b). Additionally, $\boldsymbol{T}$ contributes to evaluating (Awasthi et al., 2021) or improving (Lamy et al., 2019) the model fairness when the sensitive attribute is protected, or mitigating bias in treating different groups (e.g., racial, gender) when the label quality for different groups is different (Wang et al., 2021b). It also has medical applications such as evaluating the physician variability (McCormick et al., 2016). All of the above applications require an accurate estimate of $\boldsymbol{T}$.

In addition to the $\boldsymbol{T}$ estimators introduced in Section 1, there are related works in crowdsourcing (Liu et al., 2012; Zhang et al., 2014; Liu & Liu, 2015) and peer prediction (Liu & Chen, 2017; Liu et al., 2020). However, these works require redundant noisy labels. For a general machine learning task, the datasets that have only one noisy label for each feature are more common. Compared with these approaches, the 2-NN clusterability of HOC (Zhu et al., 2021c) can be treated as a proxy of two redundant noisy labels, where a better proxy requires well-extracted features. According to the analyses on the identifiability of the label noise transition matrix (Liu, 2022), the disentangled and informative features are crucial. It has been demonstrated that mutual information helps select more informative features (Battiti, 1994; Estévez et al., 2009; Vergara & Estévez, 2014) with clean data. Recent work (Wei & Liu, 2020) finds that some f-mutual information metrics are robust to label noise, which makes the feature selection on noisy data promising.

2. Preliminaries

In this section, we first formally define the noise transition matrix $\boldsymbol{T}$ and pose the problem in Section 2.1. As a comple-

ment to related works, we introduce more technical details of two lines of most relevant $\boldsymbol{T}$ estimators in Section 2.2 and numerically show the possible failures of these approaches in Section 2.3.

2.1. The Definition of Noise Transition Matrix

Clean/Noisy distribution Consider a K-class classification task with a dataset $\widetilde{D} := \{\boldsymbol{x}_n, \tilde{y}_n\}_{n\in[N]}$, where $\boldsymbol{x}_n$ is the feature, $\tilde{y}_n$ is the noisy label that may come from human annotations (Wei et al., 2022c;d; Luo et al., 2020), sensors (Wang et al., 2021a) and machine pseudo labels (Zhu et al., 2022), N is the number of instances, $[N] := \{1, 2, \cdots, N\}$. Suppose $\boldsymbol{x}$ is d-dimensional, i.e., $\boldsymbol{x} = [x_1, \cdots, x_d]^\top$. We denote the μ-th element by *feature variable* x_μ. Note the *feature vector* $\boldsymbol{x}$ is not necessary to be the raw input for some complicated tasks that require deep neural networks. In these tasks, the feature vector $\boldsymbol{x}$ should be the output of some feature extractors (Devlin et al., 2019; Radford et al., 2021). We will *not* discuss methods to get appropriate feature extractors in this paper since the focus is on the data processing given $\widetilde{D}$. The clean label associated with the noisy label $\tilde{y}$ is denoted by y. Both clean labels and noisy labels are in the same label space, i.e., $y \in [K]$, $\tilde{y} \in [K]$, where $[K] := \{1, 2, \cdots, K\}$. The *random variable* forms of the above realizations of features and labels are: feature vector $\boldsymbol{x} \sim \boldsymbol{X} := [X_1, \cdots, X_d]^\top$, feature variable $x_\mu \sim X_\mu$, clean label $y \sim Y$, noisy label $\tilde{y} \sim \widetilde{Y}$. The (unobservable) clean dataset is denoted by D.

Noise transition matrix $\boldsymbol{T}$ The relationship between $(\boldsymbol{X}, Y)$ and $(\boldsymbol{X}, \widetilde{Y})$ is denoted by a noise transition matrix $\boldsymbol{T}(\boldsymbol{X})$, where each element $T_{ij}(\boldsymbol{X}) := \mathbb{P}(\widetilde{Y} = j|Y = i, \boldsymbol{X})$ stands for the probability of mislabeling a clean label $Y = i$ as the noisy label $\widetilde{Y} = j$ given feature $\boldsymbol{X}$. A majority line of works assume the noise transition matrix is *class-dependent* (Natarajan et al., 2013; Liu & Tao, 2015; Patrini et al., 2017; Liu & Guo, 2020), i.e., $\boldsymbol{T}(\boldsymbol{X}) \equiv \boldsymbol{T}$, implying the following independency:

$$\mathbb{P}(\widetilde{Y} = j|Y = i, \boldsymbol{X}) = \mathbb{P}(\widetilde{Y} = j|Y = i), \forall i, j \in [K], \forall \boldsymbol{X}.$$

Problem Statement We formally frame the problem of label noise transition matrix estimation. The learner can only access noisy examples $\widetilde{D}$ in this setting. The noisy label $\widetilde{Y}$ satisfies an underlying transition probability characterized by $\boldsymbol{T}$. The goal is to minimize the estimation error calculated by the *average total variation* of the true $\boldsymbol{T}$ and the estimated $\hat{\boldsymbol{T}}$ (Zhang et al., 2021a):

$$\mathsf{Error}(\boldsymbol{T}, \hat{\boldsymbol{T}}) = \sum_{i\in[K], j\in[K]} |T_{ij} - \hat{T}_{ij}|/(2K). \quad (1)$$

2.2. Existing Estimation Approaches

There are mainly two categories of prior works on estimating the noise transition matrix $\boldsymbol{T}$. One popular solution pipeline is to estimate $\boldsymbol{T}$ with the confidence/predictions of the model *trained on the noisy data distribution*. As an alternative to the learning-based methods, one recent work (Zhu et al., 2021c) proposes a *training-free* pipeline when *clusterable* features are given. We introduce more details as follows.

Definition 2.1 (Anchor Point (Liu & Tao, 2015; Scott, 2015; Xia et al., 2019))**.** An instance $\boldsymbol{x}$ is an anchor point for class-i if $\mathbb{P}(Y = i|\boldsymbol{X} = \boldsymbol{x})$ is equal or close to 1.

Using model predictions In the learning-based methods, the anchor point, which is the instance that belongs to a particular class almost surely as defined in Definition 2.1, plays a significant role. Particularly, if $\mathbb{P}(Y = i|\boldsymbol{X} = \boldsymbol{x}) = 1$ holds for some class i, we have $\mathbb{P}(\widetilde{Y} = j|\boldsymbol{X} = \boldsymbol{x}) = \sum_{k\in[K]} T_{kj}\mathbb{P}(Y = k, \boldsymbol{X} = \boldsymbol{x}) = T_{ij}$. Note the anchor point is defined on the clean data. If the features $\boldsymbol{X}$ are of lower quality, the condition $\mathbb{P}(Y = i|\boldsymbol{X} = \boldsymbol{x}) = 1$ is hard to satisfy (Zhu et al., 2021a), not to mention finding anchor points accurately. Although recent advances relax the requirement of anchor points (Xia et al., 2019; Zhang et al., 2021a; Li et al., 2021), they tend to design a specific learning pipeline with a particular regularizer or objective related to $\boldsymbol{T}$, which is sensitive to hyperparameters and ultimately the quality of features.

Using clusterability Recent work HOC (Zhu et al., 2021c) studies this problem from a novel data-centric perspective, which proposes a statistical solution based on clusterability without fitting the data distribution. The main idea is to utilize a set of High-Order Consensus (HOC) information aggregated on the nearest neighbors' noisy labels and solve a series of equations for the noise transition matrix $\boldsymbol{T}$ and clean label prior probability $\boldsymbol{p}$. The effectiveness of this approach relies on the k-NN (k-Nearest-Neighbor) label clusterability, which is defined as:

Definition 2.2 (k-NN label clusterability (Zhu et al., 2021c))**.** A dataset D satisfies k-NN label clusterability if $\forall n \in [N]$, the feature $\boldsymbol{x}_n$ and its k-Nearest-Neighbor $\boldsymbol{x}_{n_1}, \cdots, \boldsymbol{x}_{n_k}$ belong to the same true label class.

The distance between two features $\boldsymbol{x}$ and $\boldsymbol{x}'$ can be measured by $1 - \mathsf{Sim}(\boldsymbol{x}, \boldsymbol{x}')$, where $\mathsf{Sim}(\boldsymbol{x}, \boldsymbol{x}')$ could be the cosine similarity. It has been proved by (Zhu et al., 2021c) that the 2-NN label clusterability is sufficient for uniquely getting the true $\boldsymbol{T}$. However, it is likely that the clusterability is not sufficiently satisfied for lower-quality features.

2.3. Failures on Lower-Quality Features

The above approaches to estimating $\boldsymbol{T}$ are demonstrated to perform well on some image classification datasets, such as MNIST (LeCun et al., 1998) and CIFAR (Krizhevsky et al., 2009), which usually enjoy better representation learning tools (Chen et al., 2020; Wang et al., 2022a) that would return clusterable features, as compared to text sequence

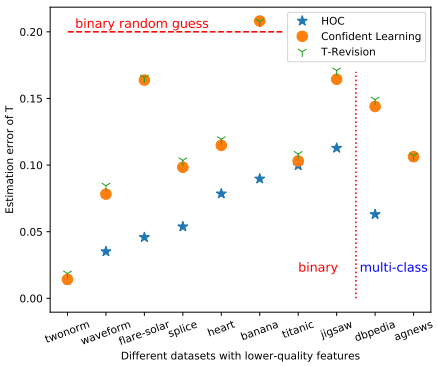

Figure 1. Existing methods may suffer from failures. **Red** horizontal dashed line shows the error of random guessing $\boldsymbol{T}$ in binary classifications. Tasks on the left side of the dotted line are binary.

and tabular data. When facing other tasks, high-quality features may not always be available. In this subsection, we explore how these existing methods fare on other datasets, possibly with lower-quality features.

Observations In Figure 1, we implement two learning-based methods built on anchor points (T-revision (Xia et al., 2019)) or confident points (Confident Learning (Northcutt et al., 2021a)), and one training-free method based on the clusterability (HOC (Zhu et al., 2021c)). The average noise rate, i.e., $\sum_{i\in[K]}(1-T_{ii})/K$, is around 0.3. Figure 1 shows the estimation error of these three methods on most of the datasets are around or larger than 0.1, and some methods may even approximate to 0.2. Note **an error of** 0.2 **is excessive** for binary classification when the noise rate is 0.3. For example, when $T_{11} = T_{22} = 0.3$, a *random guess* of $\boldsymbol{T}$, i.e., $T_{ij} = 0.5, \forall i, j$, has an error of exactly 0.2. Recall an error of 0.05 for binary classification is worse than the same error for a 10-class one as explained in Introduction. Compared with an estimation error of ≈ 0.05 on CIFAR-10 with a similar noise rate as reported in these baselines, Figure 1 shows a serious performance drop: an error of 0.05 for 83% of tests and an error of 0.1 for 57% of tests. Therefore, it is crucial to design an estimator which is also robust on datasets with lower-quality features.

Discussions Before diving into a concrete solution to lower-quality features, we discuss the advantages and disadvantages of both existing lines of work. On one hand, the *learning-based methods* (Northcutt et al., 2021a; Xia et al., 2019) could take full advantage of deep neural networks. During supervised training, different parts of features are weighted differently, thus the informative parts could weigh more than the less informative ones. The optimal weight combinations could induce a model that accurately fits the data distribution, which further help estimate $\boldsymbol{T}$. On the other hand, due to various factors such as the model capacity, the quality of features, the number of instances, and the setting of hyperparameters, the learning-based models are often infeasible to converge to the global optimum in practice. For example, with the existence of label noise, deep neural networks (DNN) tend to be overconfident (Wei et al., 2022b) and memorize wrong feature-label patterns (Cheng et al., 2021b; Liu, 2021; Wei et al., 2021a). When unintended memorization occurs, the weights for combining different parts will be non-optimal and some uninformative parts may be mis-specified with high weights. Alternatively, the *training-free method* (Zhu et al., 2021c) will not be affected by the wrong memorization since it is a fully-statistical solution without any training procedures. Nevertheless, the problem of not employing training is also severe: blindly treating different parts of features equally important may cause failures. The above observations and discussions motivate us to find a solution that *compromises between the learning-based and the training-free approaches*.

Algorithm 1 Key Steps of HOC

Input: Noisy dataset: $\widetilde{D} = \{(\boldsymbol{x}_n, \tilde{y}_n)\}_{n\in[N]}$.
// Find 2-NN. $\mathsf{Sim}(\boldsymbol{x}, \boldsymbol{x}') \to \mathsf{Sim}_{\boldsymbol{W}}(\boldsymbol{z}, \boldsymbol{z}')$ in our approach.
With $1 - \mathsf{Sim}(\boldsymbol{x}, \boldsymbol{x}')$ as the distance metric:
$\{(\tilde{y}_n, \tilde{y}_{n_1}, \tilde{y}_{n_2}), \forall n\} \leftarrow \texttt{Get2NN}(\widetilde{D})$;
// Count first-, second, and third-order consensus patterns:
$(\hat{\boldsymbol{c}}^{[1]}, \hat{\boldsymbol{c}}^{[2]}, \hat{\boldsymbol{c}}^{[3]}) \leftarrow \texttt{CountFreq}(\{(\tilde{y}_n, \tilde{y}_{n_1}, \tilde{y}_{n_2}), \forall n\})$
// Solve equations
Find $\boldsymbol{T}$ such that match the counts $(\hat{\boldsymbol{c}}^{[1]}, \hat{\boldsymbol{c}}^{[2]}, \hat{\boldsymbol{c}}^{[3]})$.

3. An Information-Theoretic Approach

We propose an information-theoretic approach to distinguish the importance of different features. To avoid complicated hyperparameters tuning and make it a light tool for more general applications, the solution is built on HOC. See more detailed rationale in Appendix B.1.

We now briefly introduce HOC (Zhu et al., 2021c). Algorithm 1 summarizes the key steps, where the high-level idea is that, when 2-NN label clusterability holds, the frequency of consensus patterns of the three grouped noisy labels $\tilde{y}_n, \tilde{y}_{n_1}, \tilde{y}_{n_2}$ encodes $\boldsymbol{T}$.[1] For instance, a triplet $\tilde{y}_n = 0, \tilde{y}_{n_1} = 0, \tilde{y}_{n_2} = 1$ will add 1 to the count of the consensus pattern $(0, 0, 1)$. With the estimated frequency of patterns $\{(\tilde{y}_n, \tilde{y}_{n_1}, \tilde{y}_{n_2}), \forall n\}$, we can simply solve equations by gradient descents. Therefore, in Algorithm 1, the 2-NN label clusterability is critical, which depends heavily on calculating the distance or similarity between features.

Overview of our approach: The main idea is to decouple features (Step 1) and down-weight the less informative parts (Step 2) when measuring the distances by HOC, which is summarized in Algorithm 2. Firstly, we motivate the

[1]It is shown later in (Liu, 2022) that three noisy labels are necessary and sufficient to identify $\boldsymbol{T}$.

Algorithm 2 Our Information-Theoretic Approach

Input: Noisy dataset: $\widetilde{D} = \{(\boldsymbol{x}_n, \tilde{y}_n)\}_{n\in[N]}$.
// Step 1: Remove correlation (Section 3.3)
Transform $\boldsymbol{x} \sim \boldsymbol{X}$ to $\boldsymbol{z} \sim \boldsymbol{Z}$ by Eqn. (3);
// Step 2: Estimate the weight matrix $\boldsymbol{W}$ (Section 3.4)
With only noisy labels:
Diagonal elements: $\hat{W}_{\mu\mu} = I_f(Z;\widetilde{Y})$ by Eqn. (4)
Off-diagonal elements: $\hat{W}_{\mu\mu'} = 0, \forall \mu \neq \mu'$;
// Step 3: Estimate the noise transition matrix
Apply HOC (Zhu et al., 2021c) with soft cosine similarity $\mathsf{Sim}_{\boldsymbol{W}}(\boldsymbol{z}, \boldsymbol{z}')$ defined in Eqn. (2).
Output: The estimated noise transition matrix $\hat{\boldsymbol{T}}$.

necessity of down-weighting less informative parts in Section 3.1. Their weights are controlled by a matrix $\boldsymbol{W}$, which is absorbed into the calculation of soft cosine similarity (Definition 3.1). A tractable proxy of $\boldsymbol{W}$ is introduced in Section 3.2 and detailed in the remainder of this section.

3.1. Vanilla Similarity Measures Are Not Sufficient

Our following analyses focus on the cosine similarity as originally implemented by HOC (Zhu et al., 2021c). The larger cosine similarity implies the smaller distances of features. We first analyze possible problems in evaluating k-NN with the vanilla (hard) cosine similarity.

Consider the cosine similarity of two feature vectors $\boldsymbol{x}$ and $\boldsymbol{x}'$, which is denoted by $\mathsf{Sim}(\boldsymbol{x}, \boldsymbol{x}') = \frac{\boldsymbol{x}^\top \boldsymbol{x}'}{\|\boldsymbol{x}\|_2 \|\boldsymbol{x}'\|_2}$, where $\|\cdot\|_2$ denotes the vector ℓ_2 norm. The above measure inherently assumes different elements of $\boldsymbol{x}$ are 1) *equally important* and 2) *uncorrelated* to each other, thus may underestimate the true similarity between imperfect feature vectors. To capture more information, we incorporate a change-of-basis matrix $\sqrt{\boldsymbol{W}}$ to obtain a soft cosine measure defined as follows.

Definition 3.1 (Soft cosine similarity (Sidorov et al., 2014))**.**

$$\mathsf{Sim}_{\boldsymbol{W}}(\boldsymbol{x}, \boldsymbol{x}') = \frac{(\sqrt{\boldsymbol{W}}\boldsymbol{x})^\top (\sqrt{\boldsymbol{W}}\boldsymbol{x}')}{\|\sqrt{\boldsymbol{W}}\boldsymbol{x}\|_2 \|\sqrt{\boldsymbol{W}}\boldsymbol{x}'\|_2}. \tag{2}$$

Hereby, the symmetric matrix $\boldsymbol{W}$ encodes the pairwise similarity between features. Note that the soft cosine similarity measure $\mathsf{Sim}_{\boldsymbol{W}}(\boldsymbol{x}, \boldsymbol{x}')$ recovers the (hard) cosine similarity when $\boldsymbol{W} = \boldsymbol{I}$, where $\boldsymbol{I}$ denotes a $K \times K$ identity matrix. In practice, the true and unknown $\boldsymbol{W}$ may be very different from $\boldsymbol{I}$. Thus simply letting $\boldsymbol{W} = \boldsymbol{I}$ may cause severe problems in using clusterability. For example, when $K = 2$, consider three instances $(\boldsymbol{x}_1, y_1) = ([1, 0, 1]^\top, 1)$, $(\boldsymbol{x}_2, y_2) = ([0, 1, 0]^\top, 2)$, and $(\boldsymbol{x}_3, y_3) = ([0.8, 1, 0.7]^\top, y)$. Based on 1-NN label clusterability, we infer label y following the rule below:

$$y = \begin{cases} 1 & \text{if} \quad \mathsf{Sim}(\boldsymbol{x}_1, \boldsymbol{x}_3) > \mathsf{Sim}(\boldsymbol{x}_2, \boldsymbol{x}_3); \\ 2 & \text{if} \quad \mathsf{Sim}(\boldsymbol{x}_1, \boldsymbol{x}_3) \leq \mathsf{Sim}(\boldsymbol{x}_2, \boldsymbol{x}_3). \end{cases}$$

Consider the following three $\boldsymbol{W}$s: $\boldsymbol{W}_1 = \boldsymbol{I}$,

$$\boldsymbol{W}_2 = \begin{pmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & \mathbf{0.1} \end{pmatrix}, \quad \boldsymbol{W}_3 = \begin{pmatrix} 1.0 & -0.2 & -0.5 \\ -0.2 & 1.0 & 0.5 \\ -0.5 & 0.5 & 1.0 \end{pmatrix}.$$

Example 1: $\boldsymbol{W}_1$ (uncorrelated and equally important)
The following hard cosine similarity shows:
$\mathsf{Sim}_{\boldsymbol{W}_1}(\boldsymbol{x}_1, \boldsymbol{x}_3) \approx 0.73 > \mathsf{Sim}_{\boldsymbol{W}_1}(\boldsymbol{x}_2, \boldsymbol{x}_3) \approx 0.69 \Rightarrow y = 1$.

Example 2: $\boldsymbol{W}_2$ (not equally important)
The following soft cosine similarity shows:
$\mathsf{Sim}_{\boldsymbol{W}_2}(\boldsymbol{x}_1, \boldsymbol{x}_3) \approx 0.64 < \mathsf{Sim}_{\boldsymbol{W}_2}(\boldsymbol{x}_2, \boldsymbol{x}_3) \approx 0.77 \Rightarrow y = 2$, which is *different* from the inferred y in Example 1. If $\boldsymbol{W}_2$ defines the true clusterability, this example shows that simply using $\boldsymbol{W} = \boldsymbol{I}$ fails to capture the diagonal values of $\boldsymbol{W}$ (the weights of x_μ, $\mu \in [d]$) and violates the clusterability.

Example 3: $\boldsymbol{W}_3$ (correlated)
The following soft cosine similarity shows:
$\mathsf{Sim}_{\boldsymbol{W}_3}(\boldsymbol{x}_1, \boldsymbol{x}_3) \approx 0.75 < \mathsf{Sim}_{\boldsymbol{W}_3}(\boldsymbol{x}_2, \boldsymbol{x}_3) \approx 0.85 \Rightarrow y = 2$, which is *different* from the inferred y in Example 1. If $\boldsymbol{W}_3$ is true, this example shows that simply using $\boldsymbol{W} = \boldsymbol{I}$ fails to capture the off-diagonal values of $\boldsymbol{W}$ (the correlations between x_μ) and violates the clusterability.

The above three examples exemplify that the less informative parts of features may damage the clusterability, where the definition of each "part" depends on both the diagonal elements and off-diagonal elements of $\boldsymbol{W}$. We propose an information-theoretic approach to construct a proxy of $\boldsymbol{W}$.

3.2. Proxy of $\boldsymbol{W}$

Noting $\boldsymbol{W}$ is a square matrix of order d, it requires $\mathcal{O}(d^2)$ operations to estimate all elements. Each operation incurs an estimation error, and the accumulated errors may not be bounded. We propose to find a proxy of $\boldsymbol{W}$. Intuitively, we expect the following properties (Battiti, 1994):

- **Symmetric**: $\forall \mu, \nu \in [d], W_{\mu\nu} = W_{\nu\mu}$.
- **Information monotone**: For every two feature variables X_μ, X_ν, if X_μ is less informative with respect to Y than X_ν, then X_μ will be less important than X_ν, i.e., $I_f(X_\mu; Y) \leq I_f(X_\nu; Y) \Rightarrow W_{\mu\mu} \leq W_{\nu\nu}, \mu, \nu \in [d]$, where $I_f(X_\mu; Y)$ measures the f-MI (f-Mutual Information) (Csiszár, 1967) between X_μ and Y.
- **Correlation monotone**: Given two feature variables X_μ, X_ν, for any other feature variable $X_{\nu'}$ ($X_\mu, X_\nu, X_{\nu'}$ are equally informative), if X_μ is less correlated to X_ν than $X_{\nu'}$, X_ν will have a lower weight than $X_{\nu'}$ in mea-

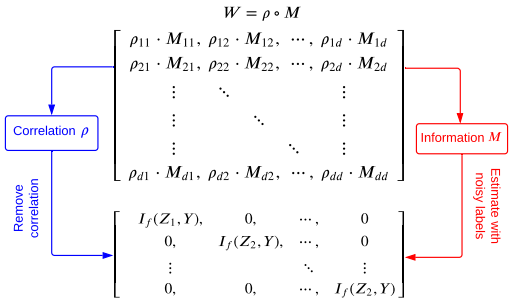

Figure 2. Illustration of the proxy of $\boldsymbol{W}$.

suring the similarity with X_ν:

$$\rho(X_\mu, X_\nu) \leq \rho(X_\mu, X_{\nu'}) \Rightarrow W_{\mu\nu} \leq W_{\mu\nu'},$$

where $\rho(X_\mu, X_\nu)$ is the correlation between random variables X_μ and X_ν.

The above properties suggest us to decompose the elements of matrix $\boldsymbol{W}$ as the product of the informativeness w.r.t Y and the correlation between features as illustrated in Figure 2. In another word, we can construct the matrix $\boldsymbol{W} = \boldsymbol{\rho} \circ \boldsymbol{M}$, where $\circ$ denotes the Hadamard product of two matrices, $\boldsymbol{\rho}$ is the correlation matrix with $\rho_{\mu\nu} = \rho(X_\mu, X_\nu)$, $\boldsymbol{M}$ includes the f-MI with $M_{\mu\nu} = \sqrt{I_f(X_\mu, Y) I_f(X_\nu, Y)}$. Directly estimating $\boldsymbol{\rho}$ might not be computation-efficient. If we can transform $\boldsymbol{X}$ to make $\boldsymbol{\rho} = \boldsymbol{I}$, the off-diagonal entries of $\boldsymbol{W}$ will become 0 thus no longer need to be estimated. This observation motivates us to firstly transform $\boldsymbol{X}$ to a non-correlated form to achieve $\boldsymbol{\rho} = \boldsymbol{I}$ (Section 3.3), then estimate only the diagonal elements by f-MI (Section 3.4).

3.3. Remove correlation

For ease of notations, we require that $\boldsymbol{X}$ is zero-mean, i.e., $\mathbb{E}[\boldsymbol{X}] = \boldsymbol{0}$, where $\boldsymbol{0}$ is a $d \times 1$ column vector with all elements being 0. Inspired by the principle component analysis (PCA) (Wold et al., 1987), we adopt $\boldsymbol{\Lambda}^{-1/2}\boldsymbol{P}^\top$ as the matrix to remove the correlation between $x_\mu, x_\nu, \mu \neq \nu$, where $\boldsymbol{P}\boldsymbol{\Lambda}\boldsymbol{P}^\top = \mathbb{E}[\boldsymbol{X}\boldsymbol{X}^\top]$, $\boldsymbol{P}$ is an orthogonal matrix whose columns are eigenvectors of $\mathbb{E}[\boldsymbol{X}\boldsymbol{X}^\top]$, $\boldsymbol{\Lambda}$ is a diagonal matrix where diagonal values are eigen values and off-diagonal values are 0. Let

$$\boldsymbol{Z} = \boldsymbol{\Lambda}^{-1/2}\boldsymbol{P}^\top \boldsymbol{X} = [Z_1, \cdots, Z_d]^\top, \tag{3}$$

where $\mathbb{E}[\boldsymbol{Z}] = \boldsymbol{\Lambda}^{-1/2}\boldsymbol{P}^\top \mathbb{E}[\boldsymbol{X}] = \boldsymbol{0}$. We have

$$\mathbb{E}[\boldsymbol{Z}\boldsymbol{Z}^\top] = \boldsymbol{\Lambda}^{-1/2}\boldsymbol{P}^\top \mathbb{E}[\boldsymbol{X}\boldsymbol{X}^\top]\boldsymbol{P}\boldsymbol{\Lambda}^{-1/2} = \boldsymbol{I}.$$

Therefore, after transforming $\boldsymbol{X}$ to $\boldsymbol{Z}$ by $\boldsymbol{\Lambda}^{-1/2}\boldsymbol{P}^\top$, we can split $\boldsymbol{X}$ into d uncorrelated parts such that $\mathbb{E}[Z_\mu Z_\nu]$ satisfies

$$\mathbb{E}[Z_\mu Z_\nu] = \mathbb{E}[(\boldsymbol{Z}\boldsymbol{Z}^\top)_{\mu,\nu}] = \begin{cases} 0 & \text{if } \mu \neq \nu, \\ 1 & \text{if } \mu = \nu. \end{cases}$$

Noting (zero-mean) $\rho(Z_\mu, Z_\nu) = \frac{\mathbb{E}[Z_\mu Z_\nu]}{\sqrt{\mathbb{E}[Z_\mu^2]}\sqrt{\mathbb{E}[Z_\nu^2]}}$, we know $\boldsymbol{\rho} = \boldsymbol{I}$ after this transformation.

3.4. Estimate $I_f(Z;Y)$ With Only Noisy Labels

After transforming $\boldsymbol{X}$ to $\boldsymbol{Z}$, the off-diagonal elements of the proxy of $\boldsymbol{W}$ are expected to be 0. Thus we only need to estimate the diagonal elements by $W_{\mu\mu} := I_f(Z_\mu; Y)$.

Define an f-MI metric with f-divergence $I_f(Z;Y) := D_f(Z \oplus Y; Z \otimes Y)$, where $D_f(P||Q)$ measures the f-divergence between two distributions P and Q with probability density function p and q:

$$D_f(P||Q) = \int_{v \in \mathcal{V}} q(v) f\left(\frac{p(v)}{q(v)}\right) dv.$$

Note the variable v in the domain $\mathcal{V}$ is a realization of random variable $V = (Z, Y)$, which follows either distribution P or distribution Q depending on where it is used. In our setting, P and Q are the joint distribution $P := Z \oplus Y = \mathbb{P}(Z = z, Y = y)$ and the marginal product distribution $Q := Z \otimes Y = \mathbb{P}(Z = z) \cdot \mathbb{P}(Y = y)$, respectively. There are lots of choices of f. Specially, when $f(v) = v \log v$, the f-divergence becomes the celebrated KL divergence and I_f is exactly the mutual information.

Alternatively, we can calculate the f-divergence using the variational form (Nowozin et al., 2016; Wei & Liu, 2020). Denote the variational difference between P and Q by

$$\mathsf{VD}_f(g) = \mathbb{E}_{V \sim P}[g(V)] - \mathbb{E}_{V \sim Q}[f^*(g(V))], \forall g,$$

where $g : \mathcal{V} \to \text{domain}(f^*)$ is a variational function, and f^* is the conjugate function of $f(\cdot)$. We instantiate some prominent variational-conjugate (g^*, f^*) pairs in Appendix A.1 (Table 4). The f-MI is calculated by

$$I_f(Z;Y) = D_f(P||Q) = \mathsf{VD}_f(g^*) = \sup_g \mathsf{VD}_f(g).$$

However, the above calculation, built on the clean distributions, will be intractable when we can only access the noisy data distribution. Let $\widetilde{P} := Z \oplus \widetilde{Y}$ and $\widetilde{Q} := Z \otimes \widetilde{Y}$. One tractable approach is to calculate $D_f(\widetilde{P}||\widetilde{Q})$ following

$$I_f(Z;\widetilde{Y}) = D_f(\widetilde{P}||\widetilde{Q}) = \widetilde{\mathsf{VD}}_f(\tilde{g}^*) = \sup_g \widetilde{\mathsf{VD}}_f(g), \tag{4}$$

where $\widetilde{\mathsf{VD}}_f(g) = \mathbb{E}_{\widetilde{V} \sim \widetilde{P}}[g(\widetilde{V})] - \mathbb{E}_{\widetilde{V} \sim \widetilde{Q}}[f^*(g(\widetilde{V}))], \forall g$. Generally, there will be a gap between our calculated $I_f(Z;\widetilde{Y})$ and the real $I_f(Z;Y)$. We defer detailed analyses of the gap to Section 4.

4. Theoretical Guarantees

The quality of our $\boldsymbol{T}$ estimator relies on the following steps:

(a) Noise-resistant estimates of f-MI using noisy labels;
(b) Accurate estimates of clean f-MI;
(c) Down-weighting less informative features with f-MI;
(d) Robust distance/similarity calculation;

(e) Satisfying clusterability (Definition 2.2);
(f) Accurate estimates of the noise transition matrix $\boldsymbol{T}$.

The most critical step in the above chain is (a)→(b), which is the key ingredient to Step (c). This step will be the focus of the theoretical results in this section. Steps (c)→(e) are explained in Section 3. Steps (e)→(f) is guaranteed by HOC (Zhu et al., 2021c).

Based on our intuition for constructing the proxy of $\boldsymbol{W}$ that the less informative parts should be assigned with lower weights, the order between two parts with different informativeness is crucial. Thus in this section, we study whether the noisy f-MI calculated using noisy labels, i.e., $I_f(Z;\widetilde{Y})$, preserves the order of the clean f-MI $I_f(Z;Y)$. Note the order-preservation property distinguishes from the robustness of f-MI between optimal classifier prediction $h^*(X)$ and noisy label $\widetilde{Y}$ by Wei & Liu (2020) since 1) Z does not have class-specific meaning; 2) Z is not optimizable and its ranking is concerned, while $h^*(X)$ is only a special (optimal) case of Z. All proofs are deferred to Appendix A. We define ϵ-order-preserving as follows.

Definition 4.1 (ϵ-Order-Preserving Under Label Noise)**.** $I_f(Z;\widetilde{Y})$ is called ϵ-order-preserving under label noise if $\forall \mu \in [d], \nu \in [d]$, given $|I_f(Z_\mu;\widetilde{Y}) - I_f(Z_\nu;\widetilde{Y})| > \epsilon$, we have

$$\text{sign}[I_f(Z_\mu;\widetilde{Y}) - I_f(Z_\nu;\widetilde{Y})] = \text{sign}[I_f(Z_\mu;Y) - I_f(Z_\nu;Y)].$$

The smaller ϵ is, the stricter the requirement is. The following analyses focus on the binary classification. Define $e_1 := \mathbb{P}(\widetilde{Y} = 2|Y = 1)$, $e_2 := \mathbb{P}(\widetilde{Y} = 1|Y = 2)$. To show an f-MI metric is ϵ-order-preserving under label noise, we need to study how $\widetilde{\mathsf{VD}}_f(\tilde{g}^*)$ differs from the order of $\mathsf{VD}_f(g^*)$.

4.1. Total-Variation is 0-Order-Preserving

When $f(v) = \frac{1}{2}|v-1|$, we get the Total-Variation (TV). To analyze the order-preserving property of TV, we first build the relationship between $\widetilde{\mathsf{VD}}_f(g)$ and $\widetilde{\mathsf{VD}}_f(g)$, $\forall g$ by the following lemma:

Lemma 4.2 (Linear relationship (Wei & Liu, 2020))**.**

$$\widetilde{\mathsf{VD}}_{TV}(g) = (1 - e_1 - e_2)\,\mathsf{VD}_{TV}(g), \forall g.$$

Lemma 4.2 shows that there is a linear relationship between $\widetilde{\mathsf{VD}}_{\text{TV}}(g)$ and $\mathsf{VD}_{\text{TV}}(g)$. The constant only depends on the noise rates. With this lemma, we only need to study the difference between $\mathsf{VD}_{\text{TV}}(\tilde{g}^*)$ and $\mathsf{VD}_{\text{TV}}(g^*)$, which is summarized in the following lemma:

Lemma 4.3. *When $e_1 + e_2 < 1$, $\mathsf{VD}_{TV}(\tilde{g}^*) = \mathsf{VD}_{TV}(g^*)$.*

Note the condition $e_1 + e_2 < 1$ indicates the label noise is not too large to be dominant (Liu & Guo, 2020; Natarajan et al., 2013; Liu & Chen, 2017). With Lemma 4.2 and Lemma 4.3, we can conclude that TV is 0-order-preserving.

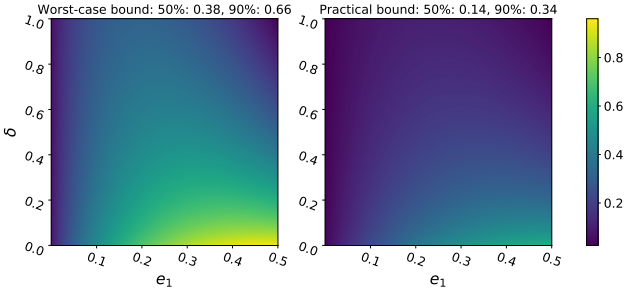

Figure 3. Illustration of the worst-case bound and a more practical bound for ϵ with different $e_1, \delta := e_2/e_1$. Color indicates the value of ϵ. The median (50%) and the 90-th percentile of ϵ are shown in the title of each plot. $\log_2(x)$ is applied for calculating mutual information.

Theorem 4.4. *When $e_1 + e_2 < 1$, total-variation is 0-order-preserving under class-dependent label noise.*

Theorem 4.4 shows the total-variation-based mutual information under class-dependent label noise preserves the order of the original clean results.

4.2. KL Divergence is ϵ-Order-Preserving

Unfortunately, Lemma 4.2 and Lemma 4.3 do not hold for KL divergence. Recall the f-MI is exactly the standard mutual information when KL divergence is adopted. Denote by $H(\cdot)$ the entropy. We can start from the definition of mutual information $I(Z,\widetilde{Y}) = H(Z) + H(\widetilde{Y}) - H(Z,\widetilde{Y})$ and decouple the effect of noisy labels as:

$$I(Z,\widetilde{Y}) = (1 - e_1 - e_2) \cdot [I(Z,Y) - H(Y)] + H(\widetilde{Y}) + \int_z \Delta_{\text{Bias}}(\beta_z, e_1, e_2)\, dz,$$

where β_z is a function of z, and $\Delta_{\text{Bias}}(\beta_z, e_1, e_2)$ is the bias caused by label noise specified in Eqn. (7). We then find the lower and upper bounds for $\Delta_{\text{Bias}}(\beta_z, e_1, e_2)$ and summarize the result as follows:

Theorem 4.5. *Assume $e_1 = \delta e_2$, $\delta \in [0,1]$ and $e_1 + e_2 < 1$. KL divergence (mutual information) is ϵ-order-preserving under class-dependent label noise, where*

$$\epsilon = e_1\,[\delta \log \delta - (1+\delta)\log(1+\delta)] + H(e_1),$$

and $H(e_1) := -e_1 \log e_1 - (1-e_1)\log(1-e_1)$.

For the symmetric label noise, we have:

Corollary 4.6. *When $e_1 = e_2 < 0.5$, KL divergence (mutual information) is $[H(e_1) - 2e_1 \log 2]$-order-preserving under class-dependent label noise.*

We evaluate the bound in the following remark.

Remark 4.7. The value of ϵ is illustrated in Figure 3, where the left figure shows the worst-case bound in Theorem 4.5, and the right figure shows the case when $\frac{\mathbb{P}(Y=1|Z=z)}{\mathbb{P}(Y=2|Z=z)} \in [\frac{1}{5}, 5]$. This is a more practical case for lower-quality features since a single feature variable cannot infer the clean label with high confidence. Noting the $\log_2$-based mutual information is within range $[0, 1]$ and $\epsilon \leq 0.34$ happens in 90% of the cases, it is reasonable to believe the mutual information has a good ϵ-order-preserving ability.

5. Evaluations

We evaluate our approaches on datasets with possibly lower-quality features in this section. The evaluation metric is the *average total variation* between the true $\boldsymbol{T}$ and the estimated $\hat{\boldsymbol{T}}$ (Zhang et al., 2021a) as Eqn. (1).

Baselines We mainly compare our methods with three baselines: T-revision (T-REV) (Xia et al., 2019), Confident Learning (CL) (Northcutt et al., 2021a), and the clusterability-based approach (Zhu et al., 2021c) (HOC). Some recent learning-based methods (Li et al., 2021; Zhang et al., 2021a) jointly optimize $\boldsymbol{T}$ and the model during learning. We find their methods are unstable compared to the above baseline approaches on non-image datasets and defer their results to Appendix B.2.

Our approach To evaluate each component of our approach, we test four variants: OURS-$\boldsymbol{X}$-KL, OURS-$\boldsymbol{X}$-TV, OURS-$\boldsymbol{A}$-KL, and OURS-$\boldsymbol{A}$-TV. Prefix OURS-$\boldsymbol{X}$ indicates that we directly use the original input features and substitute the soft cosine similarity for the original hard one employed by HOC. This setting checks the performance of ignoring correlations among different feature variables. Prefix OURS-$\boldsymbol{A}$ indicates that we firstly transform $\boldsymbol{X}$ to $\boldsymbol{A}$ as Section 3.2 then apply soft cosine similarity. Suffixes -KL and -TV denote using the f-mutual information when $f(v) = v\log(v)$ and $f(v) = \frac{1}{2}|v-1|$, respectively. The matrix $\boldsymbol{W}$ for soft cosine similarity is: $W_{\mu\mu} = \phi(I_f(X_\mu, \widetilde{Y})), W_{\mu\nu} = 0, \forall \nu \neq \mu$, where $\phi(x)$ is an order-preserving activation function. In our evaluations, we set $\phi(x) = [x]_0^1$ for tabular benchmarks and $\phi(x) = [\log(x)]_0^1$ for natural language benchmarks, where $[x]_0^1$ represents normalizing x to range $(0, 1]$.

Datasets and models We examine our approaches on two different application domains other than images: the tabular benchmarks including 7 tabular datasets from the UCI machine learning repository (Dua & Graff, 2017) and 4 natural language benchmarks including AG's news (Zhang et al., 2015), DBpedia (Auer et al., 2007), Yelp-5 (Yelp, 2015), and Jigsaw (Jigsaw, 2018). We use the raw features for tabular benchmarks. Following the same preprocessing procedure as Wang et al. (2022b), we use a pre-trained BERT model (Devlin et al., 2019) to extract 768 dimensional embedding vectors for natural language benchmarks.

Noise type We synthesize the noisy data distribution by injecting class-dependent noise. Particularly, on tabular benchmarks, we test both the symmetric and the asymmetric label noise in Table 1. On natural language benchmarks, we randomly generate the diagonal-dominant label noise following the Dirichlet distribution. Particularly, suppose the average noise rate is e. For each row of $\boldsymbol{T}$, We randomly sample a diagonal element following $T_{ii} = e + \mathsf{Unif}(-0.05, 0.05)$ and set the off-diagonal elements following $\mathsf{Dir}(\mathbf{1})$, where $\mathsf{Unif}(a, b)$ denotes the uniform distribution bounded by (a, b), $\mathsf{Dir}(\mathbf{1})$ denotes the Dirichlet distribution with parameter $\mathbf{1} := [1, \cdots, 1]$ ($K-1$ values).

5.1. Evaluation on Tabular Benchmarks

Table 1 shows the performance comparisons on tabular datasets. By counting the number of **bold** (top-2 performance) results, we know all the four variants of our proposed method perform better than three baselines statistically. Additionally, based on the counting results, OURS-$\boldsymbol{X}$ wins in 16 settings while OURS-$\boldsymbol{A}$ wins in 20 settings, indicating decoupling different parts by eigen decomposition is not statistically significantly useful. This may be due to the property of tabular data: the original correlation in $\boldsymbol{X}$ may not be strong and the decoupling operations will involve extra errors and make the results even worse.

5.2. Evaluation on Natural Language Benchmarks

We also evaluate our methods on more sophisticated text classifications tasks. We use a heuristic method to set the average noise rate e. The aim is to make the ratio between diagonal elements and off-diagonal elements of $\boldsymbol{T}$ consistent. According to our rule in the caption of Table 2, the low, medium, and high noise rates for binary, 5-class, and 10-class classifications are $(0.11, 0.2, 0.4)$, $(0.2, 0.33, 0.57)$, and $(0.27, 0.43, 0.67)$, respectively, which align with the general cognition of the community (Cheng et al., 2021a). Comparing Table 2 with Table 1, we find the direct reweighting by f-mutual information (OURS-$\boldsymbol{X}$) performs similarly to the hard cosine similarity adopted by HOC, while the information-theoretic reweighting after projecting $\boldsymbol{X}$ to its eigen space (OURS-$\boldsymbol{A}$) performs consistently better than other methods. Intuitively, the correlations of BERT embeddings are more stronger than those of tabular data. Thus we observe a clear performance improvement by removing correlations. Besides, it is interesting to see *the estimation error may decrease for higher noise rate setting*.We conjecture the reasons may comprise: 1) The original dataset may be noisy, e.g., the original Yelp dataset contains lots of noisy reviews (Luca, 2016). Consider a binary classification with inherent 20% noise. Then adding 10% (low) noise will make the average noise rate to 0.26, where the gap between the real noise rate and the hypothesized noise rate is $0.26 - 0.1 = 0.16$. Similarly, the gap of adding

Table 1. The estimation error ($\times 100$) on tabular benchmarks. Feature dimension: d. Number of clean instances in class-1 (or class-2): N_1 (or N_2). Noise Rate: LOW: $e_1 = 0.2, e_2 = 0.1$. MEDIUM: $e_1 = 0.4, e_2 = 0.2$. HIGH: $e_1 = 0.4, e_2 = 0.4$. Top-2 of each row are **bold**.

TABULAR DATASETS (d, [N_1, N_2])	NOISE RATE	METHOD T-REV	CL	HOC	OURS-$\boldsymbol{X}$-KL	OURS-$\boldsymbol{X}$-TV	OURS-$\boldsymbol{A}$-KL	OURS-$\boldsymbol{A}$-TV
HEART (23, [138, 165])	LOW	9.25	**8.00**	9.82	**8.09**	8.70	8.88	9.18
	MEDIUM	11.94	11.48	7.85	9.55	**1.48**	**3.98**	5.51
	HIGH	6.74	6.54	4.71	9.78	14.91	**1.33**	**4.21**
BANANA (2, [2924, 2376])	LOW	30.89	30.53	14.96	10.74	11.84	**10.57**	**9.26**
	MEDIUM	20.77	20.81	8.97	**4.90**	**3.98**	6.41	6.97
	HIGH	**6.71**	7.58	**4.78**	12.11	8.89	9.78	11.26
TITANIC (3, [1490, 711])	LOW	21.40	20.62	11.24	**10.83**	**9.96**	11.05	11.60
	MEDIUM	10.83	10.31	9.97	9.82	**9.65**	**9.61**	9.75
	HIGH	6.93	6.75	1.94	1.97	1.97	**1.89**	**1.92**
SPLICE (240, [1648, 1527])	LOW	10.63	9.32	7.32	**2.29**	**1.87**	3.00	3.65
	MEDIUM	10.35	9.84	5.38	2.21	3.90	**1.26**	**2.15**
	HIGH	7.86	7.74	17.43	**4.26**	**2.94**	4.41	5.81
TWONORM (20, [3697, 3703])	LOW	1.79	1.63	2.12	2.34	2.80	**0.54**	**0.65**
	MEDIUM	1.86	1.42	**1.38**	1.42	**1.30**	2.14	1.73
	HIGH	1.67	**1.22**	5.18	5.88	3.47	**1.46**	2.12
WAVEFORM (21, [3353, 1647])	LOW	10.59	10.89	7.93	6.68	6.78	**5.67**	**5.79**
	MEDIUM	8.45	7.82	3.51	**2.50**	3.09	**2.34**	2.72
	HIGH	5.04	4.76	3.84	2.72	**1.71**	**2.29**	5.42
FLARE-SOLAR (31, [477, 589])	LOW	19.28	18.43	15.24	**14.60**	14.84	15.63	**14.71**
	MEDIUM	16.57	16.38	**4.58**	**4.39**	5.05	4.64	4.82
	HIGH	8.35	8.25	4.71	4.47	4.74	**3.87**	**3.30**

Table 2. The estimation error ($\times 100$) on natural language benchmarks. Feature dimension: d. Number of clean instances in class-k: N_k. [30K $\times$ 4]: $N_1 = N_2 = N_3 = N_4 =$30k. Average noise rate follows $e = 1/(1 + r/\sqrt{K-1})$. LOW: $r = 8$. MEDIUM: $r = 4$. HIGH: $r = 1.5$. Top-2 of each row are **bold**.

TEXT DATASETS (d, [$N_1, \cdots, N_K$])	NOISE RATE	METHOD T-REV	CL	HOC	OURS-$\boldsymbol{X}$-KL	OURS-$\boldsymbol{X}$-TV	OURS-$\boldsymbol{A}$-KL	OURS-$\boldsymbol{A}$-TV
AG'S NEWS (BERT) (768, [30K $\times$ 4])	LOW	10.38	11.41	13.32	12.65	12.75	**8.36**	**8.35**
	MEDIUM	10.71	10.63	10.62	10.13	10.45	**6.44**	**6.52**
	HIGH	13.97	13.82	6.80	6.83	6.69	**4.54**	**4.19**
DBPEDIA (BERT) (768, [40K $\times$ 14])	LOW	6.80	5.31	7.57	6.76	6.94	**2.52**	**2.52**
	MEDIUM	14.91	14.40	6.30	5.66	5.78	**2.33**	**2.28**
	HIGH	24.23	23.28	6.00	5.18	5.22	**2.42**	**2.43**
YELP-5 (BERT) (768, [130K $\times$ 5])	LOW	38.49	38.75	40.87	40.71	40.58	**37.37**	**37.19**
	MEDIUM	35.46	36.05	33.63	34.23	33.88	**31.79**	**31.94**
	HIGH	21.20	20.88	19.09	18.56	20.13	**18.11**	**18.06**
JIGSAW (BERT) (768, [144,277, 15,294])	LOW	20.92	20.17	14.25	14.07	14.24	**9.76**	**9.97**
	MEDIUM	17.10	16.44	11.28	11.80	12.23	**7.45**	**7.66**
	HIGH	7.19	6.81	4.84	4.85	3.43	**0.78**	**1.02**

Table 3. The last/best epoch clean test accuracies (%) when training with high-level noise defined in Table 2.

METHOD	AG'S NEWS LAST	AG'S NEWS BEST	DBPEDIA LAST	DBPEDIA BEST
HOC (ZHU ET AL., 2021C)	82.17	83.08	91.06	91.06
OURS-A-TV	**85.01**	**85.17**	**97.71**	**97.77**

40% (high) noise is $0.44 - 0.4 = 0.04$. Therefore, even though $\boldsymbol{T}$ is accurately estimated, the absolute error under our current metric will be higher for the HIGH-noise case. 2) As analyzed in Section 2.3, the error of random guess for low-noise (10%) and high-noise (40%) settings are 0.4 and 0.1, respectively, indicating a small error may cause more severe problems in higher-noise settings. We leave more detailed discussions to Appendix B.3.

Downstream learning error Liu & Wang (2021) showed the additional learning risk is positively related to estimation error. To further consolidate the estimation error reduction of our method, we feed the estimated $\boldsymbol{T}$ into forward loss correction (Patrini et al., 2017) and check the clean test accuracy. Table 3 shows our approach significantly improves both the best and last epoch test accuracy by simply changing the $\boldsymbol{T}$ in loss correction from HOC estimates to ours.

6. Conclusions

This work has studied the problem of estimating noise transition matrix on application domains apart from images. We have proposed an information-theoretic approach to down-weight the less informative parts of features with only noisy labels for tasks with lower-quality features. Future directions include implementing and delivering this approach in real-world label noise settings, e.g., long-tail and open-set (Wei et al., 2022a; Hu et al., 2019).

Acknowledgment This work is partially supported by the National Science Foundation (NSF) under grants IIS-2007951, IIS-2143895 and CCF-2023495.

References

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. Dbpedia: A nucleus for a web of open data. In Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., and Cudré-Mauroux, P. (eds.), *The Semantic Web*, pp. 722–735, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. ISBN 978-3-540-76298-0.

Awasthi, P., Beutel, A., Kleindessner, M., Morgenstern, J., and Wang, X. Evaluating fairness of machine learning models under uncertain and incomplete information. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pp. 206–214, 2021.

Battiti, R. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on neural networks*, 5(4):537–550, 1994.

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

Cheng, H., Zhu, Z., Li, X., Gong, Y., Sun, X., and Liu, Y. Learning with instance-dependent label noise: A sample sieve approach. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=2VXyy9mIyU3.

Cheng, H., Zhu, Z., Sun, X., and Liu, Y. Demystifying how self-supervised features improve training from noisy labels. *arXiv preprint arXiv:2110.09022*, 2021b.

Csiszár, I. Information-type measures of difference of probability distributions and indirect observation. *studia scientiarum Mathematicarum Hungarica*, 2:229–318, 1967.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 4171–4186, June 2019. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Dua, D. and Graff, C. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Estévez, P. A., Tesmer, M., Perez, C. A., and Zurada, J. M. Normalized mutual information feature selection. *IEEE Transactions on neural networks*, 20(2):189–201, 2009.

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.

Hu, M., Han, H., Shan, S., and Chen, X. Weakly supervised image classification through noise regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11517–11525, 2019.

Jigsaw. Jigsaw toxic comment classification challenge, 2018. URL https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge.

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

Lamy, A., Zhong, Z., Menon, A. K., and Verma, N. Noise-tolerant fair classification. *Advances in Neural Information Processing Systems*, 32, 2019.

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, X., Liu, T., Han, B., Niu, G., and Sugiyama, M. Provably end-to-end label-noise learning without anchor points. In *International Conference on Machine Learning*, pp. 6403–6413. PMLR, 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. Understanding instance-level label noise: Disparate impacts and treatments. In *International Conference on Machine Learning*, pp. 6725–6735. PMLR, 2021.

Liu, Y. Identifiability of label noise transition matrix. *arXiv preprint arXiv:2202.02016*, 2022.

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 *International Conference on Machine Learning*, pp. 6226–6236. PMLR, 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. and Wang, J. Can less be more? when increasing-to-balancing label noise rates considered beneficial. *Advances in Neural Information Processing Systems*, 34, 2021.

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.

Luca, M. Reviews, reputation, and revenue: The case of yelp. com. *Com (March 15, 2016). Harvard Business School NOM Unit Working Paper*, (12-016), 2016.

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.

Luo, T., Li, X., Wang, H., and Liu, Y. Research replication prediction using weakly supervised learning. In *In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, 2020.

McCormick, T. H., Li, Z. R., Calvert, C., Crampin, A. C., Kahn, K., and Clark, S. J. Probabilistic cause-of-death assignment using verbal autopsies. *Journal of the American Statistical Association*, 111(515):1036–1049, 2016.

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

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, 2021a.

Northcutt, C. G., Athalye, A., and Mueller, J. Pervasive label errors in test sets destabilize machine learning benchmarks. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021b. URL https://openreview.net/forum?id=XccDXrDNLek.

Nowozin, S., Cseke, B., and Tomioka, R. f-gan: Training generative neural samplers using variational divergence minimization. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 271–279, 2016.

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.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

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

Sidorov, G., Gelbukh, A., Gómez-Adorno, H., and Pinto, D. Soft similarity and soft cosine measure: Similarity of features in vector space model. *Computacion y Sistemas*, pp. 491–504, January 2014. ISSN 1405-5546. doi: 10.13053/CyS-18-3-2043.

Vergara, J. R. and Estévez, P. A. A review of feature selection methods based on mutual information. *Neural computing and applications*, 24(1):175–186, 2014.

Wang, H., Xiao, R., Li, Y., Feng, L., Niu, G., Chen, G., and Zhao, J. PiCO: Contrastive label disambiguation for partial label learning. In *International Conference on Learning Representations*, 2022a. URL https://openreview.net/forum?id=EhYjZy6e1gJ.

Wang, J., Guo, H., Zhu, Z., and Liu, Y. Policy learning using weak supervision. *Advances in Neural Information Processing Systems*, 34, 2021a.

Wang, J., Liu, Y., and Levy, C. Fair classification with group-dependent label noise. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pp. 526–536, 2021b.

Wang, J., Wang, X. E., and Liu, Y. Understanding instance-level impact of fairness constraints. In *International Conference on Machine Learning*. PMLR, 2022b.

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, H., Tao, L., Xie, R., and An, B. Open-set label noise can improve robustness against inherent label noise. *Advances in Neural Information Processing Systems*, 34, 2021a.

Wei, H., Tao, L., Xie, R., Feng, L., and An, B. Open-sampling: Exploring out-of-distribution data for re-balancing long-tailed datasets. In *International Conference on Machine Learning (ICML)*. PMLR, 2022a.

Wei, H., Xie, R., Cheng, H., Feng, L., An, B., and Li, Y. Mitigating neural network overconfidence with logit normalization. *arXiv preprint arXiv:2205.09310*, 2022b.

Wei, H., Xie, R., Feng, L., Han, B., and An, B. Deep learning from multiple noisy annotators as a union. *IEEE Transactions on Neural Networks and Learning Systems*, 2022c.

Wei, J. and Liu, Y. When optimizing f-divergence is robust with label noise. *arXiv preprint arXiv:2011.03687*, 2020.

Wei, J., Liu, H., Liu, T., Niu, G., Sugiyama, M., and Liu, Y. To smooth or not? when label smoothing meets noisy labels, 2021b. URL https://arxiv.org/abs/2106.04149.

Wei, J., Zhu, Z., Cheng, H., Liu, T., Niu, G., and Liu, Y. Learning with noisy labels revisited: A study using real-world human annotations. In *International Conference on Learning Representations*, 2022d. URL https://openreview.net/forum?id=TBWA6PLJZQm.

Wei, J., Zhu, Z., Luo, T., Amid, E., Kumar, A., and Liu, Y. To aggregate or not? Learning with separate noisy labels, 2022e. URL https://arxiv.org/abs/2206.07181.

Wold, S., Esbensen, K., and Geladi, P. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

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.

Yang, S., Yang, E., Han, B., Liu, Y., Xu, M., Niu, G., and Liu, T. Estimating instance-dependent label-noise transition matrix using dnns. *arXiv preprint arXiv:2105.13001*, 2021.

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, 2020.

Yelp. Yelp dataset challenge, 2015. URL http://www.yelp.com/dataset_challenge.

Zhang, M., Lee, J., and Agarwal, S. Learning from noisy labels with no change to the training process. In *International Conference on Machine Learning*, pp. 12468–12478. PMLR, 2021a.

Zhang, X., Zhao, J., and LeCun, Y. Character-level convolutional networks for text classification. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper/2015/file/250cf8b51c773f3f8dc8b4be867a9a02-Paper.pdf.

Zhang, Y., Chen, X., Zhou, D., and Jordan, M. I. Spectral methods meet em: A provably optimal algorithm for crowdsourcing. *Advances in neural information processing systems*, 27:1260–1268, 2014.

Zhang, Z., Li, Y., Wei, H., Ma, K., Xu, T., and Zheng, Y. Alleviating noisy-label effects in image classification via probability transition matrix. *arXiv preprint arXiv:2110.08866*, 2021b.

Zhu, Z., Dong, Z., and Liu, Y. Detecting corrupted labels without training a model to predict, 2021a. URL https://arxiv.org/abs/2110.06283.

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 2021b.

Zhu, Z., Song, Y., and Liu, Y. Clusterability as an alternative to anchor points when learning with noisy labels. In *Proceedings of the 38th International Conference on Machine Learning*, ICML '21, 2021c.

Zhu, Z., Luo, T., and Liu, Y. The rich get richer: Disparate impact of semi-supervised learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=DXPftn5kjQK.

Appendix

We show more theoretical details in Section A and more emperical details in Section B. Particularly,

- Section A.1 numerates some popular f-divergence functions and the corresponding optimal variational-conjugate pair (g^*, f^*).
- Section A.2 shows the order-preserving property of using total variation.
- Section A.3 shows the order-preserving property of using KL divergence.
- Section B.1 explains why we build our method on HOC.
- Section B.2 compares two more baselines.
- Section B.3 discuss an interesting observation shown in Table 2 that high-noise settings may have lower errors.

A. Theorems

A.1. Common f-Divergence Functions

Following (Nowozin et al., 2016; Wei & Liu, 2020), we show some common f-divergence functions in Table 4,

Table 4. List of popular f-divergences together with generator functions $f(v)$, optimal variational functions g^* and optimal conjugate functions f^*.

Name	$f(v)$	g^*	dom_{f^*}	$f^*(u)$
Total Variation	$\frac{1}{2}\lvert v-1\rvert$	$\frac{1}{2}\,\text{sign}\left(\frac{p(v)}{q(v)}-1\right)$	$u\in[-\frac{1}{2},\frac{1}{2}]$	u
KL	$v\log v$	$1+\log\frac{p(v)}{q(v)}$	$\mathbb{R}$	e^{u-1}
Jenson-Shannon	$-(1+v)\log\frac{1+v}{2}+v\log v$	$\log\frac{2p(v)}{p(v)+q(v)}$	$u<\log 2$	$-\log(2-e^u)$
Squared Hellinger	$(\sqrt{v}-1)^2$	$1-\sqrt{\frac{q(v)}{p(v)}}$	$u<1$	$\frac{u}{1-u}$
Pearson $\mathcal{X}^2$	$(1-v)^2$	$2\left(\frac{p(v)}{q(v)}-1\right)$	$\mathbb{R}$	$\frac{1}{4}u^2+u$
Neyman $\mathcal{X}^2$	$\frac{(1-v)^2}{v}$	$1-\left(\frac{q(v)}{p(v)}\right)^2$	$u<1$	$2-2\sqrt{1-u}$
Reverse KL	$-\log v$	$-\frac{q(v)}{p(v)}$	$\mathbb{R}_-$	$-1-\log(-u)$

A.2. Total-Variation

A.2.1. Proof for Lemma 4.3

Proof. Consider TV, we have:

$$\begin{aligned}
\mathsf{VD}_f(\tilde{g}^*) =& \mathbb{E}_{V\sim P}[\tilde{g}^*(V)] - \mathbb{E}_{V\sim Q}[f^*(\tilde{g}^*(V))] \\
=& \frac{1}{2}\left[\mathbb{E}_{V\sim P}\left[\operatorname{sign}\left(\frac{\tilde{p}(V)}{\tilde{q}(V)} - 1\right)\right] - \mathbb{E}_{V\sim Q}\left[\operatorname{sign}\left(\frac{\tilde{p}(V)}{\tilde{q}(V)} - 1\right)\right]\right] \\
=& \frac{1}{2}\int_z \sum_{i\in\{1,2\}} [\mathbb{P}(Z=z, Y=i) - \mathbb{P}(Z=z)\mathbb{P}(Y=i)] \cdot \operatorname{sign}\left[\mathbb{P}(Z=z, \widetilde{Y}=i) - \mathbb{P}(Z=z)\mathbb{P}(\widetilde{Y}=i)\right] dz \\
=& \frac{1}{2}[\mathbb{P}(Z=z, Y=1) - \mathbb{P}(Z=z)\mathbb{P}(Y=1)] \cdot \operatorname{sign}\left[\mathbb{P}(Z=z, \widetilde{Y}=1) - \mathbb{P}(Z=z)\mathbb{P}(\widetilde{Y}=1)\right] \\
&+ \frac{1}{2}[\mathbb{P}(Z=z, Y=2) - \mathbb{P}(Z=z)\mathbb{P}(Y=2)] \cdot \operatorname{sign}\left[\mathbb{P}(Z=z, \widetilde{Y}=2) - \mathbb{P}(Z=z)\mathbb{P}(\widetilde{Y}=2)\right] dz \\
=& \frac{1}{2}[\mathbb{P}(Z=z, Y=1) - \mathbb{P}(Z=z)\mathbb{P}(Y=1)] \\
&\cdot \operatorname{sign}\left[\sum_{j\in\{1,2\}}\left(\mathbb{P}(\widetilde{Y}=1|Z=z, Y=j)\mathbb{P}(Z=z, Y=j) - \mathbb{P}(Z=z)\mathbb{P}(\widetilde{Y}=1|Y=j)\mathbb{P}(Y=j)\right)\right] \\
&+ \frac{1}{2}[\mathbb{P}(Z=z, Y=2) - \mathbb{P}(Z=z)\mathbb{P}(Y=2)] \\
&\cdot \operatorname{sign}\left[\sum_{j\in\{1,2\}}\left(\mathbb{P}(\widetilde{Y}=2|Z=z, Y=j)\mathbb{P}(Z=z, Y=j) - \mathbb{P}(Z=z)\mathbb{P}(\widetilde{Y}=2|Y=j)\mathbb{P}(Y=j)\right)\right] dz.
\end{aligned}$$

Note (assume class-dependent label noise)

$$\begin{aligned}
&\operatorname{sign}\left[\sum_{j\in\{1,2\}}\left(\mathbb{P}(\widetilde{Y}=1|Z=z, Y=j)\mathbb{P}(Z=z, Y=j) - \mathbb{P}(Z=z)\mathbb{P}(\widetilde{Y}=1|Y=j)\mathbb{P}(Y=j)\right)\right] \\
=& \operatorname{sign}\left[(1-e_1)\mathbb{P}(Z=z, Y=1) + e_2\mathbb{P}(Z=z, Y=2) - (1-e_1)\mathbb{P}(Z=z)\mathbb{P}(Y=1) - e_2\mathbb{P}(Z=z)\mathbb{P}(Y=2)\right] \\
=& \operatorname{sign}\left[(1-e_1)\left(\mathbb{P}(Z=z, Y=1) - \mathbb{P}(Z=z)\mathbb{P}(Y=1)\right) + e_2\left(\mathbb{P}(Z=z, Y=2) - \mathbb{P}(Z=z)\mathbb{P}(Y=2)\right)\right] \\
=& \operatorname{sign}\left[(1-e_1)\left(\mathbb{P}(Z=z, Y=1) - \mathbb{P}(Z=z)\mathbb{P}(Y=1)\right) + e_2\left(\mathbb{P}(Z=z) - \mathbb{P}(Z=z, Y=1) - \mathbb{P}(Z=z)(1-\mathbb{P}(Y=1))\right)\right] \\
=& \operatorname{sign}(1-e_1-e_2)\cdot\operatorname{sign}(\mathbb{P}(Z=z, Y=1) - \mathbb{P}(Z=z)\mathbb{P}(Y=1)).
\end{aligned}$$

Thus

$$\begin{aligned}
&\mathsf{VD}_f(\tilde{g}^*) \\
=& \frac{1}{2}\int_z [\mathbb{P}(Z=z, Y=1) - \mathbb{P}(Z=z)\mathbb{P}(Y=1)] \cdot \operatorname{sign}(1-e_1-e_2) \cdot \operatorname{sign}(\mathbb{P}(Z=z, Y=1) - \mathbb{P}(Z=z)\mathbb{P}(Y=1)) \\
&+ [\mathbb{P}(Z=z, Y=2) - \mathbb{P}(Z=z)\mathbb{P}(Y=2)] \cdot \operatorname{sign}(1-e_1-e_2) \cdot \operatorname{sign}(\mathbb{P}(Z=z, Y=2) - \mathbb{P}(Z=z)\mathbb{P}(Y=2))\, dz.
\end{aligned}$$

When $\operatorname{sign}(1-e_1-e_2) = 1$, i.e. $e_1 + e_2 < 1$, we have

$$\begin{aligned}
&\mathsf{VD}_f(\tilde{g}^*) \\
=& \frac{1}{2}\int_z \sum_{i\in\{1,2\}} [\mathbb{P}(Z=z, Y=i) - \mathbb{P}(Z=z)\mathbb{P}(Y=i)] \cdot \operatorname{sign}(\mathbb{P}(Z=z, Y=i) - \mathbb{P}(Z=z)\mathbb{P}(Y=i))\, dz \\
=& \mathsf{VD}_f(g^*).
\end{aligned}$$

□

A.2.2. Proof for Theorem 4.4

Proof. Recall that, to show an f-mutual information metric is ϵ-order-preserving under label noise, we need to study how $\widetilde{\mathsf{VD}}_f(\tilde{g}^*)$ differs from the order of $\mathsf{VD}_f(g^*)$.

For total variation, with Lemma 4.2 and Lemma 4.3, we know

$$\widetilde{\mathsf{VD}}_{\mathrm{TV}}(\tilde{g}^*) = (1-e_1-e_2)\mathsf{VD}_{\mathrm{TV}}(\tilde{g}^*) = (1-e_1-e_2)\mathsf{VD}_{\mathrm{TV}}(g^*).$$

Therefore, when $e_1+e_2<1$, $\widetilde{\mathsf{VD}}_{\mathrm{TV}}(\tilde{g}^*)$ always preserves the order of $\mathsf{VD}_{\mathrm{TV}}(g^*)$, indicating the total-variation-based mutual information is 0-order-preserving under class-dependent label noise. □

A.3. KL Divergence

The definition of MI is

$$\begin{aligned}
&I(Z,\widetilde{Y})\\
&=\sum_{j\in\{1,2\}}\int_z \mathbb{P}(Z=z,\widetilde{Y}=j)\log\left(\frac{\mathbb{P}(Z=z,\widetilde{Y}=j)}{\mathbb{P}(Z=z)\mathbb{P}(\widetilde{Y}=j)}\right)dz\\
&=\sum_{j\in\{1,2\}}\int_z \mathbb{P}(Z=z,\widetilde{Y}=j)\log\Big(\mathbb{P}(Z=z,\widetilde{Y}=j)\Big)\,dz-\sum_{j\in\{1,2\}}\int_z \mathbb{P}(Z=z,\widetilde{Y}=j)\Big(\log\left(\mathbb{P}(Z=z)+\log\Big(\mathbb{P}(\widetilde{Y}=j)\Big)\right)dz\\
&=\underbrace{\sum_{j\in\{1,2\}}\int_z \mathbb{P}(Z=z,\widetilde{Y}=j)\log\Big(\mathbb{P}(Z=z,\widetilde{Y}=j)\Big)\,dz}_{\text{Term-1: }-H(Z,\widetilde{Y})}-\underbrace{\int_z \mathbb{P}(Z=z)\log\mathbb{P}(Z=z)\,dz}_{\text{Term-2: }-H(Z)}-\underbrace{\sum_{j\in\{1,2\}}\mathbb{P}(\widetilde{Y}=j)\log\mathbb{P}(\widetilde{Y}=j)dz}_{\text{Term-3: }-H(\widetilde{Y})},
\end{aligned}$$

where

$$\begin{aligned}
-H(Z,\widetilde{Y}) &= \sum_{j\in\{1,2\}}\int_z \mathbb{P}(Z=z,\widetilde{Y}=j)\log\mathbb{P}(Z=z,\widetilde{Y}=j)dz\\
&=\int_z \mathbb{P}(Z=z,\widetilde{Y}=1)\log\mathbb{P}(Z=z,\widetilde{Y}=1)+\mathbb{P}(Z=z,\widetilde{Y}=2)\log\mathbb{P}(Z=z,\widetilde{Y}=2)\,dz.
\end{aligned}$$

We first decouple term-1. Note

$$\begin{aligned}
&\int_z \mathbb{P}(Z=z,\widetilde{Y}=1)\log\mathbb{P}(Z=z,\widetilde{Y}=1)dz\\
&=\int_z\Big[\mathbb{P}(\widetilde{Y}=1|Z=z,Y=1)\mathbb{P}(Z=z,Y=1)+\mathbb{P}(\widetilde{Y}=1|Z=z,Y=2)\mathbb{P}(Z=z,Y=2)\Big]\\
&\quad\cdot\log\Big[\mathbb{P}(\widetilde{Y}=1|Z=z,Y=1)\mathbb{P}(Z=z,Y=1)+\mathbb{P}(\widetilde{Y}=1|Z=z,Y=2)\mathbb{P}(Z=z,Y=2)\Big]\,dz\\
&=\int_z[(1-e_1)\mathbb{P}(Z=z,Y=1)+e_2\mathbb{P}(Z=z,Y=2)]\\
&\quad\cdot\log[(1-e_1)\mathbb{P}(Z=z,Y=1)+e_2\mathbb{P}(Z=z,Y=2)]\,dz\\
&=\int_z[(1-e_1-e_2)\mathbb{P}(Z=z,Y=1)+e_2\mathbb{P}(Z=z)]\\
&\quad\cdot\log[(1-e_1-e_2)\mathbb{P}(Z=z,Y=1)+e_2\mathbb{P}(Z=z)]\,dz\\
&=\int_z[(1-e_1-e_2)\mathbb{P}(Z=z,Y=1)+e_2\mathbb{P}(Z=z)]\log\mathbb{P}(Z=z,Y=1)\\
&\quad+[(1-e_1-e_2)\mathbb{P}(Z=z,Y=1)+e_2\mathbb{P}(Z=z)]\log\left[\frac{(1-e_1-e_2)\mathbb{P}(Z=z,Y=1)+e_2\mathbb{P}(Z=z)}{\mathbb{P}(Z=z,Y=1)}\right]dz\\
&=\int_z[(1-e_1-e_2)\mathbb{P}(Z=z,Y=1)+e_2\mathbb{P}(Z=z)]\log\mathbb{P}(Z=z,Y=1)\\
&\quad+[(1-e_1-e_2)\mathbb{P}(Z=z,Y=1)+e_2\mathbb{P}(Z=z)]\log\left[1-e_1+e_2\frac{\mathbb{P}(Z=z,Y=2)}{\mathbb{P}(Z=z,Y=1)}\right]dz.
\end{aligned}$$

Let $\alpha = \mathbb{P}(Z=z,Y=1)/\mathbb{P}(Z=z,Y=2) \in [0,+\infty)$ (note α is actually a function of (Z,Y)). Then

$$\begin{aligned}
&\int_z \mathbb{P}(Z=z,\widetilde{Y}=1)\log\mathbb{P}(Z=z,\widetilde{Y}=1)dz \\
=&\int_z (1-e_1-e_2)\mathbb{P}(Z=z,Y=1)\log\mathbb{P}(Z=z,Y=1)\,dz + \int_z e_2\mathbb{P}(Z=z)\left[\log\alpha+\log\mathbb{P}(Z=z,Y=2)\right] \\
&+\left[(1-e_1-e_2)\alpha\mathbb{P}(Z=z,Y=2)+e_2\mathbb{P}(Z=z)\right]\log\left(1-e_1+\frac{e_2}{\alpha}\right)dz
\end{aligned}$$

and

$$\begin{aligned}
&\int_z \mathbb{P}(Z=z,\widetilde{Y}=2)\log\mathbb{P}(Z=z,\widetilde{Y}=2)dz \\
=&\int_z (1-e_1-e_2)\mathbb{P}(Z=z,Y=2)\log\mathbb{P}(Z=z,Y=2)\,dz + \int_z e_1\mathbb{P}(Z=z)\log\mathbb{P}(Z=z,Y=2) \\
&+\left[(1-e_1-e_2)\mathbb{P}(Z=z,Y=2)+e_1\mathbb{P}(Z=z)\right]\log\left(1-e_2+e_1\alpha\right)dz
\end{aligned}$$

Thus

$$\begin{aligned}
&\sum_{j\in\{1,2\}}\int_z \mathbb{P}(Z=z,\widetilde{Y}=j)\log\mathbb{P}(Z=z,\widetilde{Y}=j)dz \\
=&(1-e_1-e_2)\sum_{i\in\{1,2\}}\int_z \mathbb{P}(Z=z,Y=i)\log\mathbb{P}(Z=z,Y=i)\,dz \\
&+\int_z e_2\mathbb{P}(Z=z)\log\alpha+(e_1+e_2)\mathbb{P}(Z=z)\log\mathbb{P}(Z=z,Y=2) \\
&+(1-e_1-e_2)\mathbb{P}(Z=z,Y=2)\left[\alpha\log\left(1-e_1+\frac{e_2}{\alpha}\right)+\log(1-e_2+e_1\alpha)\right] \\
&+\mathbb{P}(Z=z)\left[e_1\log(1-e_2+e_1\alpha)+e_2\log\left(1-e_1+\frac{e_2}{\alpha}\right)\right]dz \\
=&(1-e_1-e_2)\sum_{i\in\{1,2\}}\int_z \mathbb{P}(Z=z,Y=i)\log\mathbb{P}(Z=z,Y=i)\,dz \\
&+\int_z e_2\mathbb{P}(Z=z)\log\alpha-(e_1+e_2)\mathbb{P}(Z=z)\log(\alpha+1)+(e_1+e_2)\mathbb{P}(Z=z)\log\mathbb{P}(Z=z) \\
&+\frac{(1-e_1-e_2)}{\alpha+1}\mathbb{P}(Z=z)\left[\alpha\log\left(1-e_1+\frac{e_2}{\alpha}\right)+\log(1-e_2+e_1\alpha)\right] \\
&+\mathbb{P}(Z=z)\left[e_1\log(1-e_2+e_1\alpha)+e_2\log\left(1-e_1+\frac{e_2}{\alpha}\right)\right]dz \\
&=(1-e_1-e_2)\sum_{i\in\{1,2\}}\int_z \mathbb{P}(Z=z,Y=i)\log\mathbb{P}(Z=z,Y=i)\,dz \qquad \textbf{(Term 1.1)} \\
&+\int_z (e_1+e_2)\mathbb{P}(Z=z)\log\mathbb{P}(Z=z)+\mathbb{P}(Z=z)\Delta_{\mathsf{Bias}}(\alpha,e_1,e_2)\,dz, \qquad \textbf{(Term 1.2)}
\end{aligned}$$

where in Term 1.2:

$$\begin{aligned}
\Delta_{\mathsf{Bias}}(\alpha,e_1,e_2) =& e_2\log\alpha-(e_1+e_2)\log(\alpha+1)+\frac{(1-e_1-e_2)}{\alpha+1}\left[\alpha\log\left(1-e_1+\frac{e_2}{\alpha}\right)+\log(1-e_2+e_1\alpha)\right] \\
&+\left[e_1\log(1-e_2+e_1\alpha)+e_2\log\left(1-e_1+\frac{e_2}{\alpha}\right)\right].
\end{aligned} \tag{5}$$

In Term 1.1, recalling $\alpha = \mathbb{P}(Z=z, Y=1)/\mathbb{P}(Z=z, Y=2)$, we have

$$
\begin{aligned}
&(1-e_1-e_2)\sum_{i\in\{1,2\}}\int_z \mathbb{P}(Z=z,Y=i)\log\mathbb{P}(Z=z,Y=i)\,dz\\
=&(1-e_1-e_2)\int_z \frac{\mathbb{P}(Z=z)}{\alpha+1}\log\frac{\mathbb{P}(Z=z)}{\alpha+1}+\frac{\mathbb{P}(Z=z)\alpha}{\alpha+1}\log\frac{\mathbb{P}(Z=z)\alpha}{\alpha+1}\,dz\\
=&(1-e_1-e_2)\int_z \frac{\mathbb{P}(Z=z)}{\alpha+1}\log\mathbb{P}(Z=z)+\frac{\mathbb{P}(Z=z)}{\alpha+1}\log\frac{1}{\alpha+1}+\frac{\mathbb{P}(Z=z)\alpha}{\alpha+1}\log\mathbb{P}(Z=z)+\frac{\mathbb{P}(Z=z)\alpha}{\alpha+1}\log\frac{\alpha}{\alpha+1}\,dz\\
=&(1-e_1-e_2)\int_z \mathbb{P}(Z=z)\log\mathbb{P}(Z=z)\,dz+(1-e_1-e_2)\int_z \mathbb{P}(Z=z)\left[\frac{\alpha}{\alpha+1}\log\frac{\alpha}{\alpha+1}+\frac{1}{\alpha+1}\log\frac{1}{\alpha+1}\right]dz.
\end{aligned}
$$

Denote the effective part of MI by

$$
\Delta_{\mathsf{MI}}(\alpha,e_1,e_2)=(1-e_1-e_2)\left[\frac{\alpha}{\alpha+1}\log\frac{\alpha}{\alpha+1}+\frac{1}{\alpha+1}\log\frac{1}{\alpha+1}\right]. \tag{6}
$$

We have

$$
\begin{aligned}
-H(Z,\widetilde{Y}) &= \sum_{j\in\{1,2\}}\int_z \mathbb{P}(Z=z,\widetilde{Y}=j)\log\mathbb{P}(Z=z,\widetilde{Y}=j)dz\\
&= \int_z \mathbb{P}(Z=z)\log\mathbb{P}(Z=z)\,dz+\int_z \mathbb{P}(Z=z)\left[\Delta_{\mathsf{MI}}(\alpha,e_1,e_2)+\Delta_{\mathsf{Bias}}(\alpha,e_1,e_2)\right]\,dz,
\end{aligned}
$$

and

$$
I(Z,\widetilde{Y})=\int_z \mathbb{P}(Z=z)\left[\Delta_{\mathsf{MI}}(\alpha,e_1,e_2)+\Delta_{\mathsf{Bias}}(\alpha,e_1,e_2)\right]\,dz+H(\widetilde{Y}).
$$

Define $\Delta_{\mathsf{Bias,MI}}(\alpha,e_1,e_2)=\Delta_{\mathsf{MI}}(\alpha,e_1,e_2)+\Delta_{\mathsf{Bias}}(\alpha,e_1,e_2)$. Then

$$
\begin{aligned}
&\Delta_{\mathsf{Bias,MI}}(\alpha,e_1,e_2)\\
=&\Delta_{\mathsf{MI}}(\alpha,e_1,e_2)+\Delta_{\mathsf{Bias}}(\alpha,e_1,e_2)\\
=&(1-e_1-e_2)\left[\frac{\alpha}{\alpha+1}\log\frac{\alpha}{\alpha+1}+\frac{1}{\alpha+1}\log\frac{1}{\alpha+1}\right]+e_2\log\alpha-(e_1+e_2)\log(\alpha+1)\\
&+\frac{(1-e_1-e_2)}{\alpha+1}\left[\alpha\log\left(1-e_1+\frac{e_2}{\alpha}\right)+\log(1-e_2+e_1\alpha)\right]\\
&+\left[e_1\log(1-e_2+e_1\alpha)+e_2\log\left(1-e_1+\frac{e_2}{\alpha}\right)\right]\\
=&(1-e_1-e_2)\left[\frac{\alpha}{\alpha+1}\log\alpha+\log\frac{1}{\alpha+1}\right]+e_2\log\alpha+(e_1+e_2)\log\frac{1}{\alpha+1}+\\
&\left[\frac{(1-e_1-e_2)\alpha}{\alpha+1}+e_2\right]\log\left(1-e_1+\frac{e_2}{\alpha}\right)+\left[\frac{(1-e_1-e_2)}{\alpha+1}+e_1\right]\log(1-e_2+e_1\alpha)\\
=&\left[\frac{(1-e_1-e_2)\alpha}{\alpha+1}+e_2\right]\log\alpha+\log\frac{1}{\alpha+1}+\\
&\left[\frac{(1-e_1-e_2)\alpha}{\alpha+1}+e_2\right]\log\left(1-e_1+\frac{e_2}{\alpha}\right)+\left[\frac{(1-e_1-e_2)}{\alpha+1}+e_1\right]\log(1-e_2+e_1\alpha)\\
=&\log\frac{1}{\alpha+1}+\frac{1}{\alpha+1}\left[(1-e_1-e_2)\alpha+e_2(\alpha+1)\right]\log\left(\alpha(1-e_1)+e_2\right)\\
&+\frac{1}{\alpha+1}\left[(1-e_1-e_2)+e_1(\alpha+1)\right]\log(1-e_2+e_1\alpha)\\
=&\left[\frac{\alpha(1-e_1)+e_2}{\alpha+1}+\frac{(\alpha+1)-(\alpha(1-e_1)+e_2)}{\alpha+1}\right]\log\frac{1}{\alpha+1}+\frac{1}{\alpha+1}\left[\alpha(1-e_1)+e_2\right]\log\left(\alpha(1-e_1)+e_2\right)\\
&+\frac{1}{\alpha+1}\left[(\alpha+1)-(\alpha(1-e_1)+e_2)\right]\log((\alpha+1)-(\alpha(1-e_1)+e_2))\\
=&\frac{\alpha(1-e_1)+e_2}{\alpha+1}\log\frac{\alpha(1-e_1)+e_2}{\alpha+1}+\left[1-\frac{\alpha(1-e_1)+e_2}{\alpha+1}\right]\log\left[1-\frac{\alpha(1-e_1)+e_2}{\alpha+1}\right].
\end{aligned}
$$

Note

$$\frac{\alpha(1-e_1)+e_2}{\alpha+1} = (1-e_1-e_2)\cdot\frac{\alpha}{\alpha+1}+e_2.$$

Let $\beta = \alpha/(1+\alpha) \in [0,1)$. Note β is a function of z. We drop notation z for ease of notation. Then

$$\begin{aligned}\Delta_{\mathsf{Bias,MI}}(\beta,e_1,e_2) =& ((1-e_1-e_2)\beta+e_2)\log((1-e_1-e_2)\beta+e_2)\\ &+[1-((1-e_1-e_2)\beta+e_2)]\log[1-((1-e_1-e_2)\beta+e_2)].\end{aligned}$$

The bias caused by label noise is

$$\begin{aligned}\Delta_{\mathsf{Bias}}(\beta,e_1,e_2) =& \Delta_{\mathsf{Bias,MI}}(\beta,e_1,e_2)-\Delta_{\mathsf{MI}}(\beta,e_1,e_2)\\ =& ((1-e_1-e_2)\beta+e_2)\log((1-e_1-e_2)\beta+e_2)\\ &+[1-((1-e_1-e_2)\beta+e_2)]\log[1-((1-e_1-e_2)\beta+e_2)]\\ &-(1-e_1-e_2)[\beta\log\beta+(1-\beta)\log(1-\beta)].\end{aligned} \tag{7}$$

To get $\arg\max_{\beta\in[0,1)}$, we check the first derivative:

$$\frac{\partial\Delta_{\mathsf{Bias}}(\beta,e_1,e_2)}{\partial\beta} = (1-e_1-e_2)\left[\log\frac{(1-e_1-e_2)\beta+e_2}{1-(1-e_1-e_2)\beta-e_2}-\log\frac{\beta}{1-\beta}\right].$$

Let $\partial\Delta_{\mathsf{Bias}}(\beta,e_1,e_2)/\partial\beta = 0$, we have

$$\beta^* = \frac{e_2}{e_1+e_2}.$$

By checking $\partial^2\Delta_{\mathsf{Bias}}(\beta,e_1,e_2)/\partial\beta^2$, we can find that $\Delta_{\mathsf{Bias}}(\beta,e_1,e_2)$ is increasing when $\beta\in[0,\beta^*]$, and decreasing when $\beta\in[\beta^*,1]$. Thus $\beta^* = e_2/(e_1+e_2)$ is the global maximum and the upper bound for $\Delta_{\mathsf{Bias}}(\beta,e_1,e_2)$ is

$$\Delta_{\mathsf{Bias}}(\beta,e_1,e_2) \le e_1\log e_1+e_2\log e_2-(e_1+e_2)\log(e_1+e_2).$$

Assume $\beta\in[\underline{\beta},\bar{\beta}]$ in practice. The lower bound for $\Delta_{\mathsf{Bias}}(\beta,e_1,e_2)$ is

$$\Delta_{\mathsf{Bias}}(\beta,e_1,e_2) \ge \min(\Delta_{\mathsf{Bias}}(\underline{\beta},e_1,e_2),\Delta_{\mathsf{Bias}}(\bar{\beta},e_1,e_2)).$$

A looser bound that holds for all the possible $\beta\in[0,1)$ is:

$$\Delta_{\mathsf{Bias}}(\beta,e_1,e_2) \ge \min(e_1\log e_1+(1-e_1)\log(1-e_1), e_2\log e_2+(1-e_2)\log(1-e_2)).$$

Note (when $e_1=e_2=0$)

$$I(Z,Y) = \int_z \mathbb{P}(Z=z)\left[\Delta_{\mathsf{MI}}(\alpha,0,0)\right]\,dz + H(Y),$$

and $\Delta_{\mathsf{MI}}(\beta,e_1,e_2) = (1-e_1-e_2)\Delta_{\mathsf{MI}}(\alpha,0,0)$.

Hence (note β is actually a function of Z),

$$\begin{aligned}I(Z,\widetilde{Y}) =& \int_z \mathbb{P}(Z=z)\left[\Delta_{\mathsf{MI}}(\alpha,e_1,e_2)+\Delta_{\mathsf{Bias}}(\alpha,e_1,e_2)\right]\,dz + H(\widetilde{Y})\\ =& \int_z \mathbb{P}(Z=z)\left[(1-e_1-e_2)\Delta_{\mathsf{MI}}(\beta,0,0)+\Delta_{\mathsf{Bias}}(\beta,e_1,e_2)\right]\,dz + H(\widetilde{Y})\\ =& (1-e_1-e_2)\int_z \mathbb{P}(Z=z)\Delta_{\mathsf{MI}}(\beta,0,0)\,dz + \int_z \Delta_{\mathsf{Bias}}(\beta,e_1,e_2)\,dz + H(\widetilde{Y})\\ =& (1-e_1-e_2)I(Z,Y) + \int_z \Delta_{\mathsf{Bias}}(\beta,e_1,e_2)\,dz - (1-e_1-e_2)H(Y) + H(\widetilde{Y})\\ =& (1-e_1-e_2)I(Z,Y) + C(e_1,e_2,Y,\widetilde{Y}) + \Delta_Z(e_1,e_2),\end{aligned}$$

where

$$C(e_1, e_2, Y, \widetilde{Y}) = \min(e_1 \log e_1 + (1-e_1)\log(1-e_1), e_2 \log e_2 + (1-e_2)\log(1-e_2)) - (1-e_1-e_2)H(Y) + H(\widetilde{Y})$$

is a constant for given Y and $\widetilde{Y}$. The other part is in the range $\Delta_Z(e_1, e_2) \in [0, \mathsf{Gap}_Z(e_1, e_2)]$, and

$$\begin{aligned}\mathsf{Gap}_Z(e_1, e_2) =& e_1 \log e_1 + e_2 \log e_2 - (e_1 + e_2)\log(e_1 + e_2) \\ & - \min(e_1 \log e_1 + (1-e_1)\log(1-e_1), e_2 \log e_2 + (1-e_2)\log(1-e_2)).\end{aligned}$$

Note $\Delta_Z(e_1, e_2)$ may be different for Z_μ and Z_ν, $\mu \neq \nu$.

Therefore, when

$$|I_f(Z_\mu; \widetilde{Y}) - I_f(Z_\nu; \widetilde{Y})| > \mathsf{Gap}_Z(e_1, e_2),$$

we have

$$\operatorname{sign}\left[I_f(Z_\mu; \widetilde{Y}) - I_f(Z_\nu; \widetilde{Y})\right] = \operatorname{sign}\left[I_f(Z_\mu; Y) - I_f(Z_\nu; Y)\right], \forall \mu \in [d] \nu \in [d].$$

Now we take a further look at the gap $\mathsf{Gap}_Z(e_1, e_2)$. Assume $e_1 \geq e_2 \Rightarrow e_2 = \delta e_1$, where $\delta \in [0, 1]$. Then $H(e_1) \leq H(e_2)$, and

$$\begin{aligned}\mathsf{Gap}_Z(e_1, e_2) =& e_1 \log e_1 + e_2 \log e_2 - (e_1 + e_2)\log(e_1 + e_2) - \min(H(e_1), H(e_2)) \\ =& e_2 \log e_2 - (e_1 + e_2)\log(e_1 + e_2) - (1-e_1)\log(1-e_1) \\ =& e_2 \log \frac{e_2}{e_1 + e_2} + e_1 \log \frac{1-e_1}{e_1 + e_2} - \log(1-e_1) \\ =& \delta e_1 \log \frac{\delta}{1+\delta} + e_1 \log \frac{1-e_1}{e_1(1+\delta)} - \log(1-e_1) \\ =& e_1 \left[\delta \log \frac{\delta}{1+\delta} + \log \frac{1}{(1+\delta)}\right] + e_1 \log \frac{(1-e_1)}{e_1} - \log(1-e_1) \\ =& e_1 \left[\delta \log \delta - (1+\delta)\log(1+\delta)\right] - (1-e_1)\log(1-e_1) - e_1 \log e_1 \\ =& e_1 \left[\delta \log \delta - (1+\delta)\log(1+\delta)\right] + H(e_1)\end{aligned}$$

Note: We can also roughly estimate $\delta \log \delta - (1+\delta)\log(1+\delta)$ by the best quadratic fit (rooted mean squared error ≈ 0.02) and get

$$\delta \log \delta - (1+\delta)\log(1+\delta) \approx 0.9124\delta^2 - 2.14\delta - 0.1202.$$

B. More Discussions

B.1. Rationale for building on HOC

The rationale for building our analyses on HOC is described as follows. Our major concern is that the learning-based approaches usually require more effort in tuning their hyperparameters. The effect of reweighing feature variables will be entangled with the training procedure, making things more complicated. On the other hand, the training-free approach seems to be more lightweight to employ the reweighing treatment. In particular, HOC consistently achieves lower estimation error, as shown in Figure 1.

B.2. More Experimental Results

We only use one linear layer as the model for the learning-based methods since it can achieve satisfying performance when training with clean data.

We compare our methods with two recent works (Li et al., 2021; Zhang et al., 2021a) in Table 5. Our method achieves an overall better performance than theirs. We search the learning rate from $[0.1, 0.01, 0.001, 0.0001, 0.00001]$ and report the best result for their methods. During experiments, we find their methods tend to be sensitive to hyperparameter settings. On AG's news and Jigsaw, we find T-Vol and T-TV estimate $\boldsymbol{T} \approx \boldsymbol{I}$ for all three noise settings. Thus the low estimation error for the low-noise setting is more like a coincidence due to the trivial solution $\boldsymbol{I}$.

Table 5. The estimation error ($\times 100$) on natural language benchmarks. Feature dimension: d. Number of clean instances in class-k: N_k. [30K $\times$ 4]: $N_1 = N_2 = N_3 = N_4$ =30k. Average noise rate follows $e = 1/(1 + r/\sqrt{K-1})$. LOW: $r = 8$. MEDIUM: $r = 4$. HIGH: $r = 1.5$. Top-2 of each row are **bold**.

TEXT DATASETS $(d, [N_1, \cdots, N_K])$	NOISE RATE	METHOD T-REV	CL	HOC	T-VOL	T-TV	OURS-$\boldsymbol{A}$-KL	OURS-$\boldsymbol{A}$-TV
AG'S NEWS (BERT) (768, [30K $\times$ 4])	LOW	10.38	11.41	13.32	9.44	**7.08**	8.36	**8.35**
	MEDIUM	10.71	10.63	10.62	10.53	11.02	**6.44**	**6.52**
	HIGH	13.97	13.82	6.80	30.33	31.66	**4.54**	**4.19**
DBPEDIA (BERT) (768, [40K $\times$ 14])	LOW	6.80	5.31	7.57	34.23	34.00	**2.52**	**2.52**
	MEDIUM	14.91	14.40	6.30	26.94	27.62	**2.33**	**2.28**
	HIGH	24.23	23.28	6.00	30.72	31.35	**2.42**	**2.43**
YELP-5 (BERT) (768, [130K $\times$ 5])	LOW	38.49	38.75	40.87	**13.16**	**12.46**	37.37	37.19
	MEDIUM	35.46	36.05	33.63	**12.30**	**12.48**	31.79	31.94
	HIGH	21.20	20.88	19.09	30.09	33.20	**18.11**	**18.06**
JIGSAW (BERT) (768, [144,277, 15,294])	LOW	20.92	20.17	14.25	**3.25**	**3.26**	9.76	9.97
	MEDIUM	17.10	16.44	11.28	12.23	12.21	**7.45**	**7.66**
	HIGH	7.19	6.81	4.84	32.39	32.39	**0.78**	**1.02**

B.3. More Experiments

High-noise settings may have lower errors Table 2 shows the estimation error may decrease for higher noise rate settings. This observations mainly due to two reasons:

- Reason-1: The original dataset may be noisy. Notably, the original Yelp dataset contains lots of noisy reviews (Luca, 2016). Now we analyze the issues caused by an originally noise dataset with a toy example.
  Example: Consider a binary classification with inherent 20% noise. Define two noise settings: *1) Low noise*: Add 10% symmetric label noise ($e_1 = e_2 = 0.1$). *2) High noise*: Add 40% symmetric label noise ($e_1 = e_2 = 0.4$). For the low noise setting, the real average noise rate is:

$$e_{\text{low, real}} = 0.1 \times 0.8 + 0.2 \times 0.9 = 0.26.$$

  For the high noise setting, the real average noise rate is:

$$e_{\text{high, real}} = 0.4 \times 0.8 + 0.2 \times 0.6 = 0.44.$$

  Note $e_{\text{low, synthetic}} = 0.1$ and $e_{\text{high, synthetic}} = 0.4$. Thus the gap between the real noise rate and the synthetic noise rate is

$$e_{\text{low, real}} - e_{\text{low, synthetic}} = 0.26 - 0.1 = 0.16, \qquad e_{\text{high, real}} - e_{\text{high, synthetic}} = 0.44 - 0.4 = 0.04.$$

  Therefore, with inherent label noise exists, the perfectly estimated $\boldsymbol{T}$ will have an error of 0.16 for the low-noise setting and an error of 0.04 for the high-noise setting, which accounts for our current observation.
- Reason-2: The tolerance of noise rates for different settings are different. Consider a binary classification with symmetric label noise, i.e., $e_1 = e_2 = e$. Let the random guess be $e_1 = e_2 = 0.5$. Define the estimation error caused by the random guess as the tolerance. Thus the tolerances when $e = 0.1$ and $e = 0.4$ are 0.4 and 0.1, respectively. From this aspect, an error of 0.1 will not destroy the low-noise case since

$$|\hat{e}_{1,\text{low}} - e_{1,\text{low}}| = 0.1 \Rightarrow \hat{e}_{1,\text{low}} = 0.2.$$

  But an error of 0.1 may destroy the high-noise case since

$$|\hat{e}_{1,\text{high}} - e_{1,\text{high}}| = 0.1 \Rightarrow \hat{e}_{1,\text{high}} = 0.3 \text{ or } 0.5.$$

  Therefore, although the error of high-noise settings seems low, it may cause severe problems.

A preliminary test on calibrating inherent errors We do the following experiment to help explain Reason-1 by calibrating the inherent label noise in Yelp-5. Note the label noise accumulation follows:

$$\boldsymbol{T}_{\text{real}} = \boldsymbol{T}_{\text{org}} \boldsymbol{T}_{\text{synthetic}},$$

where $\boldsymbol{T}_{\text{synthetic}} = \boldsymbol{T}$. If we know $\boldsymbol{T}_{\text{org}}$, we can calibrate $\boldsymbol{T}_{\text{synthetic}}$ and evaluate the error by $\mathsf{Error}(\boldsymbol{T}_{\text{org}}\boldsymbol{T}, \boldsymbol{T}_{\text{org}}\hat{\boldsymbol{T}})$. Unfortunately, we cannot find the ground-truth $\boldsymbol{T}_{\text{org}}$ for Yelp-5. For a preliminary test, we estimate $\boldsymbol{T}_{\text{org}}$ by applying OURS-$\boldsymbol{A}$-KL on the original Yelp dataset. We show the calibrated error in Table 6, where we can find the high-noise settings are indeed more challenging (higher error) compared with the low-noise setting.

Table 6. The calibrated estimation error ($\times 100$) on Yelp-5. Average noise rate follows $e = 1/(1 + r/\sqrt{K-1})$. LOW: $r = 8$. MEDIUM: $r = 4$. HIGH: $r = 1.5$.

TEXT DATASETS $(d, [N_1, \cdots, N_K])$	NOISE RATE	METHOD OURS-$\boldsymbol{A}$-KL	OURS-$\boldsymbol{A}$-TV
YELP-5 (BERT) (768, [130K $\times$ 5])	LOW	3.56	4.01
	MEDIUM	3.46	2.59
	HIGH	8.59	8.56