Which is Better for Learning with Noisy Labels: The Semi-supervised Method or Modeling Label Noise?

Yu Yao [1 2] **Mingming Gong** [3 1] **Yuxuan Du** [4] **Jun Yu** [5] **Bo Han** [6] **Kun Zhang** [1 2] **Tongliang Liu** [1 7]

Abstract

In real life, accurately annotating large-scale datasets is sometimes difficult. Datasets used for training deep learning models are likely to contain label noise. To make use of the dataset containing label noise, two typical methods have been proposed. One is to employ the semi-supervised method by exploiting labeled *confident examples* and unlabeled *unconfident examples*. The other one is to *model label noise* and design *statistically consistent* classifiers. A natural question remains unsolved: which one should be used for a specific real-world application? In this paper, we answer the question from the perspective of *causal data generative process*. Specifically, the performance of the semi-supervised based method depends heavily on the data generative process while the method modeling label-noise is not influenced by the generation process. For example, for a given dataset, if it has a causal generative structure that the features cause the label, the semi-supervised based method would not be helpful. When the causal structure is unknown, we provide an intuitive method to discover the causal structure for a given dataset containing label noise.

[1]Department of Machine Learning, Mohamed bin Zayed University of Artificial Intelligence, United Arab Emirates [2]Department of Philosophy, Carnegie Mellon University, United States [3]School of Mathematics and Statistics, The University of Melbourne, Australia [4]JD Explore Academy, People's Republic of China [5]Department of Automation, University of Science and Technology of China, People's Republic of China [6]Department of Computer Science, Faculty of Science, Hong Kong Baptist University, People's Republic of China [7]School of Computer Science, Faculty of Engineering, The University of Sydney, Australia. Correspondence to: Tongliang Liu <tliang.liu@gmail.com>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

1. Introduction

Deep neural networks can achieve remarkable performance when accurately annotated large-scale training datasets are available. However, annotating a large number of examples accurately is often expensive and sometimes infeasible in real life. Cheap datasets which contain label errors are easy to obtain (Li et al., 2019) and have been widely used to train deep neural networks. Recent results (Han et al., 2018; Nguyen et al., 2019) show that deep neural networks can easily memorize label noise during training, which leads to poor test performance.

To reduce the side effect of label noise, there are two major streams of methods. One stream of methods is based on semi-supervised techniques, i.e., *SSL-based methods*. These methods focus on getting rid of label errors. Specifically, they would first construct a labeled set and an unlabeled set from the noisy training data. The labeled set is obtained by selecting *confident examples* whose labels are likely to be correct, e.g., by exploiting the memorization effect of deep networks (Jiang et al., 2018). The unlabeled set is obtained by discarding the labels of *unconfident examples* (i.e., whose labels are likely to be incorrect). Then, they employ semi-supervised (SSL) techniques on the constructed labeled set and unlabeled set to achieve state-of-the-art performance (Li et al., 2019; 2020; Wei et al., 2020; Yao et al., 2021; Tan et al., 2021; Ciortan et al., 2021; Yao et al., 2021). These methods are usually based on heuristics and do not provide a theoretical guarantee.

Another major stream of methods is to model the label noise to get rid of its side effects i.e., *model-based methods*. They mainly focus on estimating the label noise *transition matrix* $\boldsymbol{T}(\boldsymbol{x})$, i.e., $\boldsymbol{T}_{ij}(\boldsymbol{x}) = P(\tilde{Y} = i|Y = j, X = \boldsymbol{x})$ representing the probability that an instance $\boldsymbol{x}$ with a clean label $Y = i$ but flips to a noisy label $\tilde{Y} = j$. The idea is that the clean class posterior distribution $P(Y|X)$ can be inferred by learning the transition matrix $\boldsymbol{T}(\boldsymbol{x})$ and noisy class posterior distribution $P(\tilde{Y}|X)$. In general, when $\boldsymbol{T}(\boldsymbol{x})$ is well estimated (or given), these methods are *statistically consistent*, i.e., they guarantee that the classifiers learned from the noisy data converge to the optimal classifiers defined on the clean data as the size of the noisy training data increases (Patrini et al., 2017; Xia et al., 2019).

Model-based methods provide statistical guarantees, and SSL-based methods have demonstrated state-of-the-art (SOTA) performance on many benchmark datasets. It naturally raises the question that which stream of methods should be exploited when given a real-world dataset. In this paper, from a causal perspective, we answer that it is closely dependent on the generative process of the dataset, and none of the two streams of methods are dominating. The SSL-based methods can easily incorporate heuristics (e.g., prior knowledge) to make use of the finite training sample but they do not work if the feature is the cause of the label in the data generative process. The model-based methods are not influenced by the data generative process. They can make use of all the instances and noisy labels and can be statistically consistent but they need a large training sample to perform well.

Specifically, when the instance X is a cause of the clean label Y, the distributions $P(X)$ and $P(Y|X)$ are disentangled (Schölkopf et al., 2012; Zhang et al., 2015), which means that $P(X)$ contains no labeling information. In other words, exploiting the unlabeled data by SSL-based methods cannot help learn the classifier. When the clean label Y is a cause of the instance X, the distributions of $P(X)$ and $P(Y|X)$ are entangled (Schölkopf et al., 2012; Zhang et al., 2015), then $P(X)$ generally contains some information of $P(Y|X)$. Then SSL-based methods are helpful. In many real-world applications, we do not know the causal structure of the data generative process. To detect that on a specific noisy dataset, we proposed an intuitive method by exploiting an asymmetric property of the two different causal structures (X causes Y vs Y causes X) regarding estimating the transition matrix. The contribution of this paper is summarized as follows.

- From a causal perspective, by analyzing the generative processes of the data containing noisy labels, we found that the performance of SSL-based methods for learning with noisy labels will be influenced by different generative processes, i.e., when X causes Y, SSL-based methods can not leverage the unlabeled set (which is usually split from the noisy training set) to help learn $P(Y|X)$; when Y causes X, it is possible to leverage the unlabeled set to help learn $P(Y|X)$. In contrast, the performance of the model-based methods is not influenced by data generative processes but is usually hard to incorporate heuristics.
- We leverage the causal theory to the application of learning with label noise to help algorithm design. Our interpretation and analysis suggest that the algorithm design should be different according to the data generative process. Given the generative process of a dataset, it provides a high-level idea that whether SSL-based methods or model-based methods should be more focused on when designing the algorithm for it. This potentially can save a lot of resources for training and testing different algorithms and accelerates algorithm development for real-world applications.
- Given a dataset, we usually do not know the data generative process. Therefore, we have proposed an intuitive method for discovering whether a dataset is causal (X causes Y) or anticausal (Y causes X). To the best of our knowledge, this is the first discovery method when data contains noisy labels.

2. Related Work

In this section, we first introduce the two major streams, i.e., the methods employing semi-supervised learning and the model-based methods. Then we introduce the causal generation process of the noisy data.

SSL-based methods. Semi-supervised learning is widely employed in learning with noisy labels. To get rid of label errors, existing methods usually divide the dataset into confident examples and unconfident examples. Then the deep neural networks are trained on the confident examples in a supervised manner (Jiang et al., 2018; Han et al., 2018). To also make use of the unconfident examples that contain a large number of incorrect labels, by just employing the unlabeled instances, different semi-supervised learning techniques can be employed. For example, the consistency regularization (Laine & Aila, 2016) is employed by (Englesson & Azizpour, 2021); FixMatch (Sohn et al., 2020) is employed by (Li et al., 2019); the co-Regularization is employed by (Wei et al., 2020); contrastive learning is employed by (Li et al., 2020; Tan et al., 2021; Ciortan et al., 2021; Ghosh & Lan, 2021; Yao et al., 2021; Zheltonozhskii et al., 2022). Empirically, these methods have demonstrated state-of-the-art performance.

Model-based methods. This family of methods mainly focuses on designing statistically consistent methods by employing the noise transition matrix $\boldsymbol{T}(\boldsymbol{x})$. Specifically, given an instance $\boldsymbol{x}$, its transition matrix $\boldsymbol{T}(\boldsymbol{x})$ reveals the transition relationship from clean labels to noisy labels of the instance., i.e.,

$$\begin{aligned}&\boldsymbol{T}(\boldsymbol{x})[P(Y=1|\boldsymbol{x}),\ldots,P(Y=L|\boldsymbol{x})]^\top\\&=[P(\tilde{Y}=1|\boldsymbol{x}),\ldots,P(\tilde{Y}=L|\boldsymbol{x})]^\top.\end{aligned}$$

Let $h:\mathcal{X}\to\Delta_{C-1}$ models a class posterior distribution and ℓ_{ce} be the cross-entropy loss, then

$$\begin{aligned}&\arg\min_h \mathbb{E}_{\boldsymbol{x},y}[\ell_{ce}(y,h(\boldsymbol{x}))]\\&=\arg\min_h \mathbb{E}_{\boldsymbol{x},\tilde{y}}[\ell_{ce}(\tilde{y},\boldsymbol{T}(\boldsymbol{x})h(\boldsymbol{x}))].\end{aligned}\tag{1}$$

The above equation shows that if $\boldsymbol{T}(\boldsymbol{x})$ is given, the minimizer of the corrected loss under the noisy distribution is the

same as the minimizer of the original loss under the clean distribution (Liu & Tao, 2016; Patrini et al., 2017). In practice, $\boldsymbol{T}(\boldsymbol{x})$ usually is not given and needs to be estimated from noisy data (Xia et al., 2020; Li et al., 2021).

It is also worth mentioning that, methods focusing on designing robust loss functions can be closely related to modeling label-noise methods. These methods usually require the noise rate to help hyper-parameter selection (Zhang & Sabuncu, 2018; Liu & Guo, 2020). To calculate the noise rate, $\boldsymbol{T}(\boldsymbol{x})$ usually has to be estimated (Yao et al., 2020).

Causal generation process of noisy data. We introduce some background knowledge about causality and describe the data generative process by the causal graph and the structural causal model (SCM) (Spirtes & Zhang, 2016). Specifically, in Fig. 2(a), we illustrate a possible data generative process when data contains instance-dependent label noise by using the causal graph which represents a flow of information and reveals causal relationships among all the variables (Glymour et al., 2019). For example, Fig. 2(a) shows that the latent clean label Y is a cause of the instance X, and both X and Y are causes of $\tilde{Y}$. The generation process can also be described by a structural causal model (SCM). Specifically,

$$Y \sim P_Y,\ U_X \sim P_{U_X},\ U_{\tilde{Y}} \sim P_{U_{\tilde{Y}}},$$
$$X = f(Y, U_X),\ \tilde{Y} = g(X, Y, U_{\tilde{Y}}),$$

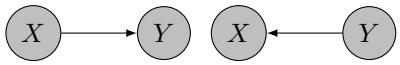
where U_X and $U_{\tilde{Y}}$ are mutually independent exogenous random variables that are also independent of Y. The occurrence of the exogenous variables models the random sampling process of X and $\tilde{Y}$. Both functions f and g can be linear or non-linear functions. Each equation species the distribution of a variable conditioned on its parents (could be an empty set). Similarly, the SCM corresponding to the causal graph in Fig. 2(b) can be written as:

$$X \sim P_X,\ U_Y \sim P_{U_Y},\ U_{\tilde{Y}} \sim P_{U_{\tilde{Y}}},$$
$$Y = f'(X, U_Y),\ \tilde{Y} = g(X, Y, U_{\tilde{Y}}).$$

Causal decomposition and modularity. By the conditional independence relations proposed by the Markov property (Pearl, 2000), the joint distribution $P(X, Y, \tilde{Y})$ when Y causes X can be factorized by following the causal direction as follows.

$$P(X, Y, \tilde{Y}) = P(Y)P(X|Y)P(\tilde{Y}|X, Y).$$

The above decomposition is called a causal decomposition. According to the *modularity property* of causal mechanisms (Schölkopf et al., 2012; Peters et al., 2017), *the conditional distribution of each variable given its causes (which could be an empty set) does not inform or influence the other conditional distributions*, which implies that all the distributions $P(Y)$, $P(X|Y)$ and $P(\tilde{Y}|X, Y)$ are disentangled.



(a) X causes Y (b) Y causes X

Figure 1: An illustration of different data generative processes without label noise. Both the instance X and the clean label Y are observable.

Similarly, when X causes Y, the causal decomposition of $P(X, Y, \tilde{Y})$ is as follows:

$$P(X, Y, \tilde{Y}) = P(X)P(Y|X)P(\tilde{Y}|X, Y).$$

3. Learning with Noisy Labels From A Causal Perspective

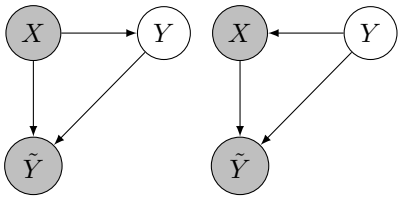
In this section, we show that the model-based method is independent of different generation processes while the SSL-based methods depend on different generation processes. We also proposed an intuitive method to detect the causal structure by exploiting an asymmetric property regarding estimating the transition matrix.

3.1. The Influence of Noisy Data Generative Processes to Different Stream of Methods

To analyze the influence of noisy data generative processes on different methods, we first explain that given a data generative process or the causal graph, whether a distribution can inform another distribution or not can be directly concluded, which is achieved by directly employing the modularity property of causal mechanisms (Peters et al., 2017). Then we explain how different generative processes of noisy data influence SSL-based methods. Specifically, we start from the simple case that analyzing relations between $P(X)$ and $P(Y|X)$ under different data generative processes without noisy labels as shown in Fig. 1.

When X causes Y illustrated in Fig. 1(a), the cause of X is an empty set. Given the definition of modularity property that the conditional distribution of each variable given its causes does not inform or influence the other conditional distributions, we can directly conclude that $P(X)$ can not inform $P(Y|X)$, i.e., $P(X)$ does not contain the relevant information of $P(Y|X)$. When Y causes X illustrated in Fig. 1(b), the cause of Y is an empty set. Then, according to the modularity property of causal mechanisms, $P(X|Y)$ can not inform $P(X)$. In this case, $P(X)$ and $P(Y|X)$ do not follow the underlying causal direction. Then they do not satisfy the modularity property anymore. Therefore, $P(X)$ can inform $P(Y|X)$. In other words, $P(X)$ generally contains the relevant information of $P(Y|X)$.

We explain why the different data generative processes can influence the performance of SSL. To make use of the unla-



(a) X causes Y (b) Y causes X

Figure 2: An illustration of different noisy data generative processes. Both the instance X and the noisy label $\tilde{Y}$ are observable, and the clean label Y is latent. We do not assume independence and instead allow for both nontrivial statistical and causal relations between the clean label Y and the noisy label $\tilde{Y}$.

beled data to help learn classifiers, SSL relies on the condition that $P(X)$ has to contain the information of $P(Y|X)$ (Schölkopf et al., 2012). When Y causes X, because $P(X)$ contains the information of $P(Y|X)$. It is possible to help learn $P(Y|X)$ by exploiting $P(X)$ by SSL-based method. An intuitive example is that, when $P(X)$ contains the information of $P(Y|X)$, it could be possible to find some low-density areas in $P(X)$, which can separate labels. Then SSL can improve the generalization ability of a classifier by exploiting these regions with unlabeled data. This is known as low density separation. However, when X causes Y, because $P(X)$ generally does not contain the information of $P(Y|X)$. It is most unlikely to find low density regions that can separate labels. Exploiting unlabeled data by using SSL then generally is not helpful. In Appendix A, we also provide the derivation and a concrete toy example to clearly illustrate the relation between $P(X)$ and $P(Y|X)$ under different data generative processes.

Now we explain how the different noisy data generative processes illustrated in Fig. 2 influence SSL-based methods and model-based methods. In learning with noisy labels, SSL-based methods change the problem setting to SSL by splitting a noisy training set into a labeled set (which is potentially clean) and an unlabeled set. Then different SSL techniques are employed which aim to improve the performance of the classifier by exploiting the unlabeled set.

When X causes Y illustrated in Fig. 2(a), X is a cause of both Y and $\tilde{Y}$. Causal modularity suggests that both $P(\tilde{Y}|X)$ and $P(Y|X)$ can not be informed by $P(X)$. Then $P(X)$ does not contain the information on either $P(\tilde{Y}|X)$ or $P(Y|X)$, the unconfident set can not help learn a classifier in general. As a result, only the confident sample which is a subset of the noisy training data is used for learning a classifier. It implies that SSL-based methods have the data sacrifice issue in this case. When Y causes X illustrated in Fig. 2(b), $P(X)$ contains the information of $P(Y|X)$ because $P(X)$ and $P(Y|X)$ not decomposed by following the underlying causal direction and do not satisfy the modularity property. Then SSL-based methods can use the constructed unlabeled set to help learn a classifier.

In contrast, the consistent model-based methods do not exploit the unlabeled set (or $P(X)$) to learn $P(Y|X)$. Specifically, these methods usually first need to estimate the noise transition matrix $T(x)$, which is usually learned in a supervised manner on the whole noisy training set and does not require exploiting $P(X)$. Then, $P(Y|X)$ can be learned by using the estimated $T(x)$ to correct the loss on the whole noisy training set, which is also learned in a supervised manner. In these processes, only supervised information is used to help learn $P(Y|X)$ but not $P(X)$. Therefore, the performance of the model-based methods is not influenced by the different data generative processes. However, these methods usually require a large number of training examples to accurately estimate the transition matrix (Yao et al., 2020). If the transition matrix is poorly estimated, the estimation error of $P(Y|X)$ will be large.

3.2. An Intuitive Method For the Causal Structure Detection

To discover the causal structure with data containing noisy labels, we provide an **c**asual structure **d**etection method for learning with **n**oisy **l**abels (i.e., CDNL estimator). To the best of our knowledge, this is the first method to discover whether X causes Y or Y causes X on noisy datasets.

Our method relies on an asymmetric property of estimating flip rates under different generalization processes, i.e., when X causes Y, the flip rate estimated by an unsupervised classification method usually has a large estimation error; when Y causes X, the estimation error is small. Specifically, let Y' be pseudo labels estimated by an unsupervised classification method. Given pseudo labels and noise labels, the flip rate $P(\tilde{Y}|Y')$ that Y' be flipped into $\tilde{Y}$ can be estimated. Let $Y^* = \arg\max_i P(Y = i|\boldsymbol{x})$ be the Bayes label on the clean class-posterior distribution. Let $P(\tilde{Y}|Y^*)$ be the underlying flip rate that the Bayes label Y^* be flipped into $\tilde{Y}$. The intuition is that given a noisy dataset, if X causes Y, $P(X)$ does not contain labeling information, then Y' should be very different from clean label Y. Therefore, the estimation error of the flip rate (the difference between $P(\tilde{Y}|Y')$ and $P(\tilde{Y}|Y^*)$) is usually large. If Y causes X, $P(X)$ contains information of $P(Y|X)$, the Y' should be "close" to clean label Y. Therefore the estimation error of $P(\tilde{Y}|Y')$ is usually small. Specifically, the estimation error is defined as follows.

$$d(P(\tilde{Y}|Y^*), P(\tilde{Y}|Y')) = \sum_{i}^{L}\sum_{j}^{L} \frac{|P(\tilde{Y}=j|Y^*=i) - P(\tilde{Y}=j|Y'=i)|}{L^2}. \quad (2)$$

Then we can discuss that how to estimate $P(\tilde{Y}|Y')$ and

Algorithm 1 CDNL Estimator

Input: a noisy training sample S_{tr}; a noisy validation sample S_{val}; a cluster algorithm z; a classification model h; a trainable stochastic matrix A

1: Optimize h and A via Eq. (5) to obtain $\hat{A}^* = \hat{P}(\tilde{Y}|Y^*)$ by employing the training set S_{tr} and the validation set S_{val};
2: Employ the cluster algorithm z to estimate the cluster IDs of all instances in training set S_{tr};
3: Obtain $\hat{Y}'$ of all instances from cluster IDs;
4: Calculate $\hat{P}(\tilde{Y}|Y)$ by Eq (4).

Output: The estimation $d(\hat{P}(\tilde{Y}|Y^*), \hat{P}(\tilde{Y}|Y'))$ via Eq. (2).

$P(\tilde{Y}|Y^*)$, respectively.

Estimation of $P(\tilde{Y}|Y')$. To estimate the flip rate $P(\tilde{Y}|Y')$, a clustering method is employed first to learn the cluster ID C for every instance. Then the cluster ID can be converted into the pseudo label Y' by calculating the overlapping between the estimated Bayes label $\hat{Y}^*$ and Cluster ID. After having the pseudo label Y', the average noise rate $P(\tilde{Y}|Y')$ obtained by a clustering method can be directly calculated. Specifically, let $C = i$ denote the cluster label i, and let $S_{C_i} = \{\boldsymbol{x}_j\}_{j=0}^{N_{C_i}}$ denote the instance with cluster label i. Similarly let $S_{\hat{Y}_j^*} = \{\boldsymbol{x}_k\}_{k=0}^{N_{\hat{Y}_j^*}}$ denote the instance with estimated Bayes label j by employing label-noise learning methods (Patrini et al., 2017). We assign the pseudo labels $\hat{Y}'$ of all instances in set S_{C_i} be the dominated estimated Bayes label $\hat{Y}^*$, i.e.,

$$\hat{Y}' = \arg\max_{j \in L} \frac{\sum_{x_k \in S_{\hat{Y}_j^*}} \mathbb{1}_{\{x_k \in S_{C_i}\}}}{N_{C_i}}. \tag{3}$$

Empirically, the assignment is implemented by applying Hungarian algorithm (Jonker & Volgenant, 1986). After the assignment, the pseudo labels of all training examples can be obtained. Then $P(\tilde{Y}|Y')$ can be estimated via counting on training examples, i.e.,

$$\hat{P}(\tilde{Y} = j|Y' = i) = \frac{\sum_{(\boldsymbol{x},\tilde{y},\hat{y}')} \mathbb{1}_{\{\hat{Y}'=i \wedge \tilde{y}=j\}}}{\sum_{(\boldsymbol{x},\tilde{y},\hat{y}')} \mathbb{1}_{\{\hat{Y}'=i\}}}, \tag{4}$$

where $\mathbb{1}_{\{.\}}$ is an indicator function, $(\boldsymbol{x}, \tilde{y}, \hat{y}')$ is a training example with the estimated pseudo label, and $\wedge$ represents the AND operation.

It is worth mentioning that the performance of the proposed CDLN estimator relies on the backbone unsupervised classification method. When Y causes X, the backbone method is expected to have reasonable classification accuracy on training instances. Thanks to the great success of the unsupervised learning methods (Likas et al., 2003; Niu et al., 2021; Ghosh & Lan, 2021; Zhou et al., 2021), some of these methods can even have compatible performance with the supervised learning on some benchmark datasets such as STL10 (Coates et al., 2011) and CIFAR10 (Krizhevsky et al., 2009).

Estimation of $P(\tilde{Y}|Y^*)$. We directly estimate the average flip rate $P(\tilde{Y}|Y^*)$ in an end-to-end manner. Specifically, let f be a deep classification model that outputs the estimated Bayes label in a one-hot fashion. (Jang et al., 2016). The distribution $P(\tilde{Y}|Y^*)$ is modeled by a trainable diagonally dominant column stochastic matrix A. Similar to the state-of-the-art method (Li et al., 2021), the matrix A and the classifier f are optimized in an end-to-end manner. They are estimated by minimizing a constrained cross-entropy loss on noisy data, i.e.,

$$\{\hat{A}^*, \hat{f}\} = \arg\min_{A,f} \frac{1}{N} \sum_{\boldsymbol{x},\tilde{y}} \ell_{ce}(\tilde{y}, Ah(\boldsymbol{x})),$$
$$s.t. \max_i h_i(\boldsymbol{x}) = 1. \tag{5}$$

The constraint that $\max_i h_i(\boldsymbol{x}) = 1$ is to let the model output the Bayes label (in a one-hot fashion). Empirically, it can be achieved by employing Gumbel-Softmax (Jang et al., 2016) which is differentiable.

It is worth mentioning that $P(\tilde{Y}|Y^*)$ can be estimated by employing existing methods that learn the noise transition matrix $P(\tilde{Y}|Y, X)$. Specifically, to estimate $P(\tilde{Y}|Y^*)$ with existing methods, $P(\tilde{Y}|X)$ and $P(\tilde{Y}|Y, X)$ have to be learned first. Then both the estimated clean label Y and the Bayes label Y^* can be revealed by (1). After that, $P(\tilde{Y}|Y^*)$ can be estimated by using the same technique as in Eq. (4). However, $P(\tilde{Y}|Y, X)$ usually is hard to estimate (Xia et al., 2020), which leads to the learned classifier (in (1)) and Bayes labels being poorly estimated. As a result, $\hat{P}(\tilde{Y}|Y^*)$ will contain a large estimation error. Therefore, we propose to avoid learning $P(\tilde{Y}|Y, X)$ and directly estimate the average flip rate $P(\tilde{Y}|Y^*)$ in an end-to-end manner. This is achieved by letting h directly estimate Bayes labels but not $\hat{P}(Y|X)$. By reducing the output complexity of h from a continuous distribution $\hat{P}(Y|X)$ to a discrete distribution, the learning difficulty of $P(\tilde{Y}|Y^*)$ can be reduced. In Section 4.1.1, we have also shown that the estimation error of $P(\tilde{Y}|Y^*)$ by employing our method above is much smaller than employing the state-of-the-art method VolMinNet (Li et al., 2021) for both instance-dependent and instance-independent label noise.

Theoretical analysis of CDNL estimator. Here, we formally justify that when X causes Y, the average flip rate $P(\tilde{Y}|Y')$ estimated by an unsupervised classification method usually has a large estimation error. However, when Y causes X, the estimation error is usually small.

Theorem 3.1. *Let $P(\tilde{Y}|Y^*)$ be the transition relationship*

from the noisy label $\tilde{Y}$ to the clean Bayes label Y^*; let $P(\tilde{Y}|Y')$ be the transition relationship from the noisy label $\tilde{Y}$ to the pseudo label Y'. Then the estimation error is

$$d(P(\tilde{Y}|Y'), P(\tilde{Y}|Y^*)) = \frac{1}{L^2}\sum_{i}^{L}\sum_{j}^{L}\frac{1}{P(Y^*=j)}$$
$$\left|\mathbb{E}_{P(X)}\left[\mathbb{1}_{\{\tilde{f}(X)=i\}}\left(P(Y'=j|X)\frac{P(Y^*=j)}{P(Y'=j)} - P(Y^*=j|X)\right)\right]\right|.$$

From the above theorem, we can find out that when the class posterior of pseudo label $P(Y'|X)$ and the class posterior $P(Y^*|X)$ of Bayes label are similar, the estimation error is small. Specifically, when $P(Y'|X)$ and $P(Y^*|X)$ are similar, $P(Y)$ and $P(Y')$ are also similar, because $P(Y') = \mathbb{E}_{P(X)}[P(Y'|X)]$ and $P(Y^*) = \mathbb{E}_{P(X)}[P(Y^*|X)]$. Then, $P(Y'=j|X=x)\frac{P(Y^*=j)}{P(Y'=j)} - P(Y^*=j|X=x) = 0$ is small, and the estimation error $d(P(\tilde{Y}|Y'), P(\tilde{Y}|Y^*))$ is small. When Y causes X, $P(X)$ can inform $P(Y^*|X)$, then $P(Y'|X)$ learned by exploiting $P(X)$ is close to $P(Y^*|X)$. Therefore, the estimation error is usually small. When X causes Y, $P(X)$ can not inform $P(Y^*|X)$, then $P(Y'|X)$ and $P(Y^*|X)$ should have a large difference. Therefore, the estimation error is usually large.

Theorem 3.1 also shows that when $P(Y'|X)$ and $P(Y^*|X)$ are identical, the estimation error $d(P(\tilde{Y}|Y'), P(\tilde{Y}|Y^*))$ is 0. This is because in this case, $P(Y)$ also identical to $P(Y')$. Then, $P(Y'=j|X=x)\frac{P(Y^*=j)}{P(Y'=j)} - P(Y^*=j|X=x) = 0$ for all x, and the estimation error is 0.

4. Experiments

In this section, we illustrate the performance of the proposed estimator and different methods under different data generative processes with the existence of label noise.

Baselines. We illustrate the performance of state-of-the-art model-based methods and SSL-based methods. The model-based methods employed are (i) Forward (Patrini et al., 2017) which estimates the transition matrix and embeds it to the neural network; (ii) Reweighting (Liu & Tao, 2016) which gives training examples with different weights according to the transition matrix by importance reweighting; (iii) T-Revision (Xia et al., 2019) which refines the learned transition matrix to improve the classification accuracy. The SSL-based methods employed are (iv) JoCoR (Wei et al., 2020) which aims to reduce the diversity of two networks during training; (v) MoPro (Li et al., 2020) which is a contrastive learning method that achieves online label noise correction (vi) Dividemix (Li et al., 2019) which leverages the techniques FixMatch (Sohn et al., 2020) and Mixup (Zhang et al., 2018); (viii) Mixup (Zhang et al., 2018) which trains a neural network on convex combinations of pairs of examples and their labels. For all baseline methods, we follow the hyper-parameters settings mentioned in their original paper. It is worth noting that, MoPro focuses on image datasets, to let it work for non-image datasets, we replace the strong data augmentation for images with small Gaussian Noise, which may influence its performance.

Datasets and noise types. We have employed 2 synthetic datasets that are XYgaussian and YXguassian and 6 real-world datasets which are KrKp, Balancescale, Splice, Waveform, MNIST, and CIFAR10. The *causal datasets* generated from X to Y are KrKp, Balancescale and Splice. The rest are *anticausal datasets* generated from Y to X. Due to the limited space, the results on Balancescale and Waveform are included in Appendix B.2. We manually inject label noise into all datasets, and 20% of data is left as the validation set. Three types of noise in our experiments are employed in our experiments. (1) symmetry flipping (Sym) (Patrini et al., 2017) which randomly replaces a percentage of labels in the training data with all possible labels. (2) pair flipping (Pair) (Han et al., 2018) where labels are only replaced by similar classes. (3) instance-dependent Label Noise (IDN) (Xia et al., 2020) where different instances have different transition matrices depending on parts of instances.

Network structure and optimization. For a fair comparison, we implement all methods by PyTorch. All the methods are trained on Nvidia Geforce RTX 2080 GPUs. For non-image datasets, a 2-hidden-layer network with batch normalization (Ioffe & Szegedy, 2015) and dropout (0.25) (Srivastava et al., 2014) is employed as the backbone method for all baselines. We employ LeNet-5 for MNIST (LeCun, 1998) dataset and ResNet-18 (He et al., 2016) for CIFAR10 (Krizhevsky et al., 2009). To estimate $P(\tilde{Y}|Y^*)$, we use SGD to train the classification network with batch size 128, momentum 0.9, and weight decay 10^{-4}. The initial learning rate is 10^{-2}, and it decays at 30th and 60th epochs at the rate 0.1, respectively. To get $P(\hat{Y}|Y')$, for XYguassian, yxGuassain, KrKp, Balancescale, Splice and Waveform and MNIST, K-means clustering method (Likas et al., 2003) is employed; for CIFAR10, the SPICE* (Niu et al., 2021) clustering method is employed.

4.1. Experiments on Synthetic Datasets

4.1.1. Estimation error of $P(\tilde{Y}|Y^*)$

In Fig. 3, we compare the estimation error of average flip rate $P(\tilde{Y}|Y^*)$ of our CDNL estimator and the state-of-the-art method VolMinNet (Li et al., 2021), respectively. To let VolMinNet estimate $P(\tilde{Y}|Y^*)$, we first train VolMinNet with a noisy training set and select the best model by using the validation set, then the estimated clean class-posterior distribution $\hat{P}(Y|X)$ is obtained. The Bayes label Y^* can be directly obtained via $\hat{P}(Y|X)$, and $P(\tilde{Y}|Y^*)$ can be

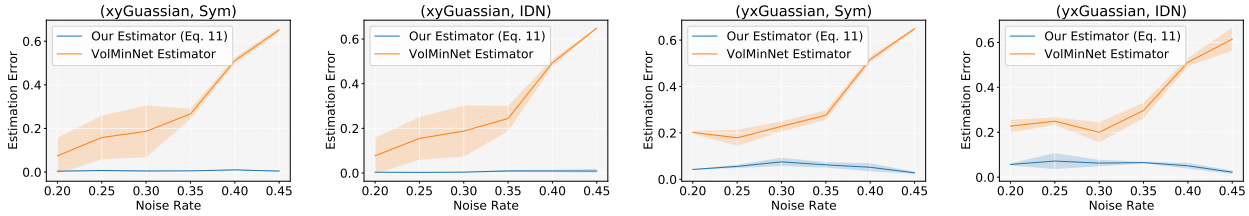


Figure 3: Estimation error of $P(\tilde{Y}|Y^*)$ on synthetic datasets with instance-independent and instance-dependent label noise. Our estimator outperforms the state-of-the-art method by a large margin.

Table 1: Test accuracies (%) of different methods on XYgaussian (causal) and YXgaussian (anticausal) datasets with different types of label noise. Estimation errors obtained by CDNL estimator are shown in the parentheses after noise rates.

XYguassian		Sym			Instance	
(causal)	20% (0.196)	30% (0.131)	40% (0.142)	20% (0.101)	30 (0.127)	40% (0.191)
Forward	98.98±0.15	98.28±0.48	96.46±1.07	98.98±0.23	98.60±0.19	97.29±0.51
Reweighting	99.26±0.23	**98.57**±0.34	**96.85**±0.71	99.42±0.30	98.42±0.35	97.14±1.07
T-Revision	**99.32**±0.24	98.55±0.37	96.82±0.72	**99.45**±0.28	**98.55**±0.65	**97.22**±0.91
JoCoR (SSL)	98.19±0.26	89.66±5.31	89.12±19.43	99.03±0.08	89.51±9.62	73.41±13.44
MoPro (SSL)	96.41±0.45	95.70±0.93	77.32±6.99	95.98±0.87	94.63±0.64	77.79±8.95
Dividemix (SSL)	97.20±0.25	96.98±0.12	95.39±0.83	97.15±0.56	97.13±0.17	90.72±0.47
Mixup (SSL)	97.15±0.14	96.88±0.35	94.12±0.76	96.93±0.44	96.15±0.65	87.68±9.06
YXguassian		Sym			Instance	
(antiausal)	20% (0.026)	30% (0.031)	40% (0.028)	20% (0.027)	30 (0.031)	40% (0.043)
Forward	86.26±0.13	85.97±0.19	84.85±0.93	86.10±0.11	85.56±0.47	83.94±2.14
Reweighting	86.31±0.18	85.85±0.27	84.68±0.55	86.22±0.25	86.03±0.26	84.19±0.84
T-Revision	**86.32**±0.17	85.81±0.32	84.42±0.56	86.25±0.23	86.02±0.23	84.18±0.83
JoCoR (SSL)	86.26±0.10	85.99±0.09	85.86±0.21	86.16±0.14	86.13±0.14	85.43±0.34
MoPro (SSL)	84.79±0.72	84.17±0.61	83.67±1.32	85.36±0.63	84.43±1.27	81.07±3.03
Dividemix (SSL)	**86.32**±0.20	**86.28**±0.11	**86.23**±0.19	**86.37**±0.09	**86.37**±0.12	**86.06**±0.15
Mixup (SSL)	86.15±0.19	85.64±0.63	82.48±2.56	85.74±0.43	85.01±0.92	81.47±5.76

estimated by using the same technique as in Eq. (4). As illustrated in Fig. 3, it shows that the estimation error of our method is close to 0 on both instance-independent label noise and instance-dependent label noise, which is much smaller than the estimated error of VolMinNet. This empirically validated the advantage of CDNL estimator that directly estimates the average noise rates but does not require learning the transition matrix for each instance.

To validate the correctness of our method, we have generated a causal dataset (from X to Y) and an anticausal dataset (from Y to X). For both datasets, $P(X)$ is a multivariate Gaussian mixture of $\mathcal{N}(0, \boldsymbol{I})$ and $\mathcal{N}(1, \boldsymbol{I})$ with dimension 5. For the causal dataset XYguassian, the causal association f and f' between X and Y are set to be linear. The parameter of the linear function is randomly drawn from the $\mathcal{N}(0, \boldsymbol{I})$. For YXguassian, we let the label be the mean value of the multivariate Gaussian distribution. For both datasets, we have balanced the positive and negative class priors to 0.5 and the sample size is 10000. The results with a large sample size are included in Appendix B.1.

4.1.2. Classification Accuracies

The estimation error $d(\hat{P}(\tilde{Y}|Y^*), \hat{P}(\tilde{Y}|Y'))$ obtained by the proposed CDNL estimator and the test accuracies of model-based methods and SSL-based methods are illustrated in Tab. 3. Estimation errors are shown in the parentheses after noise rates, and the estimation error is averaged over 5 repeated trials.

The result validates that $P(X)$ contains labeling information and can help learn $P(Y|X)$ on the anticausal dataset. Specifically, on the causal dataset (XYguassian), model-based methods perform better than SSL-based methods. On the anticausal dataset (YXguassian), SSL-based methods outperform model-based methods under the same setting. Moreover, with the increase of the noise rate, the performance of the SOTA method DivideMix drops dramatically on the causal dataset, but its performance is relatively stable on anticausal dataset.

The result also shows that estimation errors on the anticausal dataset YXgaussian are at least 2 times smaller than the causal dataset XYguassian, which validated Theorem 3.1.

Table 2: Comparing test accuracies (%) of different methods on causal and anticausal datasets with different levels and types of label noise. Estimation errors obtained by employing CDNL estimator are shown in the parentheses after noise rates.

KrKp	Sym			Instance		
(causal)	20% (0.297)	30% (0.196)	40% (0.070)	20% (0.262)	30% (0.166)	40% (0.072)
Forward	93.31±1.0	89.31±1.96	77.78±7.4	94.0±0.8	87.25±3.1	80.75±2.31
Reweighting	93.88±1.43	91.16±1.09	77.31±5.26	93.5±2.63	89.25±1.53	78.22±6.61
T-Revision	**94.72**±0.62	**91.81**±1.93	**77.97**±5.0	**94.5**±1.63	**90.78**±2.35	**79.06**±4.89
JoCoR (SSL)	93.69±0.23	89.53±0.84	67.81±2.07	93.44±0.71	87.44±2.95	67.75±6.51
MoPro (SSL)	89.47±1.13	79.47±7.03	65.94±2.06	89.31±3.82	79.59±6.2	62.62±4.78
Dividemix (SSL)	93.75±0.32	88.31±0.65	74.31±1.44	93.47±0.15	93.34±0.72	63.94±1.45
Mixup (SSL)	93.31±1.1	88.81±1.03	73.84±1.18	93.19±1.31	87.25±1.49	74.31±3.42
Splice	Sym		Pair		Instance	
(causal)	20% (0.136)	40% (0.146)	20% (0.140)	40% (0.148)	20% (0.151)	40% (0.153)
Forward	71.25±3.07	66.18±3.61	73.73±1.03	65.8±3.67	65.8±4.08	61.6±5.67
Reweighting	76.96±1.69	71.91±2.68	**75.55**±1.88	**66.68**±1.54	75.64±1.95	**63.54**±7.21
T-Revision	**76.99**±1.73	**71.94**±2.68	75.49±2.05	66.61±1.5	**75.67**±1.89	63.45±7.17
JoCoR (SSL)	69.81±4.61	63.2±1.89	59.37±1.44	57.71±3.7	59.66±2.44	55.3±5.87
MoPro (SSL)	53.6±0.19	53.51±0.0	53.51±0.0	53.25±0.43	53.79±0.38	52.17±3.27
Dividemix (SSL)	75.11±1.66	53.45±0.0	53.45±0.0	56.14±2.1	59.97±0.55	51.41±1.79
Mixup (SSL)	67.43±3.2	62.16±2.52	68.15±2.63	63.67±6.63	65.52±2.22	49.03±9.86
MNIST	Sym		Pair		Instance	
(anticausal)	20% (0.034)	40% (0.038)	20% (0.041)	40% (0.20)	20% (0.025)	40% (0.026)
Forward	98.75±0.08	97.86±0.22	98.84±0.10	94.92±0.89	96.87±0.15	90.30±0.61
Reweighting	98.71±0.11	98.13±0.19	98.54±.63	91.50±1.27	97.99±0.13	90.30±0.61
T-Revision	98.91±0.04	98.34±0.21	98.89±0.08	91.83±1.08	98.39±0.09	96.50±0.31
JoCoR (SSL)	98.06±0.13	96.64±0.19	98.01±0.19	96.85±0.43	98.62±0.06	96.07±0.31
MoPro (SSL)	98.51±0.92	95.14±1.23	96.79±1.04	94.96±1.32	98.53±0.52	96.45±1.20
Dividemix (SSL)	**99.24**±0.03	**99.21**±0.05	**99.25**±0.03	**98.50**±0.08	**99.31**±0.02	**97.75**±0.1
Mixup (SSL)	97.45±0.21	95.75±0.43	97.57±1.08	92.46±1.43	96.54±1.20	90.38±1.30
CIFAR10	Sym		Pair		Instance	
(anticausal)	20% (0.010)	40% (0.009)	20% (0.010)	40% (0.026)	20% (0.037)	40% (0.042)
Forward	88.21±0.48	78.44±0.89	88.21±0.48	77.44±6.89	85.29±0.38	74.72±3.24
Reweighting	86.77±0.40	83.16±0.46	89.60±1.01	77.06±6.47	88.72±0.41	84.52±2.65
T-Revision	90.33±0.52	84.94±2.58	89.75±0.41	80.94±2.58	90.46±0.13	85.37±3.36
JoCoR (SSL)	85.96±0.25	79.65±0.43	80.33±0.20	71.62±1.05	89.80±0.28	73.78±1.39
MoPro (SSL)	78.15±0.15	67.70±0.56	77.92±0.81	69.89±1.02	78.75±0.15	67.61±0.24
Dividemix (SSL)	**95.60**±0.10	**94.80**±1.10	**95.72**±0.04	**87.02** ±0.41	**95.50**±1.17	**94.50**±0.23
Mixup (SSL)	93.20±0.31	86.20±0.30	92.23±0.71	82.43±1.02	93.32±0.25	87.61±0.56

Specifically, when X is a cause of Y (anticausal), estimation errors $d(\hat{P}(\tilde{Y}|Y^*), \hat{P}(\tilde{Y}|Y'))$ are larger than 0.1; when Y is a cause of X, all estimation errors are smaller than 0.05. We therefore empirically use 0.05 as a threshold to distinguish different data generative processes on real-world datasets.

4.2. Experiments on Real-World Datasets

We illustrate estimations of CDNL estimator and test accuracies of different methods on real-world datasets in Tab. 2. Due to the limited space, the results on Balancescale and Waveform are included in Appendix B.2. The results show that CDNL estimator can successfully determine the causal structure of all datasets except Waveform by employing the threshold 0.05 validated on Synthetic datasets. Specifically, on all anticausal datasets except Waveform, the estimation error obtained by employing CDNL estimator is lower than 0.05, and SSL-based methods demonstrate their effectiveness. On all causal datasets, the estimation error is much larger than 0.05, and model-based methods can have better performance than SSL-based methods.

For Waveform, although it is an anticausal dataset, model-based methods have better performance than SSL-based methods, and the estimation error is also large. The reason can be that 1). $P(X)$ contains information of $P(Y|X)$, but the information contained is limited, or 2). The information of $P(Y|X)$ contained in $P(X)$ is hard to be exploited by existing unsupervised methods.

5. Conclusion

In this paper, we have investigated the influence of the data generative process containing noisy labels on SSL-based

methods and model-based methods. We show that the performance of SSL-based methods depends on the data generative process, while model-based methods are not influenced by the data generative process. Our analysis suggests that for different data generative processes, different streams of methods should be focused, or a hybrid method should be designed in the future that can simultaneously model label noise and leverage SSL to improve the model's robustness. To detect data generative processes, we have also proposed CDNL estimator which exploits the asymmetric property of estimating the flip rate under different generative processes.

Acknowledgements

MG was supported by ARC DE210101624. JY is sponsored by Natural Science Foundation of China (62276242), CAAI-Huawei MindSpore Open Fund (CAAIXSJLJJ-2021-016B, CAAIXSJLJJ-2022-001A), Anhui Province Key Research and Development Program (202104a05020007), USTC-IAT Application Sci. & Tech. Achievement Cultivation Program (JL06521001Y), Sci. & Tech. Innovation Special Zone (20-163-14-LZ-001-004-01). BH was supported by NSFC Young Scientists Fund No. 62006202 and Guangdong Basic and Applied Basic Research Foundation No. 2022A1515011652. KZ was supported in part by the NSF-Convergence Accelerator Track-D award #2134901, by the National Institutes of Health (NIH) under Contract R01HL159805, by grants from Apple Inc., KDDI Research, Quris AI, and IBT, and by generous gifts from Amazon, Microsoft Research, and Salesforce.

References

Ciortan, M., Dupuis, R., and Peel, T. A framework using contrastive learning for classification with noisy labels. *Data*, 6(6):61, 2021.

Coates, A., Ng, A., and Lee, H. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215–223. JMLR Workshop and Conference Proceedings, 2011.

Englesson, E. and Azizpour, H. Consistency regularization can improve robustness to label noise. *arXiv preprint arXiv:2110.01242*, 2021.

Ghosh, A. and Lan, A. Contrastive learning improves model robustness under label noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2703–2708, 2021.

Glymour, C., Zhang, K., and Spirtes, P. Review of causal discovery methods based on graphical models. *Frontiers in genetics*, 10:524, 2019.

Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I., and Sugiyama, M. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, pp. 8527–8537, 2018.

Hauser, A. and Bühlmann, P. Characterization and greedy learning of interventional markov equivalence classes of directed acyclic graphs. *Journal of Machine Learning Research*, 13(79):2409–2464, 2012.

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

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pp. 448–456. PMLR, 2015.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

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

Jonker, R. and Volgenant, T. Improving the hungarian assignment algorithm. *Operations Research Letters*, 5(4): 171–175, 1986.

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

Laine, S. and Aila, T. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

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

Li, J., Socher, R., and Hoi, S. C. Dividemix: Learning with noisy labels as semi-supervised learning. In *ICLR*, 2019.

Li, J., Xiong, C., and Hoi, S. C. Mopro: Webly supervised learning with momentum prototypes. *arXiv preprint arXiv:2009.07995*, 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.

Likas, A., Vlassis, N., and Verbeek, J. J. The global k-means clustering algorithm. *Pattern recognition*, 36(2):451–461, 2003.

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

Liu, Y. and Guo, H. Peer loss functions: Learning from noisy labels without knowing noise rates. In *ICML*, pp. 6226–6236. PMLR, 2020.

Nguyen, D. T., Mummadi, C. K., Ngo, T. P. N., Nguyen, T. H. P., Beggel, L., and Brox, T. Self: Learning to filter noisy labels with self-ensembling. In *ICLR*, 2019.

Niu, C., Shan, H., and Wang, G. Spice: Semantic pseudo-labeling for image clustering. *arXiv preprint arXiv:2103.09382*, 2021.

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

Pearl, J. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, New York, NY, USA, 2000. ISBN 0-521-77362-8.

Peters, J., Janzing, D., and Schölkopf, B. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

Schölkopf, B., Janzing, D., Peters, J., Sgouritsa, E., Zhang, K., and Mooij, J. On causal and anticausal learning. In *29th International Conference on Machine Learning (ICML 2012)*, pp. 1255–1262. International Machine Learning Society, 2012.

Sohn, K., Berthelot, D., Li, C.-L., Zhang, Z., Carlini, N., Cubuk, E. D., Kurakin, A., Zhang, H., and Raffel, C. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020.

Spirtes, P. and Zhang, K. Causal discovery and inference: concepts and recent methodological advances. In *Applied informatics*, volume 3, pp. 1–28. SpringerOpen, 2016.

Spirtes, P., Glymour, C. N., Scheines, R., and Heckerman, D. *Causation, prediction, and search*. MIT press, 2000.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Tan, C., Xia, J., Wu, L., and Li, S. Z. Co-learning: Learning from noisy labels with self-supervision. In *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 1405–1413, 2021.

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.

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

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

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

Yao, Y., Sun, Z., Zhang, C., Shen, F., Wu, Q., Zhang, J., and Tang, Z. Jo-src: A contrastive approach for combating noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5192–5201, 2021.

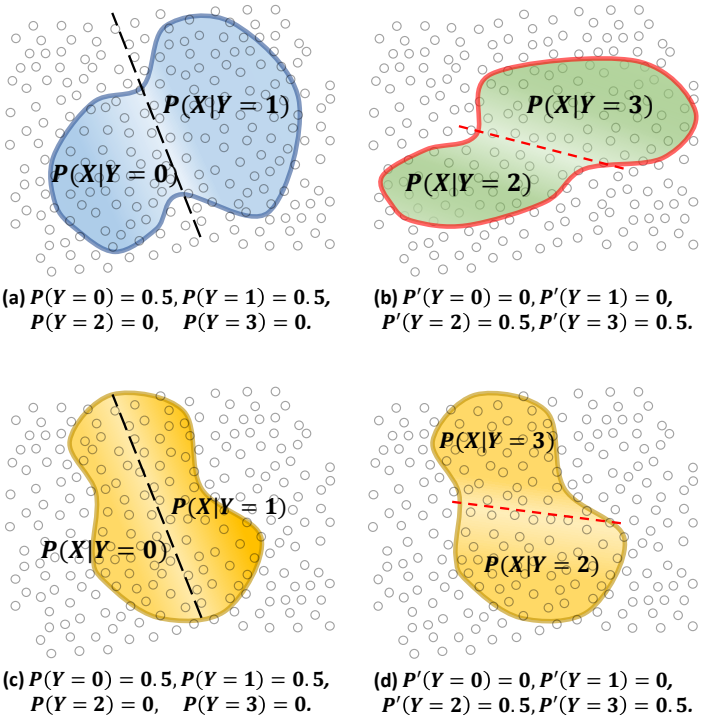
Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.

Zhang, K., Zhang, J., and Schölkopf, B. Distinguishing cause from effect based on exogeneity. *arXiv preprint arXiv:1504.05651*, 2015.

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

Zheltonozhskii, E., Baskin, C., Mendelson, A., Bronstein, A. M., and Litany, O. Contrast to divide: Self-supervised pre-training for learning with noisy labels. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1657–1667, 2022.

Zhou, J., Wei, C., Wang, H., Shen, W., Xie, C., Yuille, A., and Kong, T. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021.



(a) $P(Y=0)=0.5, P(Y=1)=0.5, P(Y=2)=0, \quad P(Y=3)=0.$

(b) $P'(Y=0)=0, P'(Y=1)=0, P'(Y=2)=0.5, P'(Y=3)=0.5.$

(c) $P(Y=0)=0.5, P(Y=1)=0.5, P(Y=2)=0, \quad P(Y=3)=0.$

(d) $P'(Y=0)=0, P'(Y=1)=0, P'(Y=2)=0.5, P'(Y=3)=0.5.$

Figure 4: (a)-(d) illustrate the influence to $P(X)$ when $P(Y)$ changes under different data generative processes. When Y causes X, as illustrated in (a) and (b), changing $P(Y)$ to $P'(Y)$ influences $P(X)$, then $P(X)$ contains labeling information; when X causes Y, as illustrated in (c) and (d), changing $P(Y)$ to $P'(Y)$ does not influence $P(X)$, then $P(X)$ does not contain labeling information.

A. Entanglement between $P(Y|X)$ and $P(X)$.

To clearly illustrate the entanglement, we will derive that, when Y causes X, $P(Y|X)$ and $P(X)$ will change simultaneously to $P'(Y|X)$ and $P'(X)$ if we *intervene* on Y, i.e., change $P(Y)$ to a different distribution $P'(Y)$.

Specifically, when $P(Y)$ is changed to $P'(Y)$, $P(X|Y)$ will not be influenced because of the modularity property (Pearl, 2000). Since $P(Y)$ is changed to $P'(Y)$, and $P(X|Y)$ remains fixed, after the intervention, the joint distribution $P(X,Y) = P(Y)P(X|Y)$ will be changed to a new joint distribution $P'(X,Y) = P'(Y)P(X|Y)$. Then $P(X)$ will be changed to $P'(X) = \int_y P'(Y)P(X|Y)\mathrm{d}y$. By applying Bayes' rule, $P(Y|X) = P(Y)P(X|Y)/P(X)$ will change to a different distribution $P'(Y|X) = P'(Y)P(X|Y)/P'(X)$ unless $P'(Y)/P'(X) = P(Y)/P(X)$ which is a special case. Therefore, $P(Y|X)$ and $P(X)$ generally are entangled when Y causes X.

To provide more intuition, we illustrate a toy example in Fig. 4. For example, as illustrated in Fig. 4(a), when $P(Y=0) = P(Y=1) = 0.5, P(Y=2) = P(Y=3) = 0$, the data is drawn from either $P(X|Y=0)$ or $P(X|Y=1)$, then $P(X) = 0.5P(X|Y=0) + 0.5P(X|Y=1)$. However, if the class prior is changed to $P'(Y=0) = P'(Y=1) = 0, P'(Y=2) = P'(Y=3) = 0.5$, as illustrated in Fig. 4(b), instead of drawing data belonging to $Y=0$ and $Y=1$, the data belonging to $Y=2$ and $Y=3$ will be drawn, and the data distribution becomes $P'(X) = 0.5P(X|Y=2) + 0.5P(X|Y=3)$. Meanwhile, the change in $P(Y)$ also leads to a change in $P(Y|X)$. The changes of $P(X)$ and $P(Y|X)$ both come from changes of $P(Y)$, indicating that $P(X)$ contains information of $P(Y|X)$. Therefore the SSL-based methods can be useful in this case.

When feature X is a cause of Y, intervention on $P(Y)$ will change the function f' or the distribution of U_Y but leave $P(X)$ unchanged. For example, from Fig. 4(c) to Fig. 4(d), the function f' will be changed to output $Y=0$ or $Y=1$ instead of $Y=2$ or $Y=3$ to account for the label distribution change. The change of the selected label sets will only change the classification rules (tasks). It is clear that relabeling the sampled data points with different labels according to the new rules

Table 3: Test accuracies (%) of different methods on XYgaussian (causal) and YXgaussian (anticausal) datasets with different types of label noise. Estimation errors obtained by CDNL estimator are shown in the parentheses after noise rates.

XYguassian		Sym			Instance	
(causal)	20% (0.196)	30% (0.142)	40% (0.131)	20% (0.150)	30 (0.127)	40% (0.171)
Forward	98.9±0.21	98.35±0.19	96.98±0.37	98.85±0.17	98.29±0.24	96.72±0.63
Reweighting	98.61±0.10	**99.01**±0.12	96.42±1.2	99.54±0.23	99.25±0.28	**98.37**±0.61
T-Revision	**99.44**±0.12	98.11±0.12	**97.08**±1.48	**99.54**±0.23	**99.26**±0.22	98.36±0.59
JoCoR (SSL)	98.05±0.03	97.63±0.16	97.11±0.19	98.0±0.11	97.65±0.21	97.26±0.09
MoPro (SSL)	96.75±0.67	95.5±1.3	79.76±4.95	95.85±0.87	95.26±1.78	78.24±6.1
Dividemix (SSL)	97.58±0.4	96.13±0.95	93.31±2.17	96.61±1.05	95.98±1.56	94.14±2.28
Mixup (SSL)	96.86±0.59	96.06±0.63	92.55±1.54	97.0±0.46	96.44±0.51	93.57±0.71
YXguassian		Sym			Instance	
(anticausal)	20% (0.021)	30% (0.008)	40% (0.005)	20% (0.023)	30 (0.013)	40% (0.005)
Forward	86.28±0.19	**86.04**±0.14	**85.24**±0.41	86.22±0.12	85.98±0.23	85.64±0.43
Reweighting	86.23±0.14	85.19±0.25	85.13±0.68	86.39±0.11	**86.04**±0.26	85.54±0.39
T-Revision	**86.43**±0.13	85.2±0.12	85.23±0.32	**86.4**±0.27	86.03±0.25	85.54±0.39
JoCoR (SSL)	86.14±0.08	85.88±0.22	85.23±0.53	86.04±0.09	85.86±0.28	85.1±0.26
MoPro (SSL)	85.17±0.71	83.73±1.32	81.11±2.35	85.17±0.49	84.4±0.54	82.2±1.06
Dividemix (SSL)	85.03±1.07	85.9 ±0.28	85.09±1.34	85.8±0.85	85.74±0.54	**85.8**±0.36
Mixup (SSL)	85.92±0.48	84.3±2.34	82.62±2.78	86.2±0.22	85.62±0.55	82.08±4.57

Table 4: Test accuracies (%) of different methods on Balancescale (causal) with different types of label noise. Estimation errors obtained by CDNL estimator are shown in the parentheses after noise rates.

Balancescale	Sym		Pair		Instance	
(Causal)	20% (0.099)	40% (0.071)	20% (0.113)	40% (0.109)	20% (0.110)	40% (0.090)
Forward	74.24±8.74	78.8±10.53	83.36±2.23	72.48±9.12	75.36±5.53	69.6±9.71
Reweighting	89.76±3.37	89.28±1.87	**94.08**±2.41	79.36±15.02	**90.72**±2.8	**86.24**±1.38
T-Revision	**92.64**±0.93	**89.76**±3.14	92.32±3.97	**81.12**±13.91	89.12±3.45	85.28±2.06
JoCoR (SSL)	76.96±3.87	58.08±13.43	72.32±10.43	60.16±12.88	73.28±4.34	51.2±6.13
MoPro (SSL)	84.29±2.38	84.13±1.81	84.73±3.16	80.79±7.93	86.19±2.59	78.1±7.28
Dividemix (SSL)	88.16±0.32	86.56±0.93	81.12±0.39	62.96±1.47	87.52±0.64	79.04±1.18
Mixup (SSL)	86.08±2.51	83.68±3.49	86.72±1.3	67.68±17.1	84.96±2.17	75.36±5.46

will not influence the distribution of the sampled data points $P(X)$, and $P(X)$ is disentangled with the different label sets. Then $P(X)$ generally does not contain information to learn clean label Y. Therefore the SSL-based methods may not work well in this case.

B. Additional experiments

B.1. Results on Synthetic Datasets with A Large Sample Size

We increase the sample size for both XYguassian and YXguassian from 10000 to 20000. The experiment settings are the same as in our main paper. The results show that on the causal dataset XYguassian, model-based methods perform better than SSL-based methods. It is because that $P(X)$ does not contain information of $P(Y|X)$, then SSL-based methods may not be helpful. On the anticausal dataset YXguassian, model-based methods also perform better than SSL-based methods. The reason is that given sufficiently a large amount of training data, the advantage of model-based methods (See Tab. 2 that is statistically consistent will be demonstrated. However, for the real-world application, the dataset can contain high dimensional features such as image datasets. In such a case, because of the curse of dimensionality, the training sample size usually is insufficient. Therefore, SSL-based methods can also be important and have many real-world applications.

B.2. More Results on Real World Datasets

As mentioned in our main paper, although Waveform is an anticausal dataset, model-based methods have better performance than SSL-based methods, and the estimation error is also large. The reason can be that 1). $P(X)$ contains information

Table 5: Test accuracies (%) of different methods on Waveform (anticausal) datasets with different types of label noise. Estimation errors obtained by CDNL estimator are shown in the parentheses after noise rates.

Waveform	Sym		Pair		Instance	
(Anticausal)	20% (0.138)	40% (0.257)	20% (0.257)	40% (0.12)	20% (0.099)	40% (0.089)
Forward	74.66±7.68	74.76±3.3	70.02±10.79	66.46±3.84	59.78±12.14	56.62±12.87
Reweighting	**84.58**±1.89	83.92±1.38	**83.30**±2.28	**73.22**±4.51	**85.02**±0.93	**83.3**±3.02
T-Revision	84.24±1.3	**85.70**±0.66	82.72±6.03	68.86±8.56	84.04±2.38	83.5±1.87
JoCoR (SSL)	83.44±0.83	60.28±1.46	80.64±1.29	57.14±4.17	63.84±8.8	54.56±4.44
MoPro (SSL)	76.62±7.16	76.37±7.0	79.55±2.32	58.44±7.11	77.36±4.04	65.14±5.61
Dividemix (SSL)	83.36±0.63	82.06±1.25	69.74±1.9	58.48±0.98	73.00±2.30	66.86±1.26
Mixup (SSL)	81.38±1.67	79.48±1.05	80.54±2.51	72.34±4.58	78.88±1.05	71.26±5.44

of $P(Y|X)$, but the information contained is limited, or 2). The information of $P(Y|X)$ contained in $P(X)$ is hard to be exploited by existing unsupervised methods.

In Tab. 6, for baselines, if the causal graph they generate contains any variables in the feature set X that cause Y, and Y does not cause any variables in the feature set, then the prediction is "causal". If Y is a cause to at least one variable in X, then the prediction is "anticausal". If there are no edges between any variables in X and Y, the prediction is considered None.

B.3. Compare with Existing Causal Discovery Methods

To the best of our knowledge, our method discovers causal and anticausal relations when the dataset contains noisy labels, there are no "natural" baselines for the CDNL. For completeness, we would like to add two existing causal discovery methods which are not designed to handle the noisy data for comparison. Specifically, PC (Spirtes et al., 2000) is a famous score-based approach for causal discovery, which is based on conditional tests on variables and sets of variables. GIES (Hauser & Bühlmann, 2012) is a score-based Bayesian algorithm that heuristically searches the graph which minimizes a likelihood score on the data.

As mentioned in our paper, on all anticausal datasets except Waveform, the difference $d(\hat{P}(\tilde{Y}|Y^*)|\hat{P}(\tilde{Y}|Y'))$ obtained by employing CDNL estimator is lower than 0.05. Here, we set the threshold of CDNL estimator to 0.05, i.e., if $d(\hat{P}(\tilde{Y}|Y^*)|\hat{P}(\tilde{Y}|Y'))$ smaller than or equal to 0.05, then it is anticausal dataset; if $d(\hat{P}(\tilde{Y}|Y^*)|\hat{P}(\tilde{Y}|Y'))$ greater than 0.05, then it is causal dataset. The results on two causal datasets show that our method is more accurate and robust than the two baselines. Note that, to the best of our knowledge, existing methods can not be directly applied to MNIST and CIFAR10 datasets because there are too many (pixel-level) variables in an image (feature set).

Table 6: Detecting causal and anticausal relations with different causal discovery methods.

		Sym		Pair		Instance	
		0.20%	0.40%	0.20%	0.40%	0.20%	0.40%
Krkp (causal)	GIES	anticausal	anticausal	anticausal	anticausal	anticausal	**causal**
	PC	anticausal	anticausal	anticausal	anticausal	anticausal	anticausal
	CDNL	**causal**	**causal**	**causal**	**causal**	**causal**	**causal**
Splice (causal)	GIES	anticausal	anticausal	anticausal	anticausal	anticausal	anticausal
	PC	anticausal	none	anticausal	anticausal	anticausal	none
	CDNL	**causal**	**causal**	**causal**	**causal**	**causal**	**causal**
Balancescale (causal)	GIES	**causal**	**causal**	**causal**	anticausal	**causal**	**causal**
	PC	anticausal	none	anticausal	anticausal	anticausal	anticausal
	CDNL	**causal**	**causal**	**causal**	**causal**	**causal**	**causal**

C. Proof of Theorem 3.1

In this section, we will prove the theorem in our main paper.

Proof. Let $\tilde{f}(x) = \arg\max_i P(\tilde{Y} = i|X = x)$ output the noisy label of every instance x.

$$\begin{aligned} P(\tilde{Y} = i|Y^* = j) &= \mathbb{E}_{P(X|Y^*=j)}[\mathbb{1}_{\{\tilde{f}(X)=i\}}] \\ &= \int_x \mathbb{1}_{\{\tilde{f}(x)=i\}} P(X = x|Y^* = j)\mathrm{d}x \\ &= \int_x \mathbb{1}_{\{\tilde{f}(x)=i\}} \frac{P(Y^* = j|X = x)P(X = x)}{P(Y^* = j)}\mathrm{d}x \\ &= \mathbb{E}_{P(X)}\left[\mathbb{1}_{\{\tilde{f}(X)=i\}} \frac{P(Y^* = j|X)}{P(Y^* = j)}\right]. \end{aligned} \tag{6}$$

Then similarly,

$$\begin{aligned} P(\tilde{Y} = i|Y' = j) &= \mathbb{E}_{P(X|Y'=j)}\left[\mathbb{1}_{\{\tilde{f}(X)=i\}}\right] \\ &= \int_x \mathbb{1}_{\{\tilde{f}(x)=i\}} P(X = x|Y' = j)\mathrm{d}x \\ &= \int_x \mathbb{1}_{\{\tilde{f}(x)=i\}} \frac{P(Y' = j|X = x)P(X = x)}{P(Y' = j)}\mathrm{d}x \\ &= \mathbb{E}_{P(X)}\left[\mathbb{1}_{\{\tilde{f}(X)=i\}} \frac{P(Y' = j|X)}{P(Y' = j)}\right]. \end{aligned} \tag{7}$$

The last equality is obtained by using the reweighting technique (Liu & Tao, 2016), which requires that $P(X|Y^* = j)$ and $P(X|Y' = j)$ have the same support. Then we calculate the difference $P(\tilde{Y} = i|Y' = j) - P(\tilde{Y} = i|Y^* = j)$ as follows.

$$\begin{aligned} & P(\tilde{Y} = i|Y' = j) - P(\tilde{Y} = i|Y^* = j) \\ =& \mathbb{E}_{P(X)}\left[\mathbb{1}_{\{\tilde{f}(X)=i\}} \frac{P(Y' = j|X)}{P(Y' = j)}\right] - \mathbb{E}_{P(X)}\left[\mathbb{1}_{\{\tilde{f}(X)=i\}} \frac{P(Y^* = j|X)}{P(Y^* = j)}\right] \\ =& \mathbb{E}_{P(X)}\left[\mathbb{1}_{\{\tilde{f}(X)=i\}} \left(\frac{P(Y' = j|X)}{P(Y' = j)} - \frac{P(Y^* = j|X)}{P(Y^* = j)}\right)\right] \\ =& \mathbb{E}_{P(X)}\left[\mathbb{1}_{\{\tilde{f}(X)=i\}} \frac{P(Y' = j|X)P(Y^* = j) - P(Y^* = j|X)P(Y' = j)}{P(Y' = j)P(Y^* = j)}\right] \\ =& \frac{1}{P(Y^* = j)}\mathbb{E}_{P(X)}\left[\mathbb{1}_{\{\tilde{f}(X)=i\}} \frac{P(Y' = j|X)P(Y^* = j) - P(Y^* = j|X)P(Y' = j)}{P(Y' = j)}\right] \\ =& \frac{1}{P(Y^* = j)}\mathbb{E}_{P(X)}\left[\mathbb{1}_{\{\tilde{f}(X)=i\}} \left(P(Y' = j|X)\frac{P(Y^* = j)}{P(Y' = j)} - P(Y^* = j|X)\right)\right] \end{aligned} \tag{8}$$

By using the above equation, the estimation error $d(P(\tilde{Y}|Y'), P(\tilde{Y}|Y^*))$ is as follows.

$$\begin{aligned} d(P(\tilde{Y}|Y'), P(\tilde{Y}|Y^*)) =& \sum_i^L \sum_j^L \frac{|P(\tilde{Y} = i|Y' = j) - P(\tilde{Y} = i|Y^* = j)|}{L^2} \\ =& \frac{1}{L^2}\sum_i^L \sum_j^L \left| \frac{1}{P(Y^* = j)}\mathbb{E}_{P(X)}\left[\mathbb{1}_{\{\tilde{f}(X)=i\}} \left(P(Y' = j|X)\frac{P(Y^* = j)}{P(Y' = j)} - P(Y^* = j|X)\right)\right]\right| \end{aligned}$$

which completes the proof. □