THE UNIVERSITY OF SYDNEY

Doctoral Thesis

On the Importance of Transition Matrix for Learning with Noisy Labels

Author:
Yu Yao

Supervisor:
Sen. Lec. Tongliang Liu

Co-Supervisor:
Prof. Dacheng Tao

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

in the

TML Machine Learning Group
School of Computer Science
Faculty of Engineering

August 24, 2023

Abstract of thesis entitled

On the Importance of Transition Matrix for Learning with Noisy Labels

Submitted by

Yu Yao

for the degree of Doctor of Philosophy

at The University of Sydney

in August, 2023

Deep neural networks can achieve remarkable performance when large annotated training datasets are available. However, annotating a large number of examples accurately is often expensive and sometimes infeasible in real life. Some cheap datasets with poor annotation quality and containing noisy labels are widely used to train deep learning models. Recent results show that deep neural networks can easily memorize noisy labels during training, which leads to poor generalization ability.

To improve the generalization ability of deep learning models when learning with noisy labels, a *noise transition matrix* $T(\boldsymbol{x})$ has been widely employed by existing methods to reveal the transition relationship from clean labels to noisy labels of instances. It acts as a central building block in designing *statistical-consistent* methods for learning with noisy labels (T-based methods). However, for real-world machine learning datasets, the transition matrix is usually unknown and needs to be estimated. Accurately estimating the transition matrix can be a challenging task. This motivates recent work to design label-noise robust methods focusing on incorporating heuristics instead of requiring estimating the transition matrix (heuristic-based methods).

The heuristic-based method has demonstrated state-of-the-art (SOTA) performance on many benchmark datasets. These methods seem to be more practical than the methods employing the transition matrix. It raises the

question that is the transition matrix still important for learning with noisy labels. In this thesis, we answer that the transition matrix still plays an important role in learning with noisy labels. We will show that the transition matrix not only can be used to design statistical-consistent methods but also can help boost the performance of heuristic-based methods. Specifically, by employing the transition matrix $T(\boldsymbol{x})$, *confident examples* can be accurately selected. Then the selected confident examples can be leveraged in the training process of heuristic-based methods to boost their performance. We will also show that given the transition matrix, the performance of T-based methods will not be influenced by different data generative processes. By contrast, the performance of SOTA *heuristic-based methods* can be influenced by different data generative processes. It implies that the transition matrix can be employed to improve the robustness of learning models for a wide range of datasets.

Since the label-noise transition matrix is important but hard to estimate, we will propose two new transition-matrix estimation methods that reduce the estimation error of the transition matrix. The first method can effectively estimate *instance-independent* transition matrix by exploiting the divide-and-conquer paradigm. The second method focuses on estimating *instance-dependent* transition matrices by leveraging a *structural causal model*.

To improve the generalization ability of deep learning models when learning with noisy labels, noise transition matrix T(x) has been widely employed by existing methods to reveal the transition relationship from clean labels to noisy labels of instances. It acts as a major building block in designing statistical-consistent methods for learning with noisy labels (T-based methods). However, for real-world datasets, the transition matrix is usually unknown and needs to be estimated. Accurately estimating the transition matrix can be a challenging task. This motivates recent work to design label-noise robust methods focusing on incorporating heuristics instead of requiring estimating the transition matrix (heuristic-based methods). The heuristic-based method has demonstrated state-of-the-art (SOTA) performance on many benchmark datasets. These methods seem to be more practical than T-based methods. It raises the question that is

the transition matrix still important for learning with noisy labels. In this thesis, we answer that the transition matrix still plays an important role in learning with noisy labels. We will show that the transition matrix not only can be used to design statistical-consistent methods but also can help boost the performance of heuristic-based methods. We will also show that given the transition matrix, the performance of T-based methods will not be influenced by different data generative processes. By contrast, the performance of SOTA heuristic-based methods can be influenced by different data generative processes. Since the label-noise transition matrix is important but hard to estimate, we will propose two new transition-matrix estimation methods that reduce the estimation error of the transition matrix. The first method can effectively estimate instance-independent transition matrix by exploiting the divide-and-conquer paradigm. The second method focuses on estimating instance-dependent transition matrices by leveraging a structural causal model.

On the Importance of Transition Matrix for Learning with Noisy Labels

by

Yu Yao
B.E. *University of New South Wales*

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Doctor of Philosophy

at

University of Sydney
August, 2023

Declaration

I, Yu YAO, declare that this thesis titled, "On the Importance of Transition Matrix for Learning with Noisy Labels", which is submitted in fulfillment of the requirements for the Degree of Doctor of Philosophy, represents my own work except where due acknowledgement have been made. I further declared that it has not been previously included in a thesis, dissertation, or report submitted to this University or to any other institution for a degree, diploma or other qualifications.

Acknowledgements

First, I would like to thank my supervisor Dr. Tongliang Liu for all the support on my research. With his patient guidance, I learned many useful research techniques during my candidature period, which helps me to become an independent researcher.

I would also sincerely thank Prof. Dacheng Tao and all members of Trustworthy Machine Learning Group for their useful discussion during my Ph.D. study period. Their valuable advice helped me better understand my research area and solve many difficult problems in this area.

Last but not least, I would like to sincerely thank my family and my friends. Their supports encourage me to face all the difficulties encountered during my candidature period.

Yu Yao
University of Sydney
August 24, 2023

Contents

Chapter 1

Introduction

While deep learning has achieved remarkable success in various tasks, it often heavily relies on large-scale human-annotated data. Due to the expensiveness of accurately annotating large datasets, alternative and inexpensive annotating methods have been widely used, e.g., querying search engines with a keyword [18, 75] and harvesting social media images [56], etc. However, as a trade-off, these alternative methods have sacrificed the accuracy of annotations for the scale of the dataset. As it has been shown that deep neural networks can easily memorize noisy labels which leads to degenerated classification performance [101], how to robustly learn with noisy labels has attracted a lot of attention in recent years [61, 54, 52, 51].

To improve the generalization ability of deep learning models when learning with noisy labels, *noise transition matrix* T has been widely employed by existing methods to reveal the transition relationship from clean labels to noisy labels of instances. Specifically, $T_{ij}(\boldsymbol{x}) = P(\tilde{Y} = j|Y = i, X = \boldsymbol{x})$, where $P(A)$ denotes the probability of the event A, X denotes the random variable of instances/features, $\tilde{Y}$ is the variable for the noisy label, and Y is the variable for the clean label. The transition matrix T acts as a central building block in designing *statistical-consistent* methods for learning with noisy labels (T-based methods). The basic idea is that, by employing T, the clean class posterior can be inferred by giving the transition matrix and the noisy class posterior.

However, for real-world machine learning datasets, the transition matrix is usually unknown. Given only noise data, generally, accurately estimating the transition matrix is a challenging task [46, 98, 97]. To avoid

estimating the transition matrix, *heuristic-based methods* have been proposed. These methods usually try to purify the noisy training dataset by incorporating prior knowledge to select confident examples and correct incorrect labels.

Heuristic-based methods seem more practical than the T-based methods because they have achieved state-of-the-art performance on different datasets [46, 47], which does not rely on the transition matrix $T(\boldsymbol{x})$. It raises the question that whether the transition matrix is $T(\boldsymbol{x})$ still important for learning with noisy labels. In this thesis, to answer this question, we analyze the usefulness of $T(\boldsymbol{x})$ from different perspectives, which shows that the role of $T(\boldsymbol{x})$ can not be replaced in learning with noisy labels. For instance, the transition matrix can be leveraged to boost the performance of the heuristic-based methods. It can be also employed to improve the robustness of learning models for a wide range of datasets with different data generative processes. Since the label-noise transition matrix is important but hard to estimate, we will propose two new transition-matrix estimation methods that can effectively estimate the transition matrix. The rest major chapters in this thesis are summarized as follows:

Chapter 2. Preliminaries. In this chapter, we will formulate the problem settings of learning with noisy labels, define the label-noise transition matrix and briefly introduce some T-based methods and heuristic-based methods.

Chapter 3. Importance of the transition Matrix in Sample Selection. In this chapter, we show that the transition matrix can be leveraged to boost the performance of the heuristic-based methods. More specifically, We have shown that the selected confident examples by existing sample-selection methods could be class imbalanced. To improve the quality of confident examples, the transition matrix should be employed during the sample selection. Motivated by this, we have proposed a new sample-selection method based on $T(\boldsymbol{x})$. The proposed method can be leveraged to help train the heuristic-based method. Empirical results show that our method boosts the heuristic-based method on benchmark datasets under different types of label noise.

Chapter 4. Importance of the Transition Matrix via Lens of Causality. In this chapter, we show that T-based methods are more general than popular heuristic-based methods in learning with noisy labels. Specifically, from a casual perspective, we show that by exploiting the transition matrix, the performance of T-based methods are not sensitive to data generative processes, i.e., T-based methods work well for both cases that the feature X causes the latent clean label Y and Y causes X. By contrast, the SOTA heuristic-based methods that leverage semi-supervised learning are sensitive to data generative processes. These methods generally are not useful when X causes the clean label Y. To detect whether a specific noisy dataset follows the causal structure that whether X causes Y or Y causes X, we have also proposed an intuitive method by exploiting an asymmetric property of the two different causal structures regarding estimating the transition matrix.

- The contributions in this Chapter are included in:

> **Yu Yao**, Tongliang Liu, Bo Han, Mingming Gong, Jiankang Deng, Gang Niu, and Masashi Sugiyama. "Which is Better for Learning with Noisy Labels: The Semi-supervised Method or Modeling Label Noise?" *International Conference on Machine Learning*, 2023.

Chapter 5. Improving estimation of the Instance-Independent Transition Matrix by Divide-and-Conquer. In this chapter, we propose a new instance-independent transition matrix estimation method by exploiting the divide-and-conquer paradigm, which is called dual-T estimator. Intuitively, instead of directly estimating the original transition matrix proposed by previous methods, we found that the original transition matrix can be factorized into the product of two easy-to-estimate transition matrices. Motivated by this, our estimator estimates the two matrices separately, then multiplies them together to obtain the original transition matrix. Both theoretical analyses and empirical results illustrate the effectiveness of the dual-T estimator.

- The contributions in this Chapter are included in:

> **Yu Yao**, Tongliang Liu, Bo Han, Mingming Gong, Jiankang Deng, Gang Niu, and Masashi Sugiyama. "Dual t: Reducing estimation error for transition matrix in label-noise learning." *Advances in neural information processing systems*, 2020.

Chapter 6. Encouraging Identifiability of the Instance-Dependent Transition Matrix by Causality. In this chapter, by leveraging a structural causal model, we propose a novel generative approach for estimating instance-dependent transition matrices. In particular, we show that properly modeling the instances will contribute to the identifiability of the label noise transition matrix and thus lead to a better classifier. Empirically, our method outperforms state-of-the-art methods on both synthetic and real-world label-noise datasets.

> **Yu Yao**, Tongliang Liu, Mingming Gong, Bo Han, Gang Niu, and Kun Zhang. "Instance-dependent label-noise learning under a structural causal model." *Advances in neural information processing systems*, 2021.

Chapter 7. Conclusion. In this chapter, we conclude the contributions of this thesis.

Chapter 2

Preliminaries

In this chapter, we will introduce the problem settings of learning with noisy labels and define the transition matrix for label noise.

Problem setup. Let D be the distribution of a pair of random variables $(X, Y) \in \mathcal{X} \times \{1, \ldots, C\}$, where X denotes the variable of instances, Y the variable of labels, $\mathcal{X}$ the feature space, $\{1, \ldots, C\}$ the label space, and C the size of classes. In many real-world classification problems, examples independently drawn from D are unavailable. Before being observed, their clean labels are randomly flipped into noisy labels because of, e.g., contamination [77]. Let $\tilde{D}$ be the distribution of the noisy pair $(X, \tilde{Y})$, where $\tilde{Y}$ denotes the variable of noisy labels. In label-noise learning, we only have a sample set $\tilde{S} = \{(\boldsymbol{x}_i, \tilde{y}_i)\}_{i=1}^{n}$ independently drawn from $\tilde{D}$. The aim is to learn a robust classifier from the noisy sample $\tilde{S}$ that can assign clean labels for test instances.

The noise transition matrix. To build statistically consistent classifiers, which will converge to the optimal classifiers defined by using clean data, the transition matrix $T(\boldsymbol{x}) \in \mathbb{R}^{C \times C}$ plays an important role [59, 50, 71]. Specifically, the ij-th entry of the transition matrix, i.e., $T_{ij}(\boldsymbol{x}) = P(\tilde{Y} = j | Y = i, X = \boldsymbol{x})$, represents the probability that the instance $\boldsymbol{x}$ with the clean label $Y = i$ will have a noisy label $\tilde{Y} = j$. The transition matrix has been widely studied to build statistically consistent classifiers, because the clean class posterior $P(\boldsymbol{Y}|\boldsymbol{x}) = [P(Y = 1|X = \boldsymbol{x}), \ldots, P(Y = C|X = \boldsymbol{x})]^\top$ can be inferred by using the transition matrix and the noisy

class posterior $P(\tilde{\boldsymbol{Y}}|\boldsymbol{x}) = [P(\tilde{Y} = 1|X = \boldsymbol{x}), \ldots, P(\tilde{Y} = C|X = \boldsymbol{x})]^\top$, i.e., we have $P(\tilde{\boldsymbol{Y}}|\boldsymbol{x}) = T(\boldsymbol{x})P(\boldsymbol{Y}|\boldsymbol{x})$.

As the noisy class posterior can be estimated by exploiting the noisy training data, the key step remains how to effectively estimate the transition matrix. Given only noisy data, the transition matrix is unidentifiable without any knowledge of the clean label [92]. Specifically, the transition matrix can be decomposed to a product of two new transition matrices, i.e., $T(\boldsymbol{x}) = T'(\boldsymbol{x})A(\boldsymbol{x})$, and a different clean class posterior can be obtained by composing $P(\tilde{\boldsymbol{Y}}|\boldsymbol{x})$ with $A(\boldsymbol{x})$, i.e., $P'(\boldsymbol{Y}|\boldsymbol{x}) = A(\boldsymbol{x})P(\boldsymbol{Y}|\boldsymbol{x})$. Therefore, $P(\tilde{\boldsymbol{Y}}|\boldsymbol{x}) = T(\boldsymbol{x})P(\boldsymbol{Y}|\boldsymbol{x}) = T'(\boldsymbol{x})P'(\boldsymbol{Y}|\boldsymbol{x})$ are both valid decompositions. The current state-of-the-art methods [25, 24, 64, 63, 59] then studied a special case by assuming that the transition matrix is *class-dependent* and *instance-independent*, i.e., $T(\boldsymbol{x}) = T$. Note that there are specific settings [16, 53, 6] where noise is independent of instances. A series of assumptions [50, 76, 70] were further proposed to identify or efficiently estimate the transition matrix by only exploiting noisy data.

T-based methods. Statistically consistent methods are primarily developed based on the Transition Matrix [50, 64, 106]. In this thesis, we call these methods T-based methods. For example, Patrini et al. [64] leveraged a two-stage training procedure of first estimating the noise transition matrix and then using it to modify the loss to ensure risk consistency. These works rely on anchor points or instances belonging to a specific class with probability one or approximately one. When there are no anchor points, all the aforementioned methods cannot guarantee statistical consistency. Another approach is to jointly learn the noise transition matrix and classifier. For instance, on top of the softmax layer of the classification network [22], a constrained linear layer or a nonlinear softmax layer is added to model the noise transition matrix [86]. [105] propose an end-to-end method for estimating the transition matrix and learning a classifier. Specifically, a total variation regularization term is used to prevent the overconfidence problem of the neural network. [48] propose another end-to-end method based on *sufficiently scattered* assumption, which is by far the mildest assumption under which the transition matrix is identifiable.

Heuristic-based methods. In learning with noisy labels, there are some methods that do not require the transition matrix. These methods focus on employing heuristics to reduce the side-effect of noisy labels. For example, many methods use a specially designed strategy to select reliable samples [99, 25, 57, 72, 36, 5] or correct labels [55, 40, 88, 71]. Although those methods empirically work well, there is not any theoretical guarantee on the consistency of the learned classifier. Recently, some methods exploiting semi-supervised learning techniques have been proposed to solve the label-noise learning problem like SELF [61] and DivideMix [46]. These methods are aggregations of multiple techniques such as augmentations, sample selection and multiple networks. Noise robustness is significantly improved with these methods. Additionally, these methods are sensitive to the choice of hyperparameters.

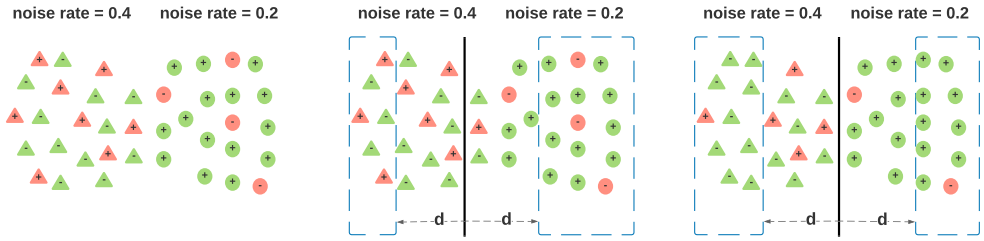
Chapter 3

Importance of the Transition Matrix in Sample Selection

In this chapter, we show that the transition matrix $T(\boldsymbol{x})$ can be employed to help sample selection and label correction. Specifically, we analyze the property of the sample-selection methods based on the small loss of noisy data from a theoretical point of view. By assuming that the transition matrix is independent of instance, i.e., $T(\boldsymbol{x}) = P(\tilde{Y} = i|Y = j, X = \boldsymbol{x}) = P(\tilde{Y} = i|Y = j) = T$, we show that the selected examples could be class imbalanced and inaccurate. To solve it, the transition matrix should be employed during the sample selection, which has been empirically validated on different datasets. We have also illustrated that the selected confident examples by employing the transition matrix can help train heuristic-based methods to boost their performance on different types of label noise.

3.1 Motivations and Contributions

To make neural networks robust to label noise, heuristics-based methods focus on designing heuristics for sample selection and label correction to reduce the side-effect of noisy labels. Most of these heuristics are designed based on the *memorization effect* of deep neural networks [4], i.e., they would memorize easy instances first, and gradually adapt to hard instances with the increasing amount of training. Inspired by this, many methods use the classification loss on noisy data as the measure of the cleanliness of examples [36, 25, 61, 46, 5], i.e., an example is likely to be clean if it has a small loss on noisy data. While these methods have shown promising results



(a) **(b)** **(c)**

Figure 3.1: Circles denote instances with clean positive labels, and triangles denote instances with clean negative labels. Different signs represent different noisy labels. Black lines denote decision boundaries. The example which is far away from the black line is more confident. The confident examples are in the blue dashed box. (a) A binary training dataset contains asymmetric label noise. (b) An illustration of confident examples selected by current sample-selection methods based on the small loss of noisy data. The instances (circles) in the class with a smaller noise rate are easier to learn based on the memorization effect. As a result, those instances are more confident and far away from the decision boundary. (c) An illustration of confident examples selected by our method, which are more robust to label noise. By exploiting the transition matrix, the estimated clean class posteriors can be employed to select and relabel confident examples.

when combined with different techniques such as warm-up [95], co-training [25], and mixup [46], they are not guaranteed to be statistically consistent and often need extensive hyperparameter tuning on clean data. Moreover, to achieve high classification accuracy on clean data, some methods need different regularization terms for different types of label noise [46, 61]. The T-based methods aim to design *classifier-consistent* algorithms, where classifiers learned by exploiting noisy data will asymptotically converge to the optimal classifiers defined on the clean domain [59, 50, 64]. However, they are not able to achieve satisfactory classification performance compared with the methods leveraging semi-supervised learning techniques [46, 61].

Currently, these two streams of methods are studied independently according to different philosophies. Sample selection and label correction methods exploited the memorization effect which is a property of the neural network, while loss correction methods focused on the transition between the noisy and clean class distributions. A natural question that arises here is whether the transition matrix T can be leveraged to help to improve

heuristic-based methods. The answer is Yes. Intuitively, the first stream of methods employs the classification loss on noisy labels as a measure of cleanliness. However, this measure is entangled with the label noise of training data. For example, in Figure 3.1(a), we illustrate a training dataset that contains asymmetric label noise which is the most general type of instance-independent label noise [77]. Specifically, the noise rate is 0.2 for the clean positive class (circle) and 0.4 for the clean negative class (triangle). Under such circumstances, existing small loss-based methods could select more instances in the class with a lower noise rate as confident examples. Additionally, the labels of these examples may contain noise and can not be fully trusted. These phenomenons are shown in Figure 3.1(b). These issues could lead to the low generalization ability of a classifier trained with these confident examples [35, 1].

To solve these issues, we show that the noise transition matrix can be employed to help heuristics-based methods select accurate and class-balanced confident examples. Specifically, we train a model with the loss corrected by the transition matrix and use the confidence of the estimated clean class posterior as the selection measure instead of the classification loss with noisy labels. In such a way, the examples are selected solely based on the confidence of the estimated clean class posteriors while the noise is handled by the transition matrix. Therefore, with the help of the transition matrix, the quality of selected examples can be improved, which is illustrated in Figure 3.1(c).

3.2 Related Work

Let $P_{\hat{\theta}}(\tilde{Y}|X)$ denote the estimated noisy class posteriors parameterized by $\hat{\theta}$ learned from noisy training data. Typically, the objective of existing methods based on small-loss sample selection is formulated as follows [36, 25]:

$$\mathcal{L}(\hat{\theta}) = \frac{1}{n}\sum_{i=1}^{n} v_i \ell(P_{\hat{\theta}}(\tilde{Y}|\boldsymbol{x}_i), \tilde{y}_i) = \frac{1}{n}\sum_{i=1}^{n} -v_i \log(P_{\hat{\theta}}(\tilde{Y} = \tilde{y}_i|\boldsymbol{x}_i)),$$

where ℓ is the cross-entropy loss and $v_i \in [0, 1]$ is the per-instance weight. The idea is that if the given label $\tilde{y}_i$ from a training data pair $(\boldsymbol{x}_i, \tilde{y}_i)$ is likely to be clean, then v_i should be equal or close to 1, so that it contributes more than those data pairs whose labels are likely to be incorrect.

To find the weight v_i for each instance, based on the memorization effect, one popular criterion is using the classification loss on noisy data:

$$v_i = \mathbb{1}(\ell_i \leq \lambda) = \mathbb{1}(\ell(P_{\hat{\theta}}(\tilde{Y}|\boldsymbol{x}_i), \tilde{y}_i) \leq \lambda), \tag{3.1}$$

where $\mathbb{1}$ is the indicator function, ℓ_i is the loss for instance $\boldsymbol{x}_i$ and λ is the loss threshold. Specifically, if a data pair $(\boldsymbol{x}_i, \tilde{y}_i)$ has a loss smaller than the threshold λ, then it is treated as a "clean" data, and will be selected in training ($v_i^* = 1$) as a confident example. Otherwise, it will not be selected ($v_i^* = 0$). For example, Jiang et al. [36] used a mentor network to select confident examples. Han et al. [25] maintained two networks that select small-loss instances, where the loss threshold is continuously increased during training so that more instances are dropped when the number of epochs gets large. Except for selecting small-loss instances, some methods reweighted examples so that mislabeled samples contribute less to the loss, e.g., Ren et al. [72] reweighted instances according to their gradient directions. Arazo et al. [3] and Li, Socher, and Hoi [46] calculated per-instance weights by modeling the classification loss distribution with a mixture model. Recently, some methods exploiting semi-supervised learning techniques have been proposed to solve the label-noise learning problem like SELF [61] and DivideMix [46]. These methods are aggregations of multiple techniques such as augmentations, sample selection and multiple networks. Noise robustness is significantly improved with these methods. Additionally, these methods are sensitive to the choice of hyperparameters.

3.3 Methodology

3.3.1 Importance of the Transition Matrix in Sample Selection

The real-world label-noise datasets usually contain asymmetric noise which means that different classes could have different noise rates. Intuitively, the reason is that the examples from similar classes are usually much easier to be incorrectly labeled compared with the examples from dissimilar classes. For real-world applications, asymmetric noise is also general. For example, the symmetric noise is a special case of the asymmetric noise when: 1). all diagonal entries of its transition matrix have the same value; 2). all off-diagonal entries also have the same value. This motivates us to investigate the issues of existing confident examples methods on asymmetric noise.

Existing methods based on the small loss mainly rely on the memorization effect of the deep neural network to select samples. We show that, in general, confident examples selected with the small loss on noisy data can be class imbalanced and inaccurate while the noise is asymmetric. This is because, based on the memorization effect, empirically, the instances from a class with a small noise rate tend to be learned "faster" and have smaller losses than examples from a class with a relatively large noise rate. Thus, instances from the class with a small noise rate or low complexity will be too frequently selected and examples from the class with a relatively large noise rate will not be learned well.

We further show that, theoretically, even an optimal hypothesis f^* which perfectly learns the noisy class posterior distribution can be obtained, the small-loss selection criteria still have the bias issue mentioned above. Let loss function ℓ be the widely used cross-entropy loss. Intuitively, the examples with smaller losses are those which have higher confidence in noisy class posteriors [58]. Furthermore, the examples from a class with a lower noise rate are expected to have higher confidence than examples from other classes. Therefore, the examples in the class with a lower noise rate are more likely to be selected as confident examples than other classes, i.e., the selected examples could be class-imbalanced. Moreover, the selected confident examples should not be treated as "clean" data, because the noisy

labels can be different from *Bayes labels*[1] on the clean class-posterior distribution. As a result, the classification accuracy can degenerate if a model is directly trained with those selected examples. We analyze these problems in Theorem 3.3.1 and Theorem 3.3.2. We leave all proofs in Appendix A.

Theorem 3.3.1. *Let* $\boldsymbol{x}_1$, $\boldsymbol{x}_2$ *be two instances such that* $\arg\max_{i\in\{0,1\}} P(Y = i|\boldsymbol{x}_1) = \arg\max_{j\in\{0,1\}} P(\tilde{Y} = j|\boldsymbol{x}_1) = 1$, $\arg\max_{i\in\{0,1\}} P(Y = i|\boldsymbol{x}_2) = \arg\max_{j\in\{0,1\}} P(\tilde{Y} = j|\boldsymbol{x}_2) = 0$, *and* $P(Y = 0|\boldsymbol{x}_2) = P(Y = 1|\boldsymbol{x}_1)$. *If* $P(\tilde{Y} = 1|Y = 0) - P(\tilde{Y} = 0|Y = 1) > 0$, *then* $\min_{i\in\{0,1\}} \ell(f^*(\boldsymbol{x}_2), i) > \min_{i\in\{0,1\}} \ell(f^*(\boldsymbol{x}_1), i)$.

Intuitively, the above theorem shows that, under asymmetric noise, the instances with the same confidence on underlying clean class posteriors do not have the same loss defined on noisy class posteriors. The instance $\boldsymbol{x}_1$ from the class with a lower noise rate $P(\tilde{Y} = 0|Y = 1)$ could have a smaller loss than the instance $\boldsymbol{x}_2$ from the other class with higher noise rate $P(\tilde{Y} = 1|Y = 0)$.

Because the instances in the class with a lower noise rate are more likely to be selected as confident examples. This generally will cause a class-imbalanced issue. The reason is that the existence of the class imbalance issue depends on the clean class prior, noise ratio, and sparsity of class-conditional densities. When the noise type is asymmetric, the clean class prior and sparsity of class-conditional densities have to cancel the contribution of the asymmetric noise to the class-imbalance issue. Empirically, we show that the class imbalance issue exists in the state-of-the-art method in Appendix B.

Theorem 3.3.2. *When* $P(\tilde{Y} = 1|Y = 0) - P(\tilde{Y} = 0|Y = 1) > 0$, *if an instance* x_1 *such that* $0.5 < P(Y = 0|\boldsymbol{x}_1) < \frac{(1-2P(\tilde{Y}=0|Y=1))}{(1-2P(\tilde{Y}=1|Y=0))} P(Y = 1|\boldsymbol{x}_1)$, *then* $P(\tilde{Y} = 1|\boldsymbol{x}_1) > 0.5$.

Theorem 3.3.2 shows that the largest clean and noisy class posteriors of an instance may not be identical if the noise is asymmetric. Under such circumstances, the training examples could have different Bayes labels on

[1]The Bayes label is the label with the largest class posterior. For example, the Bayes label on the clean class-posterior distribution Y^* of an instance $\boldsymbol{x}$ is defined as $Y^* = \arg\max_{i\in\{0,1\}} P(Y = i|x)$ [58].

the clean and noisy class posteriors, respectively. As a result, the confident examples selected by using the small-loss criterion could be inaccurate, because the examples have been treated as "clean" data directly [36, 25].

It is worth mentioning that the results in Theorem 3.3.1 and Theorem 3.3.2 can also be extended to a multi-class classification problem (which can be reduced to a set of binary classification problems). For example, to formulate a multi-class classification problem with binary classification problems, the one-versus-one decomposition strategy [2] can be employed. Specifically, the one-versus-one strategy builds a binary classifier for each pair of classes. To predict the label of a test example, we first put all available classes into a candidate label set. Then, let each classifier gives a predicted label of this example, and the other label will be removed from the candidate label set. This process is repeated until only one class is left in the label set, and it will be the final predicted label. It is worth mentioning that if the multi-class dataset contains asymmetric noise, then the set of binary datasets by the one-versus-one decomposition strategy can also contain asymmetric noise. As a result, classifiers learned on these binary datasets will be problematic.

3.3.2 Sample selection and Label Correction with the Transition Matrix

Loss correction. Our method selects confident examples based on the estimated clean class posterior which can be obtained by exploiting the noisy posterior and the transition matrix. Let $P_\theta(\tilde{Y}|X)$ be the noisy class posterior parameterized by θ, and $P_\phi(Y|X)$ be the clean class posterior parameterized by ϕ. We first learn ϕ with the loss corrected by the transition matrix $\boldsymbol{T}$:

$$\mathcal{L}(\phi) = \frac{1}{n}\sum_{i=1}^{n} \ell(\boldsymbol{T}P_\phi(Y|X=\boldsymbol{x}_i), \tilde{y}_i). \tag{3.2}$$

where ℓ is the cross-entropy loss and the transition matrix $\boldsymbol{T}$ can be estimated beforehand [64, 92, 98] or jointly learned with the network [22, 48].

Sample selection and label correction. After training, the estimated clean class posterior of an instance $\boldsymbol{x}_i$ can be calculated by $P_{\hat{\phi}}(Y|X = \boldsymbol{x}_i)$. Then, instead of using the classification loss $\ell(P_{\hat{\theta}}(\tilde{Y}|X = \boldsymbol{x}_i), \tilde{y}_i)$ by exploiting noisy data to select instances, we use $H(P_{\hat{\phi}}(Y|X = \boldsymbol{x}_i))$ as the selection measure where $H(\cdot)$ denote a function for measuring the confidence on the clean class posterior, i.e., we select an instance if we are confident that the estimated clean class posterior of the instance is correct. The problem remains how to design an appropriate measure of confidence. For classification problems, it is obvious that easy examples are ones whose correct labels can be predicted easily (they lie far from the decision boundary or they are close to anchor points). To this end, we use the entropy of the estimated clean class posterior as the confidence measure and our selection criterion can be formulated as follows:

$$v_i = \mathbb{1}(H(P_{\hat{\phi}}(Y|X = \boldsymbol{x}_i)) \leq \beta), \forall i \in [1, n]. \tag{3.3}$$

where $H(\cdot)$ is the entropy function and β is the selection threshold. Intuitively, an instance whose estimated clean class posterior has entropy smaller than the threshold β will be selected ($v_i = 1$). Otherwise, it will not be selected ($v_i = 0$).

With the proposed criterion, we divide the training data into a labeled set and an unlabeled set. However, since the network is trained with the corrected loss, confident prediction of an instance does not necessarily mean that the label of the instance is clean. Thus, we re-label those selected instances as follows:

$$\hat{y}_i = \arg\max_c P_{\hat{\phi}}(Y = c|X = \boldsymbol{x}_i). \tag{3.4}$$

3.3.3 Implementation

Empirically, the clean class-posterior distribution $P_\phi(Y|X)$ can be modeled by a mapping (e.g., neural network) $g_\phi : \mathcal{X} \to \Delta^{C-1}$, where Δ^{C-1} denotes a probability simplex. Given the transition matrix, the model parameter ϕ

can be directly estimated from noisy data as follows:

$$\hat{\phi} = \arg\min_{\phi} \frac{1}{n} \sum_{i=1}^{n} \ell(\boldsymbol{T} g_{\phi}(\boldsymbol{x}_i), \tilde{y}_i). \tag{3.5}$$

However, the transition matrix $\boldsymbol{T}$ can be unknown and needed to be estimated. In experiments, we assume the transition matrix $\boldsymbol{T}$ is not given, and the state-of-the-art method VolMinNet [48] is used to estimate $\boldsymbol{T}$. The reasons we use this method are that 1) it is general and can identify the transition matrix under the mildest assumption by far; 2) it is a computationally efficient method that allows us to learn the transition matrix and the noisy class posterior simultaneously. After having the estimated transition matrix $\hat{\boldsymbol{T}}$ and model parameter $\hat{\phi}$, we could re-label the training data to get a confident labeled set S_l as follows:

$$S_{\mathrm{l}} = \{(\boldsymbol{x}_i, \hat{y}_i) | H(g_{\hat{\phi}}(\mathbf{x}_i)) \leq \beta \ , \boldsymbol{x}_i \in S\}. \tag{3.6}$$

In Section 3.4, we show that our method significantly improves the quality of selected examples, and therefore, the classification accuracy of existing label-noise learning methods based on sample selection can also be improved by employing our method.

3.4 Experiments

In this section, we demonstrate the performance of the proposed method on multiple bench-mark datasets under various types of noise.

Datasets. We verify the effectiveness of our approach on the manually corrupted version of two datasets, i.e., *CIFAR10*, *CIFAR100* [41], and one real-world noisy dataset, i.e., *Clothing1M* [94]. *CIFAR10* contains 50,000 training images and 10,000 test images. *CIFAR10* and *CIFAR100* both contain 50,000 training images and 10,000 test images but the former has 10 classes of images, and the latter has 10 classes of images. The two datasets contain clean data, and different types of instance-independent label noise are manually added to the training sets. *Clothing1M* has 1M images with real-world noisy labels and 10k images with clean labels for

	CIFAR-10		CIFAR-100	
	Sym-20%	Sym-50%	Sym-20%	Sym-50%
Decoupling	77.32 ± 0.35	54.07 ± 0.46	41.92 ± 0.49	22.63 ± 0.44
MentorNet	81.35 ± 0.23	73.47 ± 0.15	42.88 ± 0.41	32.66 ± 0.40
Co-teaching	82.27 ± 0.07	75.55 ± 0.07	48.48 ± 0.66	36.77 ± 0.52
Forward	85.20 ± 0.80	74.82 ± 0.78	54.90 ± 0.74	41.85 ± 0.71
T-Revision	87.95 ± 0.36	80.01 ± 0.62	62.72 ± 0.69	49.12 ± 0.22
DMI	87.54 ± 0.20	82.68 ± 0.21	62.65 ± 0.39	52.42 ± 0.64
VolMinNet	$89.58 \pm \pm 0.26$	83.37 ± 0.25	64.94 ± 0.40	53.89 ± 1.26
DivideMix	95.13 ± 0.081	94.59 ± 0.33	74.72 ± 0.25	70.74 ± 0.36
T-SSLC-DM	$\mathbf{95.51 \pm 0.11}$	$\mathbf{94.97 \pm 0.29}$	$\mathbf{75.46 \pm 0.31}$	$\mathbf{72.92 \pm 0.42}$
	CIFAR-10		CIFAR-100	
	Pair-20%	Pair-45%	Pair-20%	Pair-45%
Decoupling	77.12 ± 0.30	53.71 ± 0.99	40.12 ± 0.26	27.97 ± 0.12
MentorNet	77.42 ± 0.00	61.03 ± 0.20	39.22 ± 0.47	26.48 ± 0.37
Co-teaching	80.65 ± 0.20	73.02 ± 0.23	42.79 ± 0.79	27.97 ± 0.20
Forward	88.21 ± 0.48	77.44 ± 6.89	56.12 ± 0.54	36.88 ± 2.32
T-Revision	90.33 ± 0.52	78.94 ± 2.58	64.33 ± 0.49	41.55 ± 0.95
DMI	89.89 ± 0.45	73.15 ± 7.31	59.56 ± 0.73	38.17 ± 2.02
VolMinNet	90.37 ± 0.30	88.54 ± 0.21	68.45 ± 0.69	58.90 ± 0.89
DivideMix	95.72 ± 0.04	87.02 ± 0.41	75.54 ± 0.43	45.20 ± 0.16
T-SSLC-DM	$\mathbf{95.80 \pm 0.05}$	$\mathbf{95.01 \pm 0.01}$	$\mathbf{76.68 \pm 0.25}$	$\mathbf{63.50 \pm 0.19}$

Table 3.1: Classification accuracy (percentage) on CIFAR-10 and CIFAR-100.

testing. It also has an additional 50k clean training data and 14k clean validation data. Note that we only exploit the 1M data for the training and validate our model on the 14k clean validation data. For all the synthetic noisy datasets, the experiments are repeated 5 times.

Noise Types. Following prior works [61, 48], we conduct experiments with two commonly used types of noise: (1) symmetry flipping [64] which randomly replaces a percentage of labels in the training data with all possible labels. (2) pair flipping [25] which is a specific type of asymmetric noise, where labels are only replaced by similar classes. It is worth mentioning that the noise rate is calculated differently compared with the original paper of DivideMix [46] because the noise generative process is different. We use the same noise generative process proposed by [25]. As a result, for example, pair flipping with 45% noise (pair-45%) in our paper is equivalent to asymmetric noise 50% (Asym-50%) in the paper of DivideMix [46]. We have also designed a new type of asymmetric noise that only constrains the

	CIFAR-10		CIFAR-100	
	Asym-50%	Asym-70%	Asym-50%	Asym-70%
DivideMix	95.13 ± 0.23	68.12 ± 1.57	64.37 ± 2.63	35.47 ± 3.53
T-SSLC-DM	$\mathbf{95.51 \pm 0.31}$	$\mathbf{72.12 \pm 1.95}$	$\mathbf{75.46 \pm 3.06}$	$\mathbf{49.36 \pm 3.41}$

Table 3.2: Classification accuracy (percentage) on CIFAR-10 and CIFAR-100.

Decoupling	MentorNet	Co-teaching	Forward	Joint-Optim
54.53	56.79	60.15	71.79	72.16
DMI	VolMinNet	DivideMix	T-SSLC-DM	
72.46	72.62	74.48	**74.92**	

Table 3.3: Classification accuracy on Clothing1M.

transition matrix to be diagonally dominant. Generating such a transition matrix is complicated. Because each column of the transition matrix has to be a probability simplex with bounded elements. Sampling such a column is an over-constrained problem and needs to be re-sampled until the constraints are satisfied.

Here, we illustrate the new type of asymmetric noise that is used in our experiments. This asymmetric noise is general, We only require the transition matrix to be diagonally dominant. Generating such a transition matrix is complicated. Because each column of the transition matrix has to be a probability simplex with bounded elements. Sampling such a column is an over-constrained problem and needs to be re-sampled until the constraints are satisfied. In Algorithm 1, we illustrate our generation method which can efficiently generate such type of noise.

Network structure and optimization. For a fair comparison, we implement all methods with default parameters by PyTorch on Nvidia Geforce RTX 3090 GPUs. We use a PreResNet-18 network and PreResNet-32 network for CIFAR10 and CIFAR100, respectively. We use SGD to train the classification network with batch size 128, momentum 0.9, weight decay 10^{-3} and an initial learning rate 10^{-2}, the learning rate is divided by 10 after 40 epochs. The algorithm is run for 80 epochs for the sample selection and relabeling. For clothing1M, we use a ResNet-50 pre-trained on

Algorithm 1 Generating Asymmetric Transition Matrix

Input: Noise Rate σ, Dimension C
Initialize $\boldsymbol{T} \in [0,1]^{C \times C}$
$valid \leftarrow False$
While not valid:
  For $j \leftarrow 1$ to C
    $\boldsymbol{T}_{j,j} \sim \mathcal{U}(0,1)$
$Sum \leftarrow \Sigma_{j=1}^{C} \boldsymbol{T}_{j,j}$
  For $j \leftarrow 1$ to C
    $\boldsymbol{T_{j,j}} \leftarrow \frac{\boldsymbol{T}_{i,i}}{Sum}\sigma$
  For $j \leftarrow 1$ to C
    For $i \leftarrow 1$ to C
      If $i \neq j$
        $\boldsymbol{T}_{i,j} \sim \mathcal{U}(0,1)$
    $Sum_j \leftarrow \Sigma_{i=1,i\neq j}^{C} \boldsymbol{T}_{i,j}$
    For $i \leftarrow 1$ to C
      If $i \neq j$
        $\boldsymbol{T}_{i,j} \leftarrow \frac{\boldsymbol{T}_{i,j}}{Sum_j}(1 - \boldsymbol{T}_{j,j})$
valid $\leftarrow True$
  For $j \leftarrow 1$ to C
    For $i \leftarrow 1$ to C
      If $i \neq j$ and $\boldsymbol{T}_{j,j} \leq \boldsymbol{T}_{i,j}$
        valid $\leftarrow False$

ImageNet. For each epoch, we also ensure the noisy labels for each class are balanced with undersampling.

Baselines. We compare our method with the following baselines: (i) Decoupling [57], which trains two networks on samples whose predictions from the two networks are different. (ii) MentorNet [36], Co-teaching [25], which mainly handles noisy labels by training on instances with small loss values. (iii) Forward [64], Reweight [50], and T-Revision [92]. These approaches utilize a class-dependent transition matrix T to correct the loss function. (iv) DivideMix [46] which aggregates multiple techniques such as augmentations, multiple networks, and example selection. For all baselines, we follow the settings from their original papers.

	CIFAR-100				
percentile	20%	30%	40%	50%	60%
sym-50%	89.25 ± 0.12	87.94 ± 0.09	86.10 ± 0.12	85.79 ± 0.14	79.23 ± 0.52
pair-45%	91.14 ± 0.20	90.63 ± 0.23	88.72 ± 0.36	88.40 ± 0.40	80.21 ± 0.64

Table 3.4: Experiments of the hyper-parameter selection on synthetic-noise dataset CIFAR-100.

3.4.1 Classification Accuracy Evaluation

Classification accuracy on synthetic noisy datasets. To investigate how the sample selection of T-SSLC will affect the classification accuracy in label-noise learning, we embed our sample-selection method T-SSLC into the state-of-the-art DividMix [46] called T-SSLC-DM. We report average accuracy over the last ten epochs of each model on the test set. Higher classification accuracy means that the algorithm is more robust to the label noise. In Table 3.1, we compare classification accuracies of T-SSLC-DM with DivideMix and other baseline methods on synthetic noisy datasets. T-SSLC-DM outperforms baseline methods in almost all settings of noise. This result is natural after we have shown that T-SSLC leads to a high clean ratio of selected examples. These results show the advantage of using the proposed T-SSLC.

In Table 3.2, we compare the classification accuracies of T-SSLC-DM with the state-of-the-art method DivideMix. The asymmetric noise is employed, i.e., labels can be randomly flipped to all other classes with different probabilities but noise rates for each class are the same. We report the average accuracy over the last ten epochs of each model on the test set. The results show that T-SSLC-DM outperforms DivideMix on asymmetric noise.

Classification accuracy on Clothing1M. We show the results on Clothing1M in Table 3.3 which should contain instance-dependent label noise. T-SSLC-DM outperforms previous transition-matrix based methods and heuristic methods on the Clothing1M dataset. The performance on Clothing1M dataset shows that the proposed method also has certain robustness against instance-dependent label noise.

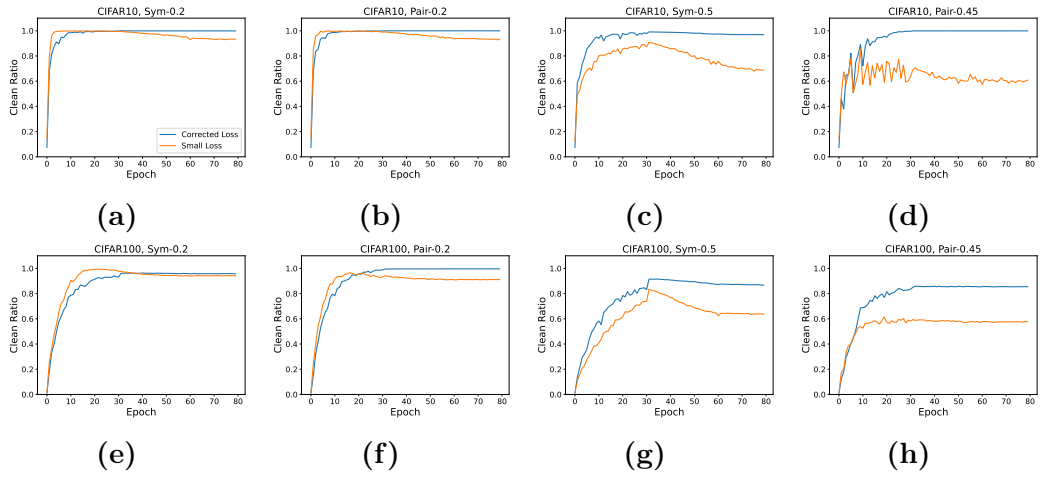


Figure 3.2: Sample selection on CIFAR-10 and CIFAR-100 with different settings of label noise. When the noise rate is small (sym-0.2 and pair-0.2) or symmetric (sym-0.2 and sym-0.5), both methods can effectively select clean labels. With the help of the transition matrix, the proposed method (blue) shows better robustness against asymmetric label noise and a high noise rate (pair-0.45 and sym-0.5) compared with the existing small-loss sample-selection method (orange).

3.4.2 Hyperparameter Selection

To select a suitable value of the hyper-parameter β, we carry out experiments on dataset CIFAR-100. Different types of noise are injected into the dataset. Table 3.4 illustrates the clean ratios of selected confident examples by varying β to be 20%, 30%, 40%, 50% and 60%-percentile in place of the training examples' entropy list (ordered from low values to high values). The results show that from 50%-percentile to 60%-percentile, clean ratios drop dramatically for both symmetric flipping 50% and pairwise flipping 45%. From 20%-percentile to 40%-percentile, clean ratios do not drop too much. Therefore, in other experiments, we let β be the entropy value of 50%-percentile in place of the training examples' entropy list.

3.4.3 Clean Ratio Comparison

To illustrate that our proposed method is more effective in selecting clean examples, we compare the clean ratio of the selected examples with the small-loss criteria. Specifically, we train a neural network for 200 epochs on CIFAR10 and CIFAR100 with different settings of label noise, at each

CIFAR10			
	asym-20%	asym-30%	asym-50%
DivideMix	0.012	0.043	0.052
T-SSLC-DM	**0.009**	**0.031**	**0.029**

Table 3.5: The degree of class imbalance of a state-of-the-art DivideMix and our method on CIFAR10 with asymmetric noise.

epoch, we use our proposed method and Co-teaching [25] using small-loss criteria to select 50% examples in the training dataset as confident examples and compare their clean ratio, i.e., the number of selected clean labels divided by the size of the set.

The results in Fig. 3.2 validate that our method is disentangled from the label errors. Specifically, for different noise rates and types of noise, our method has similar performance, i.e., clean ratios of the selected examples by using our method do not change a lot. However, clean ratios of the selected examples by the small loss-based method Co-teaching dramatically decrease with the increase of label noise. It is worth mentioning that because of the memorization effect, clean ratios of obtained by Co-teaching method is some times high at the beginning. Specifically, previous work shows that the neural network tends to learn the easy examples at the early stage and gradually fit the hard examples. When the noise rate is small, most examples are clean and easy to learn. These examples will be learned first at the early learning stage and the incorrect examples will be ignored, as a result, the small-loss sample selection is effective in the early stage under small noise rates.

3.4.4 Class Imbalance Ratio Comparison

In Table 3.5, we have measured the degree of imbalance due to sample selection with the KL divergence between the class distribution of confident example and the class-prior distribution (uniform) on CIFAR10 with asymmetric noise. The smaller value of KL divergence implies a smaller degree of class imbalance. The result shows that the sampled confident examples by employing our method are more balanced than the small loss-based method DivideMix, and the transition matrix can help the existing

small loss-based method on sample selection.

3.5 Summary

We have shown that the confident examples selected with the small classification loss on noisy data could be class-imbalanced and inaccurate. We show that the transition matrix can be applied to help confident examples selection. Specifically, we use the transition matrix to estimate the clean class-posterior distribution, then the estimated clean class posterior for each instance is used for sample selection and label correction. Empirical results on both synthetic and real-world noisy datasets show that our method significantly improves the quality of selected confident examples and the performance of downstream classification tasks.

Chapter 4

Importance of the Transition Matrix via Lens of Causality

In this chapter, we illustrate the advantage of methods based on the transition matrix $T(\boldsymbol{x})$ from a causal perspective. We show that the performance improvement of SOTA heuristic-based methods depends heavily on the data generative process while the method based on the noise transition matrix is independent of the generative process. This further explains the important role of the transition Matrix in learning with noisy labels. Considering that 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 have also 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.

4.1 Motivations and Contributions

Although T-based methods have statistical guarantees, recent heuristic methods that leverage semi-supervised learning (SSL) have achieved SOTA performance on different datasets, which seem more practical than T-based methods. It raises the question that *is heuristic-based methods more powerful than T-based methods and can always help learn clean labels?* This question is crucial for the label-noise learning problem. If the answer is affirmative, heuristic-based methods can be more useful, which can help learn Y on any noisy dataset. Then future research should mainly focus on designing heuristic-based methods rather than T-based methods. By

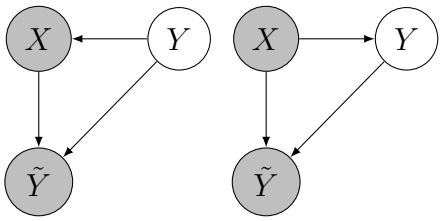
contrast, if the answer is negative, it is important to illustrate when the heuristic-based methods will not work, and how to detect the failure case in practice.

In this chapter, we seek answers to the aforementioned question by investigating the properties of both streams of methods from a causal perspective. We show that the performance of the SOTA method (heuristic SSL-based methods) depends on the underlying data generative process, but the performance of T-based methods is not influenced by the generative process. The reason is that heuristic SSL-based methods exploit the distribution of instances $P(X)$ by SSL to learn noise-robust representations, which sometimes is not feasible. Intuitively, to learn noise-robust representations from $P(X)$, it requires that the distribution $P(X)$ has to contain information about $P(Y|X)$. However, from a causal perspective, the amount of information of $P(Y|X)$ contained in $P(X)$ depends on the data generative process. More specifically, when the latent clean label Y is a cause of X, the distributions of $P(X)$ and $P(Y|X)$ are entangled [73, 104], then $P(X)$ will generally contain some information about $P(Y|X)$. Then exploiting label-dependent information contained in $P(X)$ encourages the identifiability of $P(Y|X)$. When the X is a cause of Y, the distributions of $P(X)$ and $P(Y|X)$ are disentangled, which means that label-dependent information contained in $P(X)$ is limited. Therefore, $P(X)$ can not help learn $P(Y|X)$, thus the benefits provided by SOTA heuristic-based methods exploiting the distribution of instances $P(X)$ are limited.

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.

4.2 Related Work

Heuristic methods based on semi-supervised learning. To help identify incorrect labels, SOTA heuristic methods aim to learn noise-robust representations by exploiting the data distribution $P(X)$ [46, 47, 90, 96,



(a) Y cause X **(b)** X cause Y

Figure 4.1: Different data generative processes of noisy data.

87, 12, 20, 96, 107]. The semi-supervised learning (SSL) techniques are usually employed by these methods. For example, the consistency regularization [42] is employed by [17]; FixMatch [82] is employed by [46]; the co-Regularization is employed by [90]; contrastive learning is employed by [87, 12, 47, 20, 96, 107]. Empirically, these methods have demonstrated great success on different datasets.

Causal generative 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) [84]. Specifically, in Fig. 4.1(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 [21]. For example, Fig. 4.1(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 generative process can also be described by a structural causal model (SCM). Specifically,

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

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 $\tilde{Y}$. f and g can be linear or non-linear functions. Each equation species a distribution of a variable conditioned on its parents (could be an empty set). Similarly, the SCM

corresponding to the causal graph in Fig. 4.1(b) can be written as:

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

Causal factorization and modularity . By the conditional independence relations proposed by the Markov property [65], 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 [73, 69], 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. Similarly, when X causes Y, the causal decomposition of $P(X, Y, \tilde{Y})$ is as follows:

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

Causal discovery methods. In order to build a graph that captures these conditional independencies, the majority of constraint-based techniques look for conditional independencies in the empirical joint distribution. Since numerous graphs frequently satisfy a given set of conditional dependencies, as was discussed above, constraint-based methods frequently produce a graph that represents some Markov equivalence classes. Unfortunately, large sample sizes are necessary for conditional independence tests to be reliable, and Shah and Peters [79] highlight further difficulties to control the Type I error.

Score-based approaches test the validity of a candidate graph $\mathcal{G}$ according to some scoring function S. The goal is therefore stated as [67]:

$$\hat{\mathcal{G}} = \text{argmax}_{\mathcal{G}} \text{ over } \mathbf{x} S(\mathcal{D}, \mathcal{G}) \tag{4.1}$$

where the empirical data for the variables $\mathbf{X}$ is represented by $\mathcal{D}$. Common scoring functions include the Bayesian Information Criterion (BIC) [19], the Minimum Description Length (as an approximation of Kolmogorov Complexity) [34, 23, 38], the Bayesian Gaussian equivalent (BGe) score [19], the Bayesian Dirichlet equivalence (BDe) score [27], the Bayesian Dirichlet equivalence uniform (BDeu) score [27], and others [30, 29, 28].

Methods based on causal function provide an alternate strategy for estimating causal effects. Assumptions about the data generative process are used in these causal function-based techniques. The causal function-based approach fits the causal function model among variables and then infers causal directions using causal assumptions, such as a non-Gaussian assumption of the noise [80, 81] the independence assumption between cause variables and noise [103, 68, 66] and the independence assumption between the distribution of cause variables and the causal function [33]. Most LiNGAM-based approaches for the linear case Shimizu et al. [80] assume non-Gaussian noise and linear causal relations between variables. This model seeks to determine a causal order among the random observed variables.

To deal with linear latent confounders, an estimation method utilizing overcomplete ICA [45] is suggested. However, overcomplete ICA algorithms usually suffer from local optimum and cannot be employed when the number of variables is large. By evaluating the independence between the estimated exogenous variables and the residual, Tashiro et al. [89] identify latent confounders. They discover that variables from subsets that are not impacted by latent confounders are included, and they estimate causal orders one at a time. Chen and Chan [10] investigate linear non-Gaussian acyclic models in the presence of latent Gaussian confounders (LiNGAM-GC), which assumes that the latent confounders are Gaussian distributed independently.

To the best of our knowledge, none of the existing methods discover the causal structure between clean labels and features by only using noisy data.

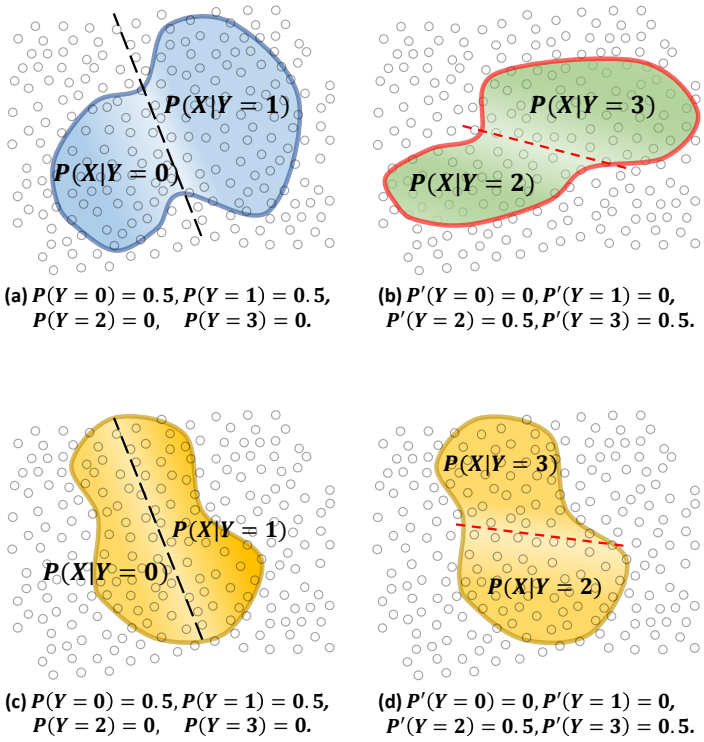


Figure 4.2: The change of $P(X)$ with the change of $P(Y)$ under different data generative processes.

4.3 Learning with noisy labels from a causal perspective

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

4.3.1 The influence of noisy data generative processes to different methods

The T-based method is independent of different generative processes. The reason is that these methods mainly rely on estimating the transition matrix $\boldsymbol{T}(\boldsymbol{x})$, which can be estimated by exploiting the noisy class posterior

$P(\tilde{Y}|X)$ learned on the noisy data [91, 48]. It is clear that the data generative process does not influence learning $P(\tilde{Y}|X)$ and $\boldsymbol{T}(\boldsymbol{x})$.

By contrast, heuristic SSL-based methods are influenced by data generative processes because they exploit the unlabeled data to help learn the classifier. The helpfulness of unlabeled data depends on whether $P(X)$ contains labeling information or not. According to the causal modularity property, when X causes Y, $P(X)$ does not contain labeling information, because $P(Y|X)$ and $P(X)$ are disentangled with each other. However, when Y causes X, $P(X)$ should contain labeling information, because $P(X)$ and $P(Y|X)$ are entangled with each other.

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 [65]. 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. 4.2. For example, as illustrated in Fig. 4.2(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. 4.2(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 about $P(Y|X)$. Therefore heuristic 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. 4.2(c) to Fig. 4.2(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 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 heuristic SSL-based methods may not work well in this case.

4.3.2 An intuitive method for the causal structure detection

In many real-world applications, the causal structure of the noisy data generative process is unknown. To discover the causal structure, we provide an intuitive **c**asual structure **d**etection method for learning with **n**oisy **l**abels (i.e., CDLN estimator). Our method relies on an asymmetric property of estimating flip rates under different generalization processes. Specifically, when X causes Y, the flip rate $P(\tilde{Y}|Y')$ estimated by an unsupervised classification method usually has a large estimation error, where Y' is pseudo labels estimated by the unsupervised method. However, when Y causes X, the estimation error is small.

Let $Y^* = \arg\max_i P(Y = i|\boldsymbol{x})$ be the Bayes label on the clean class-posterior distribution. To obtain the estimation error, we calculate the average difference between the noise rate estimated by the method based on modeling label noise and the noise rate estimated by a clustering algorithm,

i.e.,

$$d(P(\tilde{Y}|Y^*), P(\tilde{Y}|Y')) = \sum_{i}^{L}\sum_{j}^{L} \frac{|P(\tilde{Y}=j|Y^*=i) - P(\tilde{Y}=j|Y'=i)|}{L^2}. \tag{4.2}$$

The intuition is that given a noisy dataset, suppose that Bayes labels and pseudo labels of all instances are known and fixed, then the $P(\tilde{Y}|Y^*)$ and $P(\tilde{Y}|Y')$ are different in general unless Y^* and Y' are identical to each other.

To be more specific, the flip rate $P(\tilde{Y}|Y')$ can be obtained by letting a clustering method estimate pseudo labels Y' on all training instances. Given pseudo labels and noise labels, $P(\tilde{Y}|Y')$ then can also be estimated. On the *causal dataset* (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 $P(\tilde{Y}|Y')$ is large. On the *anticausal dataset* (Y causes X), $P(X)$ contains labeling information, the Y' should be "close" to clean label Y. Therefore the estimation error of $P(\tilde{Y}|Y')$ is small. We formally show this in the theorem below.

Theorem 4.3.1. *Let $P(\tilde{Y}|Y^*)$ be the transition relationship from the clean Bayes label Y^* to the noisy label $\tilde{Y}$; let $P(\tilde{Y}|Y')$ be the transition relationship from the pseudo label Y' to the noisy label $\tilde{Y}$. 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 Theorem 4.3.1, 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

Algorithm 2 CDLN 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. (4.5) 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 clustering 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 (4.4).

Output: The estimation $d(\hat{P}(\tilde{Y}|Y^*), \hat{P}(\tilde{Y}|Y'))$ via Eq. (4.2).

$P(Y^*|X)$, 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 4.3.1 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.

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 [49, 62, 20, 108], some of these methods can even have compatible performance with the supervised learning on some benchmark datasets such as STL10 [13] and CIFAR10 [41].

Estimation of $P(\tilde{Y}|Y')$. To estimate the flip rate $P(\tilde{Y}|Y')$, a clustering method is employed first to learn the clusters C. Then the clusters C can be converted into the pseudo label Y' by exploiting the estimated Bayes label $\hat{Y}^*$, and the average noise rate $P(\tilde{Y}|Y')$ obtained by a clustering method can be directly calculated. Be more specific, 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

$\hat{Y}^* = j$. Note that the estimated Bayes label $\hat{Y}^*$, is determined by the class i that maximizes the estimated clean class posterior $\hat{P}(Y = i|\boldsymbol{x})$, where the posterior is estimated using existing label-noise learning methods [64]. 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.,

$$\hat{Y}' = \arg\max_{j \in L} \frac{\sum_{x_k \in S_{\hat{Y}_j^*}} \mathbb{1}_{\{x_k \in S_{C_i}\}}}{N_{C_i}}. \tag{4.3}$$

Empirically, the assignment is implemented by applying Hungarian algorithm [37]. 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.,

$$\hat{P}(\tilde{Y} = j|Y' = i) = \frac{\sum_{(\boldsymbol{x},\tilde{y},\hat{y}')} \mathbb{1}_{\{\hat{Y}'=i \wedge \tilde{y}=j\}}}{\sum_{(\boldsymbol{x},\tilde{y},\hat{y}')} \mathbb{1}_{\{\hat{Y}'=i\}}}, \tag{4.4}$$

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.

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. [32]. 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 [48], 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.,

$$\begin{aligned} &\{\hat{A}^*, \hat{f}\} = \arg\min_{A,f} \frac{1}{N} \sum_{\boldsymbol{x},\tilde{y}} \ell_{ce}(\tilde{y}, Ah(\boldsymbol{x})), \\ &s.t. \max_i h_i(\boldsymbol{x}) = 1. \end{aligned} \tag{4.5}$$

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 [32] 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 [92]. After that, $P(\tilde{Y}|Y^*)$ can be estimated by using the same technique as in Eq. (4.4). However, $P(\tilde{Y}|Y,X)$ usually is hard to estimate [91], which leads to the learned classifier 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 4.4.1, 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 [48] for both instance-dependent and instance-independent label noise.

4.4 Experiments

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

Datasets and noise types. To validate the correctness of our estimator, we have employed 2 synthetic datasets which are xyGuassian and yxGuassian. We have also demonstrated the performance of our methods on 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. We manually inject label noise into all the datasets, and 20% of the data is left as the validation set. Three types of noise in our experiments are employed in our experiments. (1) symmetry flipping (Sym) [64] which randomly replaces a percentage of labels in the training data with all possible labels. (2) pair flipping (Pair) [25] where labels are only replaced by similar classes. (3) instance-dependent Label Noise (IDN) [91] where

different instances have different transition matrices depending on different 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 [31] and dropout (0.25) [85] is employed as the backbone method for all baselines. We employ LeNet-5 for MNIST [43] dataset and ResNet-18 [26] for CIFAR10 [41]. To estimate $P(\tilde{Y}|Y^*)$, we use SGD to train the classification network with batch size 128, momentum 0.9, weight decay 10^{-4}. The initial learning rate is 10^{-2}, and it decays at 30th and 60th epochs at the rate of 0.1, respectively. To get $P(\hat{Y}|Y')$, for xyGuassian, yxGuassain, KrKp, Balancescale, Splice and waveform which have low-dimensional features and small sample size, K-means clustering method [49] is employed; for MNIST, minibatch K-Means clustering method [78] is employed; for CIFAR10, the SPICE* [62] clustering method is employed.

Baselines. We compare the performance of heuristic SSL-based methods with T-based methods. The T-based methods employed are: (i) Forward [64] which estimates the transition matrix and embeds it to the neural network; (ii) Reweighting [50] which gives training examples different weights according to the transition matrix by importance reweighting; (iii) T-Revision [92] which refines the learned transition matrix to improve the classification accuracy. The heuristic SSL-based methods employed are (iv) JoCoR [90] which aims to reduce the diversity of two networks during training; (v) MoPro [47] which is a contrastive learning method that achieves online label noise correction (vi) Dividemix [46] which leverages the techniques FixMatch [82] and Mixup [102]; (viii) Mixup [102] which trains a neural network on convex combinations of pairs of examples and their labels. For all baseline methods, we follow their hyper-parameters settings as mentioned in their original paper. It is worth noting that, to let MoPro [47] work on non-image datasets, we have to modify its strong data augmentation for images to small Gaussian Noise, which may influence its performance.

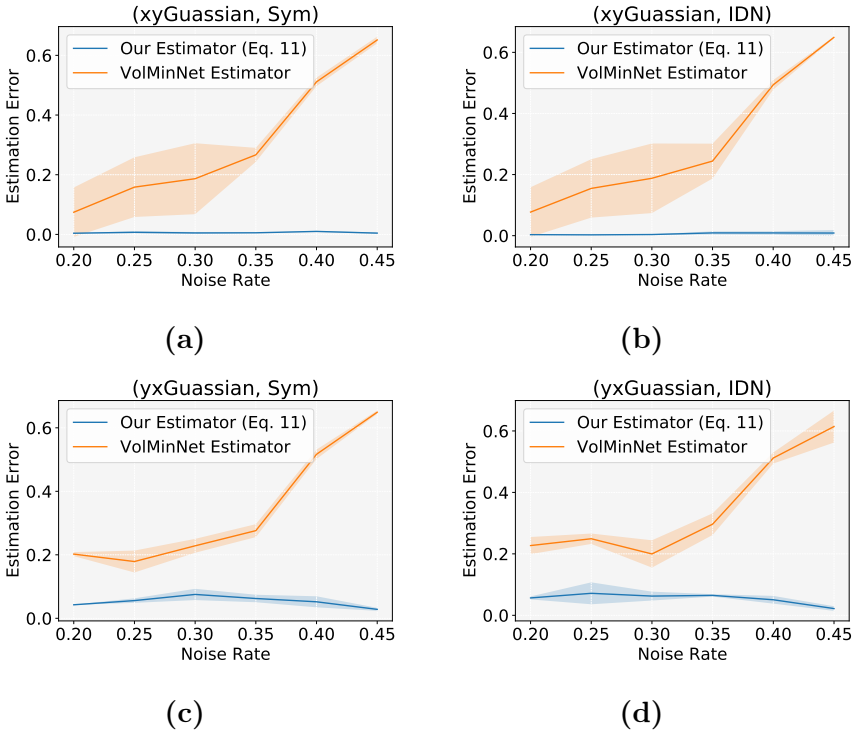


Figure 4.3: Estimation error of $P(\tilde{Y}|Y^*)$ on synthetic datasets with instance-independent and instance-dependent label noise. Our estimator outperforms the state-of-the-art method by a large margin.

4.4.1 Experiments on synthetic datasets

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 casual 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.

Estimation Error of $P(\tilde{Y}|Y^*)$

Note that, 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 $\hat{P}(\tilde{Y}|X)$. is obtained. The Bayes label Y^* can be directly obtained by using $\hat{P}(\tilde{Y}|X)$, and

	Sym			Instance		
	20% (0.196)	30% (0.142)	40% (0.081)	20% (0.180)	30 (0.127)	40% (0.071)
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	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	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	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	96.86±0.59	96.06±0.63	92.55±1.54	97.0±0.46	96.44±0.51	93.57±0.71

Table 4.1: Comparing test accuracies (%) of different methods on xyGuassian (anticausal) datasets with different levels and types of label noise. Estimations of CDLN estimator are shown in parentheses.

	Sym			Instance		
	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	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	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	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	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 4.2: Comparing test accuracies (%) of different methods on yxuassian (anticausal) datasets with different levels and types of label noise. Estimations of CDLN estimator are shown in parentheses.

$P(\tilde{Y}|Y^*)$ can be estimated by using the same technique as in Eq. (4.4). As illustrated in Fig. 4.3 the shows that the estimation error of our method is close to 0 not only on instance-independent label noise but also on instance-dependent label noise, which is much smaller than the estimated error obtained by employing VolMinNet. This clearly illustrates the advantage of our new $P(\tilde{Y}|Y^*)$ estimation method which does not require learning the transition matrix and clean label for each instance, but only requires estimating the average level of noise rates (Section 4.3.2).

Estimations of CDLN Estimator vs Classification Accuracies

In Tab. 4.2 and Tab. 4.1, we illustrate the estimations of CDLN estimator and the test accuracies of T-based methods and heuristic SSL-based methods for learning with label noise. The estimations of CDLN estimator are shown in parentheses, and each estimation is averaged over 5 repeated experiments. The matrix $\hat{A}^* = \hat{P}(\hat{Y}|Y^*)$ estimated by our method is embedded into T-based methods.

	Sym			Instance		
	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	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	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	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	93.31±1.1	88.81±1.03	73.84±1.18	93.19±1.31	87.25±1.49	74.31±3.42

Table 4.3: Comparing test accuracies (%) of different methods on KrKp (causal) datasets with different levels and types of label noise. Estimations of CDLN estimator are shown in parentheses.

	Sym		Pair		Instance	
	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	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	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	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	86.08±2.51	83.68±3.49	86.72±1.3	67.68±17.1	84.96±2.17	75.36±5.46

Table 4.4: Comparing test accuracies (%) of different methods on Balancescale (causal) datasets with different levels and types of label noise. Estimations of CDLN estimator are shown in parentheses.

It shows that the estimations on the anticausal dataset yxGuassian are much smaller than the causal dataset xyGuassian, which illustrates the effectiveness of our estimator. Specifically, when X is a cause of Y (anticausal), $P(X)$ does not contain information of $P(Y|X)$, then we expect that the difference $d(.)$ obtained by employing our estimator is large; when Y is a cause of X, $P(X)$ contain information of $P(Y|X)$, the difference $d(.)$ obtained by employing our estimator should be large. Additionally, on each of these datasets, for the same types of noise, the estimations of our estimator decrease with the increase in noise rates. It is mainly because that the labels of these two datasets are binary, and $P(\tilde{Y}|Y^* = 1)$ only has one degree of freedom, i.e., $P(\tilde{Y} = 1|Y^* = 1) = 1 - P(\tilde{Y} = 0|Y^* = 1)$. Therefore, if the difference between $\hat{P}(\tilde{Y} = 1|Y^* = 1)$ and $\hat{P}(\tilde{Y} = 1|Y' = 1)$ is small, the difference between $\hat{P}(\tilde{Y} = 0|Y^* = 1)$ and $\hat{P}(\tilde{Y} = 0|Y' = 1)$ will also be small.

The results show that on the causal dataset xyGuassian, T-based methods perform better than heuristic SSL-based methods. It is because that

	Sym		Pair		Instance	
	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	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	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	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	67.43±3.2	62.16±2.52	68.15±2.63	63.67±6.63	65.52±2.22	49.03±9.86

Table 4.5: Comparing test accuracies (%) of different methods on Splice (causal) datasets with different levels and types of label noise. Estimations of CDLN estimator are shown in parentheses.

	Sym		Pair		Instance	
	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	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	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	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	81.38±1.67	79.48±1.05	80.54±2.51	72.34±4.58	78.88±1.05	71.26±5.44

Table 4.6: Comparing test accuracies (%) of different methods on Waveform (anticausal) datasets with different levels and types of label noise. Estimations of CDLN estimator are shown in parentheses.

$P(X)$ does not contain information of $P(Y|X)$, then the state-of-the-art heuristic SSL-based methods relying on semi-supervised techniques may not be helpful. On the anticausal dataset xyGuassian, heuristic SSL-based methods are effective and have a similar performance to T-based methods. When the complexity of anticausal datasets is high, with a limited sample size, the heuristic SSL-based method should have better performance than T-based methods (See Tab. 4.7 and Tab. 4.8).

4.4.2 Experiments on Real-World Datasets

We illustrate the estimations of CDLN estimator and the test accuracies of T-based methods and heuristic SSL-based methods for learning with label noise on 6 real-world datasets. It illustrates similar results as on synthetic datasets. When the estimation of CDLN estimator is lower than 0.005, the heuristic SSL-based methods demonstrate their effectiveness. When the estimation of CDLN estimator is high, T-based methods are more helpful to improve the robustness of learning models. It is also worth

	Sym		Pair		Instance	
	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	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	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	**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	97.45±0.21	95.75±0.43	97.57±1.08	92.46±1.43	96.54±1.20	90.38±1.30

Table 4.7: Comparing test accuracies (%) of different methods on MNIST (anticausal) datasets with different levels and types of label noise. Estimations of CDLN estimator are shown in parentheses.

	Sym		Pair		Instance	
	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	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	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	**95.6**±0.10	**94.8**±1.10	**95.72**±0.04	**87.02** ±0.41	**95.5**±1.17	**94.5**±0.23
Mixup	93.2±0.31	86.2±0.3	92.23±0.71	82.43±1.02	93.32±0.25	87.61±0.56

Table 4.8: Comparing test accuracies (%) of different methods on CIFAR-10 (anticausal) datasets with different levels and types of label noise. Estimations of CDLN estimator are shown in parentheses.

mentioning that for waveform, although it is an anticasual dataset, T-based methods have better performance than heuristic SSL-based methods, and the estimation of CDLN estimator is large. The reason could be that 1). $P(X)$ may not always contain information about $P(Y|X)$ even if the data generative process is from X to Y, or 2). $P(X)$ may contain information about $P(Y|X)$, but the information can be hard to be exploited by existing methods.

4.5 Summary

In this Chapter, We show that T-based methods are disentangled with the data generative process. By contrast, SOTA heuristic methods require that $P(X)$ contains information about $P(Y|X)$ to improve the robustness. In this case, Y should be a cause of X. This further explains the importance of the transition matrix in learning with noisy labels. In many real-world applications, the causal structure of the data generative process is not given.

Then we proposed an intuitive method by exploiting the asymmetric property of estimating the flip rate under different generalization processes.

Chapter 5

Improving Transition-Matrix Estimation by Divide-and-Conquer

In this chapter, we propose a new transition matrix estimation method by exploiting the divide-and-conquer paradigm. Specifically, we introduce an intermediate class to avoid directly estimating the noisy class posterior. By this intermediate class, the original transition matrix can then be factorized into the product of two easy-to-estimate transition matrices. We term the proposed method the dual-T estimator. Both theoretical analyses and empirical results illustrate the effectiveness of the dual-T estimator for estimating transition matrices, leading to better classification performances.

5.1 Motivations and Contributions

The *transition matrix* $T(\boldsymbol{x}) = P(\tilde{Y} = j|Y = i, X = \boldsymbol{x})$ plays an essential role in designing *statistically consistent* classifiers. The basic idea is that the clean class posterior can be inferred by using the transition matrix and noisy class posterior (which can be estimated by using noisy data). In general, the transition matrix $T(\boldsymbol{x})$ is unidentifiable and thus hard to learn [92]. Current state-of-the-art methods [25, 24, 64, 63, 59] assume that the transition matrix is *class-dependent* and *instance-independent*, i.e., $P(\tilde{Y} = j|Y = i, X = \boldsymbol{x}) = P(\tilde{Y} = j|Y = i)$. Given *anchor points*, i.e., the data points that belong to a specific class almost surely, the class-dependent and instance-independent transition matrix is identifiable [50, 76], and it

could be estimated by exploiting the noisy class posterior of anchor points [50, 64, 100] (more details can be found in Chapter 5.2). In this paper, we will focus on learning the class-dependent and instance-independent transition matrix which can be used to improve the classification accuracy of the current methods if the matrix is learned more accurately.

The estimation error for the noisy class posterior is usually much larger than that of the clean class posterior, especially when the sample size is limited. An illustrative example is in Fig. 5.1. The rationale is that label noise is randomly generated according to a class-dependent transition matrix. Specifically, to learn the noisy class posterior, we need to fit the mapping from instances to clean (latent) labels, as well as the mapping from clean labels to noisy labels. Since the latter mapping is random and independent of instances, the learned mapping that fits label noise is prone to overfitting and thus will lead to a large estimation error for the noisy class posterior. The error will also lead to a large estimation error for the transition matrix. As estimating the transition matrix is a bottleneck for designing consistent classifiers, the large estimation error will significantly degenerate the classification performance [92].

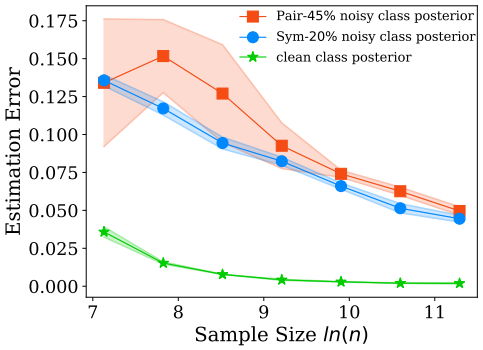


Figure 5.1: Estimation errors for clean class posteriors and noisy class posteriors on synthetic data. The estimation errors are calculated as the average absolute value between the ground truth and estimated class posteriors on $1,000$ randomly sampled test data points. The other details are the same as those of the synthetic experiments in Chapter 5.4

.

Motivated by this problem, in this paper, to reduce the estimation error of the transition matrix, we propose the *dual transition estimator* (*dual-T estimator*) to effectively estimate transition matrices. At a high level, by properly introducing an intermediate class, the dual-T estimator avoids directly estimating the noisy class posterior via factorizing the original transition matrix into two new transition matrices, which we denote

as $T^{\varkappa}$ and T^{ξ}. $T^{\varkappa}$ represents the transition from the clean labels to the intermediate class labels and T^{ξ} the transition from the clean and intermediate class labels to the noisy labels. Note that although we are going to estimate two transition matrices rather than one, we are not reducing the original problem to a harder one. In philosophy, our idea belongs to the divide and conquer paradigm, i.e., decomposing a hard problem into simple sub-problems and composing the solutions of the sub-problems to solve the original problem. The two new transition matrices are easier to estimate than the original transition matrix because we will show that (1) there is no estimation error for the transition matrix $T^{\varkappa}$, (2) the estimation error for the transition matrix T^{ξ} relies on predicting noisy class labels, which is much easier than learning a class posterior as the labels are discrete while the posteriors are continuous, and (3) the estimators for the two new transition matrices are easy to implement in practice. We will also theoretically analyze that the two new transition matrices are easier to predict than the original transition matrix. Empirical results on several datasets and label noise settings consistently justify the effectiveness of the dual-T estimator in reducing the estimation error of transition matrices and boosting the classification performance.

5.2 Related Work

Estimating transition matrix by exploiting the noisy class posterior of anchor points has been widely used [50, 64, 92] in label-noise learning and we term it the *transition estimator* (T estimator). Specifically, an instance $\boldsymbol{x}^i \in \mathcal{X}$ is an anchor point of the i-th clean class if $P(Y = i|\boldsymbol{x}^i) = 1$ [50, 92]. Suppose we can assess the noisy class posterior and anchor points, the transition matrix can be obtained via $P(\tilde{Y} = j|\boldsymbol{x}^i) = \sum_{k=1}^{C} P(\tilde{Y} = j|Y = k, \boldsymbol{x}^i)P(Y = k|\boldsymbol{x}^i) = P(\tilde{Y} = j|Y = i, \boldsymbol{x}) = T_{ij}$, where the second equation holds because $P(Y = k|\boldsymbol{x}^i) = 1$ when $k = i$ and $P(Y = k|\boldsymbol{x}^i) = 0$ otherwise. The last equation holds because the transition matrix is independent of the instance. According to the equation, to estimate the transition matrix, we need to find anchor points and estimate the noisy class posterior,

then the transition matrix can be estimated as follows,

$$\hat{P}(\tilde{Y}=j|\boldsymbol{x}^i)=\sum_{k=1}^{C}\hat{P}(\tilde{Y}=j|Y=k,\boldsymbol{x}^i)P(Y=k|\boldsymbol{x}^i)=\hat{P}(\tilde{Y}=j|Y=i,\boldsymbol{x}). \tag{5.1}$$

As the example illustrated in Fig. 5.1, with the same number of training examples, the estimation error of the noisy class posterior is significantly larger than that of the clean class posterior, which leads to a large estimation error of T estimator. This motivates us to seek an alternative estimator that avoids directly using the estimated noisy class posterior to approximate the transition matrix.

5.3 Methodology

To avoid directly using the estimated noisy class posterior to approximate the transition matrix, we propose a new estimator in this section.

5.3.1 dual-T estimator

By introducing an intermediate class, the transition matrix T can be factorized in the following way:

$$\begin{aligned} T_{ij}=P(\tilde{Y}=j|Y=i)&=\sum_{l\in\{1,\dots,C\}}P(\tilde{Y}=j|Y'=l,Y=i)P(Y'=l|Y=i)\\ &\triangleq\sum_{l\in\{1,\dots,C\}}T^{\xi}_{lj}(Y=i)T^{\varkappa}_{il}\ , \end{aligned} \tag{5.2}$$

where Y' represent the random variable for the introduced intermediate class, $T^{\xi}_{lj}(Y=i)=P(\tilde{Y}=j|Y'=l,Y=i)$, and $T^{\varkappa}_{il}=P(Y'=l|Y=i)$. Note that T^{ξ} and $T^{\varkappa}$ are two transition matrices representing the transition from the clean and intermediate class labels to the noisy class labels and the transition from the clean labels to the intermediate class labels, respectively.

By looking at Eq. (5.2), it seems we have changed an easy problem into a hard one. However, this is not true. Actually, we break down a problem into simple sub-problems. Combining the solutions to the sub-problems

gives a solution to the original problem. Thus, in philosophy, our idea belongs to the divide-and-conquer paradigm. In the rest of this subsection, we will explain why it is easy to estimate the transition matrices T^{ξ} and $T^{\varkappa}$. Moreover, in the next subsection, we will theoretically compare the estimation error of the dual-T estimator with that of the T estimator.

It can be found that $T^{\varkappa}_{ij} = P(Y' = j|Y = i)$ has a similar form to $T_{ij} = P(\tilde{Y} = j|Y = i)$. We can employ the same method that is developed for T, i.e., the T estimator, to estimate $T^{\varkappa}$. However, there seem to have two challenges: (1) it looks as if difficult to access $P(\boldsymbol{Y}'|\boldsymbol{x})$; (2) we may also have an error for estimating $P(\boldsymbol{Y}'|\boldsymbol{x})$. Fortunately, these two challenges can be well addressed by properly introducing the intermediate class. Specifically, we design the intermediate class Y' in such a way that $P(\boldsymbol{Y}'|\boldsymbol{x}) \triangleq \hat{P}(\tilde{\boldsymbol{Y}}|\boldsymbol{x})$, where $\hat{P}(\tilde{\boldsymbol{Y}}|\boldsymbol{x})$ represents an estimated noisy class posterior. Note that $\hat{P}(\tilde{\boldsymbol{Y}}|\boldsymbol{x})$ can be obtained by exploiting the noisy data at hand. As we have discussed, due to the randomness of label noise, estimating T directly will have a large estimation error especially when the noisy training sample size is limited. However, as we have access to $P(\boldsymbol{Y}'|\boldsymbol{x})$ directly, according to Eq. (5.1), the estimation error for $T^{\varkappa}$ is zero if anchor points are given[1].

Although the transition matrix T^{ξ} contains three variables, i.e., the clean class, intermediate class, and noisy class, we have class labels available for two of them, i.e., the intermediate class and noisy class. Note that the intermediate class labels can be assigned by using $P(\boldsymbol{Y}'|\boldsymbol{x})$. Usually, clean class labels are not available. This motivates us to find a way to eliminate the dependence on clean class for T^{ξ}. From an information-theoretic point of view [14], if the clean class Y is less informative for the noisy class $\tilde{Y}$ than the intermediate class Y', in other words, given $\tilde{Y}$, Y' contains no more information for predicting $\tilde{Y}$, then Y is independent of $\tilde{Y}$ conditioned on Y', i.e.,

$$T^{\xi}_{lj}(Y = i) = P(\tilde{Y} = j|Y' = l, Y = i) = P(\tilde{Y} = j|Y' = l). \tag{5.3}$$

[1]If the anchor points are to learn, the estimation error remains unchanged for the T estimator and dual-T estimator by employing $\boldsymbol{x}^i = \arg\max_{\boldsymbol{x}} \hat{P}(\tilde{Y} = i|\boldsymbol{x})$.

A sufficient condition for holding the above equalities is to let the intermediate class labels be identical to noisy labels. Note that it is hard to find an intermediate class whose labels are identical to noisy labels. The mismatch will be the main factor that contributes to the estimation error for T^{ξ}. Additionally, since we have labels for the noisy class and intermediate class, $P(\tilde{Y} = j|Y' = l)$ in Eq. (5.3) is easy to estimate by just counting the discrete labels, and it will have a small estimation error which converges to zero exponentially fast [9].

Based on the above discussion, by factorizing the transition matrix T into T^{ξ} and T^{η}, we can change the problem of estimating the noisy class posterior into the problem of fitting the noisy labels. Note that the noisy class posterior is in the range of $[0, 1]$ while the noisy class labels are in the set $\{1, \ldots, C\}$. Intuitively, learning the class labels are much easier than learning the class posteriors. In Chapter 5.4, our empirical experiments on synthetic and real-world datasets further justify this by showing a significant error gap between the estimation error of the T estimator and dual-T estimator.

Implementation of the dual-T estimator. The dual-T estimator is described in Algorithm 1. Specifically, the transition matrix T^{η} can be easily estimated by letting $P(Y' = i|\boldsymbol{x}) \triangleq \hat{P}(\tilde{Y} = i|\boldsymbol{x})$ and then employing the T estimator (see Chapter 5.2). By generating intermediate class labels, e.g., letting $\arg\max_{i\in\{1,\ldots,C\}} P(Y' = i|\boldsymbol{x})$ be the label for the instance $\boldsymbol{x}$, the transition matrix T^{ξ} can be estimating via counting, i.e.,

$$\hat{T}^{\xi}_{lj} = \hat{P}(\tilde{Y} = j|Y' = l) = \frac{\sum_i \mathbb{1}_{\{(\arg\max_k P(Y'=k|\boldsymbol{x}_i)=l)\wedge\tilde{y}_i=j\}}}{\sum_i \mathbb{1}_{\{\arg\max_k P(Y'=k|\boldsymbol{x}_i)=l\}}}, \tag{5.4}$$

where $\mathbb{1}_{\{A\}}$ is an indicator function which equals one when A holds true and zero otherwise, $(\boldsymbol{x}_i, \tilde{y}_i)$ are examples from the training sample S_{tr}, and $\wedge$ represents the AND operation.

Many statistically consistent algorithms [22, 64, 100, 92] consist of a two-step training procedure. The first step estimates the transition matrix and the second step builds statistically consistent algorithms, for example, via modifying loss functions. Our proposed dual-T estimator can be

Algorithm 3 dual-T estimator

Input: Noisy training sample S_{tr}; Noisy validation sample S_{val}.

1: Obtain the learned noisy class posterior, i.e., $\hat{P}(\tilde{\boldsymbol{Y}}|\boldsymbol{x})$, by exploiting training and validation sets;
2: Let $P(\boldsymbol{Y}'|\boldsymbol{x}) \triangleq \hat{P}(\tilde{\boldsymbol{Y}}|\boldsymbol{x})$ and employ T estimator to estimate $\hat{T}^{\curlywedge}$ according to Eq. (5.1);
3: Use Eq. (5.4) to estimate $\hat{T}^{\xi}$;
4: $\hat{T} = \hat{T}^{\xi}\hat{T}^{\curlywedge}$;

Output: The estimated transition matrix $\hat{T}$.

seamlessly embedded into their frameworks. More details can be found in Chapter 5.4.

5.3.2 Theoretical Analysis

In this subsection, we will justify that the estimation error could be greatly reduced if we estimate T^{ξ} and $T^{\curlywedge}$ rather than estimating T directly.

As we have discussed before, the estimation error of the T estimator is caused by estimating the noisy class posterior; the estimation error of the dual-T estimator comes from the estimation error of T^{ξ}, i.e., fitting the noisy class labels and estimating $P(\tilde{Y}|Y')$ by counting discrete labels. Note that to eliminate the dependence on the clean label for T^{ξ}, we need to achieve $P(Y' = \tilde{Y}|\boldsymbol{x}) = 1$. Let the estimation error for the noisy class posterior be Δ_1, i.e., $\left|P(\tilde{Y} = j|\boldsymbol{x}) - \hat{P}(\tilde{Y} = j|\boldsymbol{x})\right| = \Delta_1$. Let the estimation error for $P(\tilde{Y} = j|Y' = l)$ by counting discrete labels is Δ_2, i.e., $|P(\tilde{Y} = j|Y' = l) - \hat{P}(\tilde{Y} = j|Y' = l)| = \Delta_2$. Let the estimation error for fitting the noisy class labels is Δ_3, i.e., $P(Y' = \tilde{Y}|\boldsymbol{x}) = 1 - \Delta_3$. We will show that under the following assumption, the estimation error of the dual-T estimator is smaller than the estimation error of the T estimator.

Assumption 5.3.1. *For all* $\boldsymbol{x} \in \tilde{S}$, $\Delta_1 \geq \Delta_2 + \Delta_3$.

Assumption 5.3.1 is easy to hold. Theoretically, the error Δ_2 involves no predefined hypothesis space, and the probability that Δ_2 is larger than any positive number will converge to zero exponentially fast [9]. Thus, Δ_2 is usually much smaller than Δ_1 and Δ_3. We therefore focus on comparing Δ_1 with Δ_3 by ignoring Δ_2. Intuitively, the error Δ_3 is smaller than Δ_1

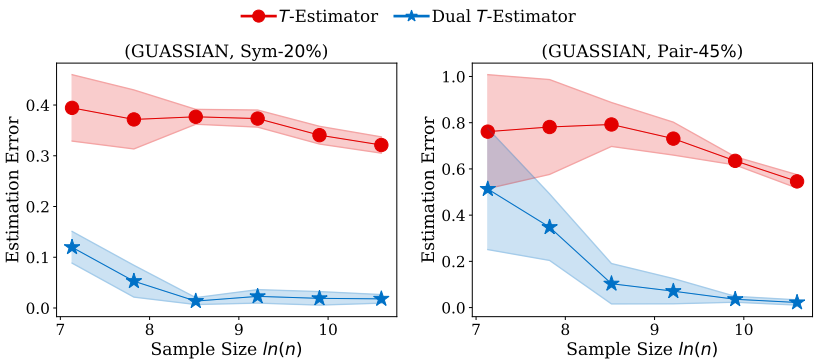


Figure 5.2: Estimation error of transition matrix on the synthetic dataset.

because it is easy to obtain a small estimation error for fitting noisy class labels than that for estimating noisy class posteriors. We note that the noisy class posterior is in the continuous range of $[0, 1]$ while the noisy class labels are in the discrete set $\{1, \ldots, C\}$. For example, suppose we have an instance $(\boldsymbol{x}, \tilde{y})$, then, as long as the empirical posterior probability $\hat{P}(\tilde{Y} = \tilde{y}|\boldsymbol{x})$ is greater than $1/C$, the noisy label will be accurately learned. However, the estimated error of the noisy class posterior probability can be up to $1 - 1/C$. We also empirically verify the relation among these errors in Appendix 2.

Theorem 5.3.1. *Under Assumption 5.3.1, the estimation error of the dual-T estimator is smaller than the estimation error of the T estimator.*

5.4 Experiments

We compare the transition matrix estimator error produced by the proposed dual-T estimator and the T estimator on both synthetic and real-world datasets. We also compare the classification accuracy of state-of-the-art label-noise learning algorithms [50, 64, 36, 25, 92, 102, 57] obtained by using the T estimator and the dual-T estimator, respectively. The MNIST [44], Fashion-MINIST (or F-MINIST) [93], CIFAR10, CIFAR100 [41], and Clothing1M [94] are used in the experiments. Note that as there is no estimation error for $T^{\nearrow}$, we do not need to do an ablation study to show how the two new transition matrices contribute to the estimation error for transition matrix estimation.

5.4.1 Transition Matrix Estimation

We compare the estimation error between our estimator and the T estimator on both synthetic and real-world datasets with different sample sizes and different noise types. The synthetic dataset is created by sampling from 2 different 10-dimensional Gaussian distributions. One of the distributions has unit variance and zero mean among all dimensions. Another one has variance=1 and mean=2 among all dimensions. The real-world image datasets used to evaluate transition matrices estimation error are MNIST [44], F-MINIST [93], CIFAR10, and CIFAR100 [41].

We conduct experiments on the commonly used noise types [25, 92]. Specifically, two representative structures of the transition matrix T will be investigated: Symmetry flipping (Sym-ϵ) [64]; (2) Pair flipping (Pair-ϵ) [25]. To generate noisy datasets, we corrupt the training and validation set of each dataset according to the transition matrix T.

Neural network classifiers are used to estimate transition matrices. For fair comparisons, the same network structure is used for both estimators. Specifically, on the synthetic dataset, a two-hidden-layer network is used, and the hidden unit size is 25; on the real-world datasets, we follow the network structures used by the state-of-the-art method [64], i.e., using a LeNet network with dropout rate 0.5 for MNIST, a ResNet-18 network for F-MINIST and CIFAR10, a ResNet-34 network for CIFAR100, and a ResNet-50 pre-trained on ImageNet for Clothing1M. The network is trained for 100 epochs, and stochastic gradient descent (SGD) optimizer is used. The initial learning rate is 0.01, and it is decayed by a factor of 10 after 50-th epoch. We use 20% training examples for validation, and the model with the best validation accuracy is selected for estimating the transition matrix. The estimation error is calculated by measuring the ℓ_1-distance between the estimated transition matrix and the ground truth T. The average estimation error and the standard deviation over 5 repeated experiments for both estimators are illustrated in Fig. 5.2, Fig. 5.3 and Fig. 5.4.

Fig. 5.2 illustrates the estimation error of the T estimator and the dual T estimation on the synthetic dataset. For two different noise types and sample sizes, the estimation error of both estimation methods tends to

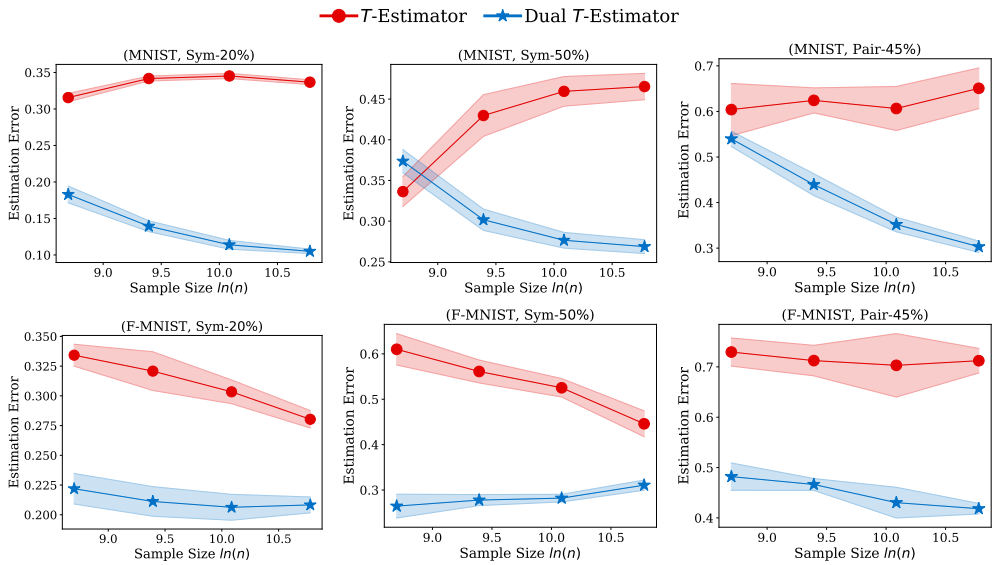


Figure 5.3: Transition matrix estimation error on MNIST and F-MNIST. The error bar for standard deviation in each figure has been shaded. The lower the better.

decrease with the increase in training sample sizes. However, the estimation error of the dual-T estimator is continuously smaller than that of the T estimator. Moreover, the estimation error of the dual-T estimator is less sensitive to different noise types compared to the T estimator. Specifically, even T estimator is trained with all the training examples, its estimation error on Pair-45% noise is approximately doubled than that on Sym-20 noise, which is observed by looking at the right-hand side of the estimation error curves. In contrast, when training the dual T estimator with all the training examples, its estimation error on the different noise types does not significantly differ, which is all less than 0.1. Similar to the results on the synthetic dataset, the experiments on the real-world image datasets illustrated in Fig. 5.3 and Fig. 5.4 also show that the estimation error of the dual-T estimator is continuously smaller than that of the T estimator except for CIFAR100, which illustrates the effectiveness of the proposed *DT*-estimator. On CIFAR100, both estimators have a larger estimation error compared to the results on MNIST, F-MINIST, and CIFAR10. The dual-T estimator outperforms the T estimator with a large sample size. However, when the training sample size is small, the estimation error of the dual-T estimator can be larger than that of the T estimator, it is because

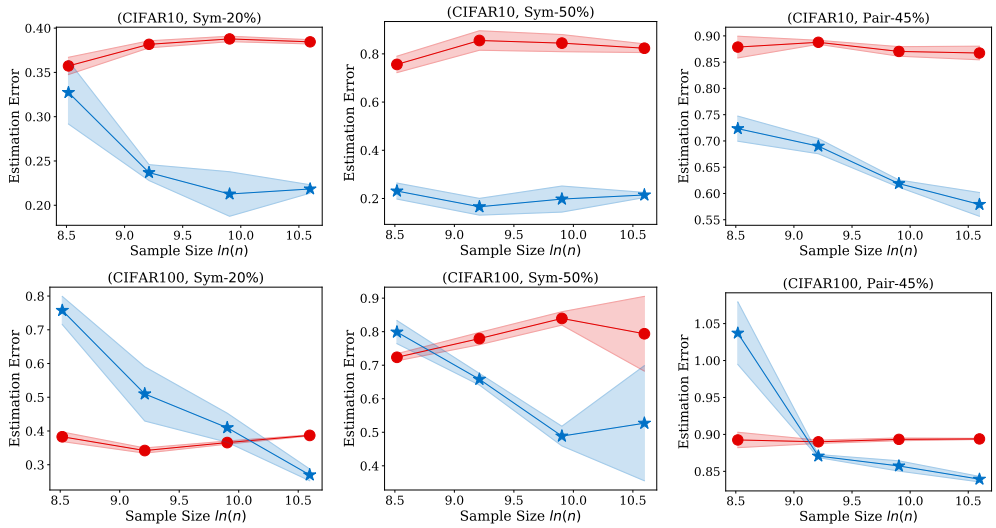


Figure 5.4: Transition matrix estimation error on CIFAR10 and CIFAR100. The error bar for standard deviation in each figure has been shaded. The lower the better.

the number of images per class is too small to estimate the transition matrix $\hat{T}^{\xi} \in \mathbb{R}^{100\times100}$ which can be very sparse and lead to a large estimation error.

5.4.2 Classification accuracy Evaluation

We investigate how the estimation of the T estimator and the dual-T estimator will affect the classification accuracy in label-noise learning. The experiments are conducted on MNIST, F-MINIST, CIFAR10, CIFAR100, and Clothing1M. The classification accuracy are reported in Table 5.1 and Table 5.2. Eight popular baselines are selected for comparison, i.e., Coteaching [25], and MentorNet [36] which use diagonal entries of the transition matrix to help select reliable examples used for training; Forward [64], and Revision [92], which use the transition matrix to correct hypotheses; Reweighting [50], which uses the transition matrix to build risk-consistent algorithms. There are three baselines without requiring any knowledge of the transition matrix, i.e., CE, which trains a network on the noisy sample directly by using cross-entropy loss; Decoupling [57], which trains two networks and updates the parameters only using the examples which have a different prediction from two classifiers; Mixup [102] which reduces the memorization of corrupt labels by using linear interpolation to feature-target pairs. The

	MNIST			F-MNIST		
	Sym-20%	Sym-50%	Pair-45%	Sym-20%	Sym-50%	Pair-45%
CE	95.77 ± 0.11	93.99 ± 0.21	90.11 ± 0.96	89.70 ± 0.14	87.22 ± 0.29	73.94 ± 1.44
Mixup	91.14 ± 0.28	77.18 ± 2.89	80.14 ± 1.74	91.82 ± 0.09	89.83 ± 0.11	86.98 ± 0.85
Decoupling	98.34 ± 0.12	63.70 ± 0.52	56.66 ± 0.25	92.03 ± 0.37	86.96 ± 0.86	70.87 ± 2.00
T-MentorNet	91.51 ± 0.31	**81.59** ± 3.25	62.10 ± 4.11	87.18 ± 0.31	**79.32** ± 2.08	49.65 ± 3.18
DT-MentorNet	**96.73** ± 0.07	78.99 ± 0.4	**85.27** ± 1.19	**92.93** ± 0.07	75.67 ± 0.31	**81.84** ± 1.34
T-Coteaching	93.41 ± 0.15	**84.13** ± 2.77	63.60 ± 3.10	88.10 ± 0.29	**83.43** ± 0.41	58.18 ± 7.00
DT-Coteaching	**97.52*** ± 0.07	83.20 ± 0.43	**86.78** ± 0.76	**93.90*** ± 0.06	77.45 ± 0.59	**87.37** ± 1.13
T-Forward	96.85 ± 0.07	95.22 ± 0.13	94.92 ± 0.89	90.99 ± 0.16	88.58 ± 0.30	82.50 ± 3.45
DT-Forward	**97.24** ± 0.07	**95.89** ± 0.14	**97.24*** ± 0.10	**91.37** ± 0.09	**89.52** ± 0.27	**91.91*** ± 0.24
T-Reweighting	96.80 ± 0.05	95.25 ± 0.23	91.50 ± 1.27	90.94 ± 0.29	88.82 ± 0.52	80.94 ± 3.38
DT-Reweighting	**97.34** ± 0.04	**96.19** ± 0.13	**96.62** ± 0.21	**91.68** ± 0.21	**90.17** ± 0.12	**88.31** ± 1.76
T-Revision	96.79 ± 0.04	95.26 ± 0.21	91.83 ± 1.08	91.20 ± 0.12	88.77 ± 0.36	85.26 ± 5.29
DT-Revision	**97.40** ± 0.04	**96.21*** ± 0.13	**96.71** ± 0.12	**91.78** ± 0.16	**90.18*** ± 0.10	**90.70** ± 0.37
	CIFAR10			CIAR100		
	Sym-20%	Sym-50%	Pair-45%	Sym-20%	Sym-50%	Pair-45%
CE	69.37 ± 0.47	55.92 ± 0.44	46.47 ± 1.81	33.16 ± 0.56	22.65 ± 0.37	21.62 ± 0.58
Mixup	80.33 ± 0.59	61.10 ± 0.26	58.37 ± 2.66	47.79 ± 0.91	30.17 ± 0.74	30.34 ± 0.72
Decoupling	81.63 ± 0.34	57.63 ± 0.47	52.30 ± 0.16	48.51 ± 0.61	26.01 ± 0.40	33.13 ± 0.49
T-MentorNet	79.00 ± 0.20	31.09 ± 3.99	26.19 ± 2.24	50.09 ± 0.28	36.66 ± 9.13	20.14 ± 0.77
DT-MentorNet	**88.07** ± 0.54	**69.34** ± 0.61	**69.31** ± 1.90	**59.7** ± 0.41	**37.23** ± 5.69	**30.88** ± 0.58
T-Coteaching	79.47 ± 0.20	39.71 ± 3.52	33.96 ± 3.24	50.87 ± 0.77	38.09 ± 8.63	24.58 ± 0.70
DT-Coteaching	**90.37*** ± 0.12	**71.49** ± 0.65	**76.51** ± 4.97	**60.63*** ± 0.36	**38.21*** ± 5.91	**35.46*** ± 0.33
T-Forward	75.36 ± 0.39	65.32 ± 0.57	54.70 ± 3.07	37.45 ± 0.54	27.91 ± 1.48	25.10 ± 0.77
DT-Forward	**78.36** ± 0.34	**69.94** ± 0.66	**55.75** ± 1.53	**41.76** ± 0.97	**32.69** ± 0.73	**26.08** ± 0.93
T-Reweighting	73.28 ± 0.44	64.20 ± 0.38	50.19 ± 1.10	38.07 ± 0.34	27.26 ± 0.50	**25.86** ± 0.55
DT-Reweighting	**79.09** ± 0.21	**73.29** ± 0.23	**52.65** ± 2.25	**41.04** ± 0.72	**34.56** ± 1.39	25.84 ± 0.42
T-Revision	75.71 ± 0.93	65.66 ± 0.44	75.14 ± 2.43	38.25 ± 0.27	27.70 ± 0.64	25.74 ± 0.44
DT-Revision	**80.45** ± 0.39	**73.76*** ± 0.22	**77.72*** ± 1.80	**42.11** ± 0.76	**35.09** ± 1.44	**26.10** ± 0.43

Table 5.1: Classification accuracy (percentage) on MNIST, F-MNIST, CIFAR10, and CIFAR100.

CE	Mixup	Decoupling	$T(DT)$-MentorNet
69.03	71.29	54.63	57.63 (**60.25**)
$T(DT)$-Coteaching	$T(DT)$-Forward	$T(DT)$-Reweighting	$T(DT)$-Revision
60.37 (**64.54**)	69.93 (**70.17**)	70.38 (**70.86**)	71.01 (**71.49***)

Table 5.2: Classification accuracy (percentage) on Clothing1M.

estimation of the T estimator and the dual-T estimator are both applied to the baselines which rely on the transition matrix. The baselines using the estimation of T estimator are called T-Coteaching, T-MentorNet, T-Forward, T-Revision, and T-Reweighting. The baselines using estimation of dual-T estimator are called DT-Coteaching, DT-MentorNet, DT-Forward, DT-Revision, and DT-Reweighting.

The settings of our experiments may be different from the original paper, thus the reported accuracy can be different. For instance, in the original paper of Coteaching [25], the noise rate is given, and all data are used for training. In contrast, we assume the noise rate is unknown and needed to be estimated. We only use 80% data for training, since 20% data

are left out as the validation set for transition matrix estimation. In the original paper of T-revision [92], the experiments on Clothing1M use clean data for validation. In contrast, we only use noisy data for validation.

In Table 5.1 and Table 5.2, we bold the better classification accuracy produced by the baseline methods integrated with the T estimator or the dual-T estimator. The best classification accuracy among all the methods in each column is highlighted with $*$. The tables show the classification accuracy of all the methods by using our estimation is better than using that of the T estimator for most of the experiments. It is because the dual-T estimator leads to a smaller estimation error than the T estimator when training with a large sample size, which can be observed at the right-hand side of the estimation error curves in Fig. 5.3 and Fig. 5.4. The baselines with the most significant improvement by using our estimation are Coteaching and MentorNet. *DT*-Coteaching outperforms all the other methods under Sym-20% noise. On Clothing1M dataset, *DT*-revision has the best classification accuracy. The experiments on the real-world datasets not only show the effectiveness of the dual-T estimator for improving the classification accuracy of the current noisy learning algorithms but also reflect the importance of transition matrix estimation in label-noise learning.

5.4.3 Empirical Validation of Assumption 5.3.1

We empirically verify the relations among the three different errors in Assumption 5.3.1. Note that Δ_1 is the estimation error for the noisy class posterior, i.e., $\Delta_1 = \left|P(\tilde{Y}=j|\boldsymbol{x}) - \hat{P}(\tilde{Y}=j|\boldsymbol{x})\right|$; Δ_2 is the estimation error for counting discrete labels, i.e., $|P(\tilde{Y}=j|Y'=l) - \hat{P}(\tilde{Y}=j|Y'=l)| = \Delta_2$; Δ_3 is the estimation error for fitting the noisy class labels, i.e., $P(Y'=\tilde{Y}|\boldsymbol{x}) = 1-\Delta_3$.

The experiments are conducted on the synthetic dataset, and the setting is the same as those of the synthetic experiments in Chapter 4. The three errors are calculated on the training set since both the T estimator and the dual-T estimator estimate the transition matrix on the training set.

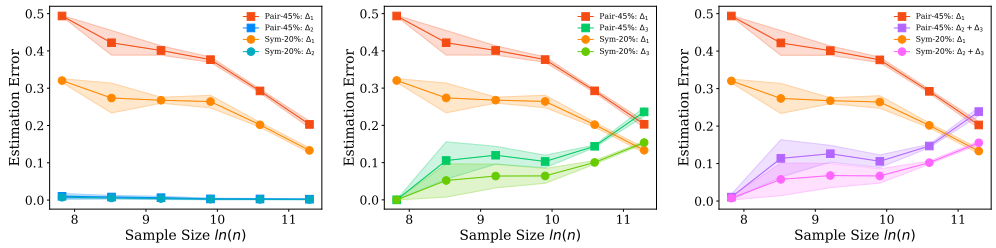


Figure 5.5: The relations among Δ_1, Δ_2 and Δ_3

Figure 5.5 shows that the error Δ_2 is very small and can be ignored. Δ_3 is continuously smaller than Δ_1 when the sample size is small. The recent work [15] shows that the sample complexity of the network is linear in the number of parameters, which means that, usually, we may not have enough training examples to learn the noisy class posterior well (e.g., CIFAR10, CIFAR100, and Fashion-MNIST), and Assumption 5.3.1 can be easily satisfied. It is worth mentioning that, even Assumption 5.3.1 does not hold, the estimation error of the dual-T estimator may also be smaller than the T estimator. Specifically, the error ϵ_{DT} of the proposed estimator is upper bounded by $C^2(\Delta_2 + \Delta_3)$. Generally, the increase of the upper bound $C^2(\Delta_2 + \Delta_3)$ does not imply the increase of the error ϵ_{DT}.

5.5 Summary

The transition matrix T plays an important role in label-noise learning. In this paper, to avoid the large estimation error of the noisy class posterior leading to the poorly estimated transition matrix, we have proposed a new transition matrix estimator named dual-T estimator. The new estimator estimates the transition matrix by exploiting the divide-and-conquer paradigm, i.e., factorizes the original transition matrix into the product of two easy-to-estimate transition matrices by introducing an intermediate class state. Both theoretical analysis and experiments on both synthetic and real-world label noise data show that our estimator reduces the estimation error of the transition matrix, which leads to a better classification accuracy for the current label-noise learning algorithms.

Chapter 6

Learning Instance-Independent Transition Matrices via Causality

In this chapter, by leveraging a structural causal model, we propose a novel generative approach for instance-dependent transition-matrix learning. In particular, we show that properly modeling the instances will contribute to the identifiability of the label noise transition matrix and thus lead to a better classifier. Empirically, our method outperforms all state-of-the-art methods on both synthetic and real-world label-noise datasets.

6.1 Motivations and Contributions

Inspired by causal learning [65, 69, 74], we provide a new causal perspective of label-noise learning by exploiting the causal information to further contribute to the identifiability of the transition matrix $P(\tilde{Y}|Y, X)$ other than making assumptions directly on the transition relationship. Specifically, we assume that the data containing instance-dependent label noise is generated according to the causal graph in Fig. 6.1. In real-world applications, many datasets are generated according to the proposed generative process. For example, for the Street View House Numbers (SVHN) dataset [60], X represents the image containing the digit; Y represents the clean label of the digit shown on the plate; Z represents the latent variable that captures the information affecting the generation of the images, e.g., orientation, lighting, and font style. Here Y is clearly a cause of X because the causal

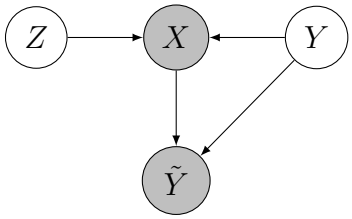


Figure 6.1: A graphical causal model reveals a generative process of the data that contains instance-dependent label noise, where the shaded variables are observable and the unshaded variables are latent.

generative process can be described in the following way. First, the house plate is generated according to the street number and attached to the front door. Then, the house plate is captured by a camera (installed in a Google street view car) to form X, taking into account other factors such as illumination and viewpoint. Finally, the images containing house numbers are collected and relabeled to form the dataset. Let us denote the annotated label by the noisy label $\tilde{Y}$ as the annotator may not be always reliable, especially when the dataset is very large but the budget is limited. During the annotation process, the noisy labels were generated according to both the images and the range of predefined digit numbers. Hence, both X and Y are causes of $\tilde{Y}$. Note that most existing image datasets are collected with the causal relationship that Y causes X. For example, the widely used *FashionMNIST* and *CIFAR*. When we synthesize instance-dependent label noise based on them, we will have the causal graph illustrated in Fig. 6.1. Note also that some datasets are generated with the causal relationship that X causes Y. Other than using domain knowledge, the different causal relationships can be verified by employing causal discovery [83, 84, 69].

When the latent clean label Y is a cause of X, $P(X)$ will contain some information about $P(Y|X)$. This is because, under such a generative process, the distributions of $P(X)$ and $P(Y|X)$ are entangled [73]. To help estimate $P(Y|X)$ with $P(X)$, we make use of the causal generative process to estimate $P(X|Y)$, which directly benefits from $P(X)$ by generative modeling. The modeling of $P(X|Y)$ in turn encourages the identifiability of the transition relationship and benefits the learning $P(Y|X)$. For example, in Fig. 6.2(a), we have added instance-dependent label-noise with rate 45% (i.e., IDLN-45%) to the MOON dataset and employed different methods [25, 102] to solve the label-noise learning problem. As illustrated in Fig. 6.2(b) and Fig. 6.2(c), previous methods fail to infer clean labels. In contrast, by constraining the conditional distribution of the instances,

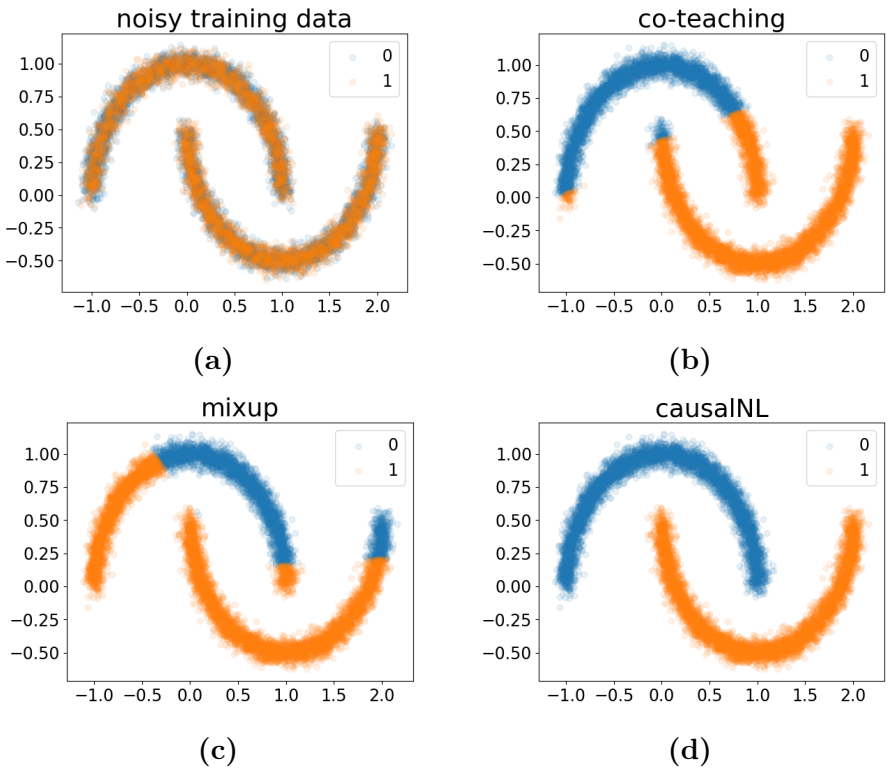


Figure 6.2: (a) An illustration of the MOON training dataset which contains 45% of instance-dependent label noise. (b)-(d) The illustration of the classification performance of co-teaching, mixup and our method, respectively.

i.e., restricting the data of each class to be on a manifold by setting the dimension of the latent variable Z to be 1-dimensional, the label transition, as well as the clean labels, can be successfully recovered (by the proposed method), which is showed in Fig. 6.2(d).

Specifically, to make use of the causal graph to contribute to the identifiability of the transition matrix, we propose a causally inspired deep generative method, which models the causal structure with all the observable and latent variables, i.e., the instance X, noisy label $\tilde{Y}$, latent feature Z, and the latent clean label Y. The proposed generative model captures the variables' relationship indicated by the causal graph. Furthermore, built on the variational autoencoder (VAE) framework [39], we build an inference network which could efficiently infer the latent variables Z and Y when maximizing the marginal likelihood $p(X, \tilde{Y})$ on the given noisy data. In

the decoder phase, the data will be reconstructed by exploiting the conditional distribution of instances $P(X|Y,Z)$ and the transition relationship $P(\tilde{Y}|Y,X)$, i.e.,

$$p_\theta(X,\tilde{Y}) = \int_{z,y} P(Z=z)P(Y=y)p_{\theta_1}(X|Y=y,Z=z)p_{\theta_2}(\tilde{Y}|Y=y,X)\mathrm{d}z\mathrm{d}y$$

will be exploited, where $\theta := (\theta_1, \theta_2)$ are the parameters of the causal generative model (more details can be found in Section 6.3). At a high level, according to the equation, given the noisy data and the distributions of Z and Y, constraining $p_{\theta_1}(X|Y,Z)$ will also greatly reduce the uncertainty of $p_{\theta_2}(\tilde{Y}|Y,X)$ and thus contribute to the identifiability of the transition matrix. Note that adding a constraint on $p_{\theta_1}(X|Y,Z)$ is natural, for example, images often have a low-dimensional manifold [7]. We can restrict $P(Z)$ to fulfill the constraint on $p_{\theta_1}(X|Y,Z)$. By exploiting the causal structure and the constraint on instances to better model label noise, the proposed method significantly outperforms the baselines. When the label noise rate is large, the superiority is evidenced by a large gain in the classification performance.

6.2 Related Work

Here we briefly introduce some background knowledge of causality [65] used in this paper. A structural causal model (SCM) consists of a set of variables connected by a set of functions. It represents a flow of information and reveals the causal relationship among all the variables, providing a fine-grained description of the data generation process. The causal structure encoded by SCMs can be represented as a graphical causal model as shown in Fig. 6.1, where each node is a variable and each edge is a function. The SCM corresponding to the graph in Fig. 6.1 can be written as

$$Z = \epsilon_Z, \quad Y = \epsilon_Y, \quad X = f(Z,Y,\epsilon_X), \quad \tilde{Y} = f(X,Y,\epsilon_{\tilde{Y}}), \tag{6.1}$$

where $\epsilon_\cdot$ are independent exogenous variables following some distributions. The occurrence of the exogenous variables makes the generation of X and $\tilde{Y}$ a stochastic process. Each equation specifies a distribution of a variable

conditioned on its parents (could be an empty set).

By observing the SCM, the helpfulness of the instances to learning the classifier can be clearly explained. Specifically, the instance X is a function of its label Y and latent feature Z which means that the instance X is generated according to Y and Z. Therefore X must contain information about its clean label Y and latent feature Z. That is the reason that $P(X)$ can help identify $P(Y|X)$ and also $P(Z|X)$. However, since we do not have clean labels, it is hard to fully identify $P(Y|X)$ from $P(X)$ in the unsupervised setting. For example, on the MOON dataset shown in Fig. 6.2, it is possible to discover the two clusters by enforcing the manifold constraint, but it is impossible to determine which class each cluster belongs to. We discuss in the following that we can make use of the property of $P(X|Y)$ to help model label noise, i.e., encourage the identifiability of the transition relationship, thereby learning a better classifier.

Specifically, under the Markov condition [65], which intuitively means the independence of exogenous variables, the joint distribution $P(\tilde{Y}, X, Y, Z)$ specified by the SCM can be factorized into the following

$$P(X, \tilde{Y}, Y, Z) = P(Y)P(Z)P(X|Y, Z)P(\tilde{Y}|Y, X). \tag{6.2}$$

This motivates us to extend VAE [39] to perform inference in our causal model to fit the noisy data in the next section. In the decoder phase, given the noisy data and the distributions of Z and Y, adding a constraint on $P(X|Y, Z)$ will reduce the uncertainty of the distribution $P(\tilde{Y}|Y, X)$. In other words, modeling $P(X|Y, Z)$ will encourage the identifiability of the transition relationship and thus better model label noise. Since $P(\tilde{Y}|Y, X)$ functions as a bridge to connect the noisy labels to clean labels, we therefore can better learn $P(Y|X)$ or the classifier by only using the noisy data.

There are normally two ways to add constraints on the instances, i.e., assuming a specific parametric generative model or introducing prior knowledge of the instances. In this paper, since we mainly study the image classification problem with noisy labels, we focus on the manifold property of images and add the low-dimensional manifold constraint to the instances.

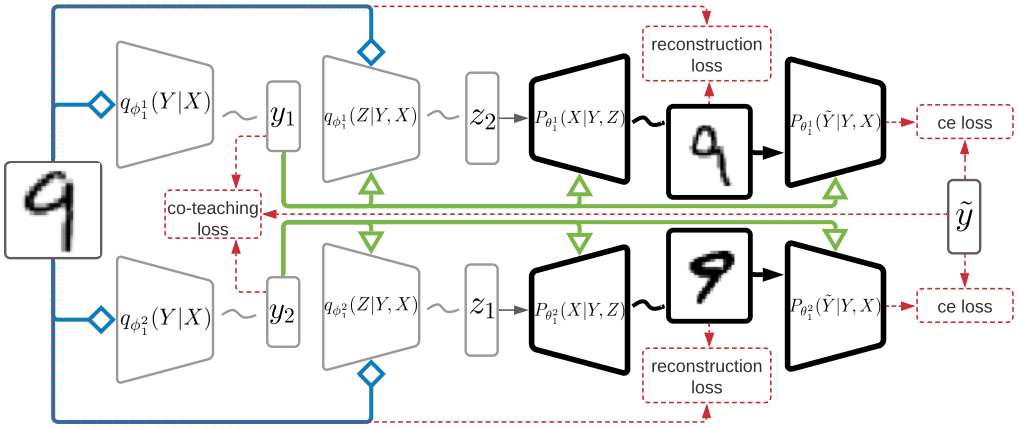


Figure 6.3: A working flow of our method.

6.3 Causality Captured Instance-dependent Label-Noise Learning

In this section, we propose a structural generative method that captures the causal relationship and utilizes $P(X)$ to help identify the label-noise transition matrix, therefore, our method leads to a better classifier that assigns more accurate labels.

6.3.1 Variational Inference under the Structural Causal Model

To model the generation process of noisy data and to approximate the distribution of the noisy data, our method is designed to follow the causal factorization (see Eq. (6.2)). Specifically, our model contains two decoder networks that jointly model a distribution $p_\theta(X, \tilde{Y}|Y, Z)$ and two encoders (inference) networks that jointly model the posterior distribution $q_\phi(Z, Y|X)$. Here we discuss each component of our model in detail.

Let the two decoder networks model the distributions $p_{\theta_1}(X|Y, Z)$ and $p_{\theta_2}(\tilde{Y}|Y, X)$, respectively. Let θ_1 and θ_2 be learnable parameters of the distributions. Without loss of generality, we set $p(Z)$ to be a standard normal distribution and $p(Y)$ to be a uniform distribution. Then, modeling the joint distribution in Eq. (6.2) boils down to modeling the distribution

$p_\theta(X, \tilde{Y}|Y, Z)$, which is decomposed as follows:

$$p_\theta(X, \tilde{Y}|Y, Z) = p_{\theta_1}(X|Y, Z)p_{\theta_2}(\tilde{Y}|Y, X). \tag{6.3}$$

To infer latent variables Z and Y with only observable variables X and $\tilde{Y}$, we could design an inference network that models the variational distribution $q_\phi(Z, Y|\tilde{Y}, X)$. Specifically, let $q_{\phi_2}(Z|Y, X)$ and $q_{\phi_1}(Y|\tilde{Y}, X)$ be the distributions parameterized by learnable parameters ϕ_1 and ϕ_2, the posterior distribution can be decomposed as follows:

$$q_\phi(Z, Y|\tilde{Y}, X) = q_{\phi_2}(Z|Y, X)q_{\phi_1}(Y|\tilde{Y}, X), \tag{6.4}$$

where we do not include $\tilde{Y}$ as a conditioning variable in $q_{\phi_2}(Z|Y, X)$ because the causal graph implies $Z \perp\!\!\!\perp \tilde{Y}|X, Y$. One problem with this posterior form is that we cannot directly employ $q_{\phi_1}(Y|\tilde{Y}, X)$ to predict labels on the test data, on which $\tilde{Y}$ is absent.

To allow our method efficiently and accurately infer clean labels, we approximate $q_{\phi_1}(Y|\tilde{Y}, X)$ by assuming that given the instance X, the clean label Y is conditionally independent of the noisy label $\tilde{Y}$, i.e., $q_{\phi_1}(Y|\tilde{Y}, X) = q_{\phi_1}(Y|X)$. This approximation does not have a very large approximation error because the images contain sufficient information to predict the clean labels. Thus, we could simplify Eq. (6.4) as follows

$$q_\phi(Z, Y|X) = q_{\phi_2}(Z|Y, X)q_{\phi_1}(Y|X), \tag{6.5}$$

such that our encoder networks model $q_{\phi_2}(Z|Y, X)$ and $q_{\phi_1}(Y|X)$, respectively. In such a way, $q_{\phi_1}(Y|X)$ can be used to infer clean labels efficiently. We also found that the encoder network modeling $q_{\phi_1}(Y|X)$ acts as a regularizer which helps to identify $p_{\theta_2}(\tilde{Y}|Y, X)$. Moreover, be benefited from this, our method can be a general framework which can easily integrate with the current discriminative label-noise methods [92, 57, 25], and we will showcase it by collaborating co-teaching [25] with our method.

Optimization of Parameters Because the marginal distribution $p_\theta(X, \tilde{Y})$ is usually intractable, to learn the set of parameters $\{\theta_1, \theta_2, \phi_1, \phi_2\}$ given only noisy data, we follow the variational inference framework [8] to

minimize the negative evidence lower-bound $-\text{ELBO}(x, \tilde{y})$ of the marginal likelihood of each datapoint $(x, \tilde{y})$ instead of maximizing the marginal likelihood itself. By ensembling our decoder and encoder networks, $-\text{ELBO}(x, \tilde{y})$ is derived as follows:

$$\begin{aligned} -\text{ELBO}(x, \tilde{y}) = & -\mathbb{E}_{(z,y)\sim q_\phi(Z,Y|\boldsymbol{x})}\left[\log p_{\theta_1}(x|y,z)\right] - \mathbb{E}_{y\sim q_{\phi_1}(Y|\boldsymbol{x})}\left[\log p_{\theta_2}(\tilde{y}|y,x)\right] \\ & + kl(q_{\phi_1}(Y|\boldsymbol{x})\|p(Y)) + \mathbb{E}_{y\sim q_{\phi_1}(Y|\boldsymbol{x})}\left[kl(q_{\phi_2}(Z|y,x)\|p(Z))\right], \end{aligned} \tag{6.6}$$

where $kl(\cdot)$ is the Kullback–Leibler divergence between two distributions. The derivation details are left out in Appendix A. Our model learns the class-conditional distribution $P(X|Y)$ by maximizing the first expectation in ELBO, which is equivalent to minimizing the reconstruction loss [39]. By learning $P(X)$, the inference network $q_{\phi_1}(Y|X)$ has to select a suitable parameter ϕ^* which samples the y and z to minimize the reconstruction loss $\mathbb{E}_{(z,y)\sim q_\phi(Z,Y|\boldsymbol{x})}\left[\log p_{\theta_1}(x|y,z)\right]$. When the dimension of Z is chosen to be much smaller than the dimension of X, to obtain a smaller reconstruction error, the decoder has to utilize the information provided by Y, and force the value of Y to be useful for prediction. Furthermore, we constrain the Y to be a one-hot vector, then Y could be a cluster id to which the manifold of the X belongs.

So far, the latent variable Y can be inferred as a cluster id instead of a clean class id. To further link the clusters to clean labels, a naive approach is to select some reliable examples and keep the cluster numbers to be consistent with the noisy labels on these examples. In such a way, the latent representation Z and clean label Y can be effectively inferred, and therefore, it encourages the identifiability of the transition relationship $p_{\theta_2}(\tilde{Y}|Y,X)$. To achieve this, instead of explicitly selecting the reliable example in advance, our method is trained in an end-to-end favor, i.e., the reliable examples are selected dynamically during the update of parameters of our model by using the co-teaching technique [25]. The advantage of this approach is that the selection bias of the reliable example [11] can be greatly reduced. Intuitively, the accurately selected reliable examples can encourage the identifiability of $p_{\theta_2}(\tilde{Y}|Y,X)$ and $p_{\theta_1}(X|Y,Z)$, and the accurately estimated $p_{\theta_2}(\tilde{Y}|Y,X)$ and $p_{\theta_1}(X|Y,Z)$ will encourage the network

Algorithm 4 CausalNL

Input: A noisy sample S, Average noise rate ρ, Total epoch T_{max}, Batch size N .

1: **For** T $= 1, \ldots, T_{max}$:
2: **For** mini-batch $\bar{S} = \{x_i\}_{i=0}^{N}, \tilde{L} = \{\tilde{y}_i\}_{i=0}^{N}$ in S:
3: Feed $\bar{S}$ to encoders $\hat{q}_{\phi_1^1}$ and $\hat{q}_{\phi_1^2}$ to get clean label sets L_1 and L_2, respectively;
4: Feed $(\bar{S}, L_1)$ to encoder $\hat{q}_{\phi_2^1}$ to get a representation set H_1, feed $(\bar{S}, L_2)$ to $\hat{q}_{\phi_2^2}$ to get H_2;
5: Update $\hat{q}_{\phi_2^1}$ and $\hat{q}_{\phi_2^2}$ with co-teaching loss;
6: Feed (L_1, H_1) to decoder $\hat{p}_{\theta_1^1}$ to get reconstructed dataset $\bar{S}_1$, feed (L_2, H_2) to $\hat{p}_{\theta_1^2}$ to get $\bar{S}_2$;
7: Feed $(\bar{S}_1, L_1)$ to decoder $\hat{p}_{\theta_2^1}$ to get predicted noisy labels $\tilde{L}_1$, feed $(\bar{S}_2, L_2)$ to $\hat{p}_{\theta_2^2}$ to get $\tilde{L}_2$;
8: Update networks $\hat{q}_{\phi_1^1}$, $\hat{q}_{\phi_2^1}$, $\hat{p}_{\theta_1^1}$ and $\hat{p}_{\theta_2^1}$ by calculating ELBO on $(\bar{S}, \bar{S}_1, \tilde{L}, \tilde{L}_1)$, update networks $\hat{q}_{\phi_1^2}$, $\hat{q}_{\phi_2^2}$, $\hat{p}_{\theta_1^2}$ and $\hat{p}_{\theta_2^2}$ by calculating ELBO on $(\bar{S}, \bar{S}_2, \tilde{L}, \tilde{L}_2)$;

Output: The inference network $\hat{q}_{\phi_1^1}$.

to select more reliable examples.

6.3.2 Practical Implementation

Our method is summarized in Algorithm 4 and illustrated in Fig. 6.3, Here we introduce the structure of our model and loss functions.

Model Structure Because we incorporate co-teaching in our model training, we need to add a copy of the decoders and encoders in our method. As the two branches share the same architectures, we first present the details of the first branch and then briefly introduce the second branch.

For the first branch, we need a set of encoders and decoders to model the distributions in Eq. (6.3) and Eq. (6.5). Specifically, we have two encoder networks

$$Y_1 = \hat{q}_{\phi_1^1}(X), \quad Z_1 \sim \hat{q}_{\phi_2^1}(X, Y_1)$$

for Eq. (6.5) and two decoder networks

$$X_1 = \hat{p}_{\theta_1^1}(Y_1, Z_1), \quad \tilde{Y}_1 = \hat{p}_{\theta_2^1}(X_1, Y_1)$$

for Eq. (6.3). The first encoder $\hat{q}_{\phi_1^1}(X)$ takes an instance X as input $\hat{q}_{\phi_1^1}(X)$ and output a predicted clean label Y_1. The second encoder $\hat{q}_{\phi_2^1}(X, Y_1)$ takes both the instance X and the generated label Y_1 as input and outputs a latent feature Z_1. Then the generated Y_1 and Z_1 are passed to the decoder $\hat{p}_{\theta_1^1}(Y_1, Z_1)$ which will generate a reconstructed image X_1. Finally, the generated X_1 and Y_1 will be the input for another decoder $\hat{p}_{\theta_2^1}(X_1, Y_1)$ which returns predicted noisy labels $\tilde{Y}_1$. It is worth mentioning that the reparameterization trick [39] is used for sampling, which allows backpropagation in $\hat{q}_{\phi_2^1}(X, Y_1)$.

Similarly, the encoder and decoder networks in the second branch are defined as follows

$$Y_2 = \hat{q}_{\phi_1^2}(X), \quad Z_2 \sim \hat{q}_{\phi_2^2}(X, Y_2), \quad X_2 = \hat{p}_{\theta_1^2}(Y_2, Z_2), \quad \tilde{Y}_2 = \hat{p}_{\theta_2^2}(X_2, Y_2).$$

During training, we let two encoders $\hat{q}_{\phi_1^1}(X)$ and $\hat{q}_{\phi_1^2}(X)$ teach each other given every mini-batch.

Loss Functions We divide the loss functions into two parts. The first part is the negative ELBO in Eq. (D.4), and the second part is a co-teaching loss. The detailed formulation will be left in Appendix D.

For the negative ELBO, the first term $-\mathbb{E}_{(z,y)\sim q_\phi(Z,Y|\boldsymbol{x})}\left[\log p_{\theta_1}(x|y,z)\right]$ is a reconstruction loss, and we use the $\ell 1$ loss for reconstruction. The second term is $-\mathbb{E}_{y\sim q_{\phi_1}(Y|\boldsymbol{x})}\left[\log p_{\theta_2}(\tilde{y}|y,x)\right]$, which aims to learn noisy labels given inference y and x, this can be simply replaced by using cross-entropy loss on outputs of both decoders $\hat{p}_{\theta_2^1}(X_1, Y_1)$ and $\hat{p}_{\theta_2^2}(X_2, Y_2)$ with the noisy labels contained in the training data. The additional two terms are two regularizers. To calculate $kl(q_{\phi_1}(Y|\boldsymbol{x})\|p(Y))$, we assume that the prior $P(Y)$ is a uniform distribution. Then minimizing $kl(q_{\phi_1}(Y|\boldsymbol{x})\|p(Y))$ is equivalent to maximizing the entropy of $q_{\phi_1}(Y|\boldsymbol{x})$ for each instance x, i.e., $-\sum_y q_{\phi_1}(y|\boldsymbol{x}) \log q_{\phi_1}(y|\boldsymbol{x})$. The benefit of having this term is that it could reduce the overfitting problem of the inference network.

For $\mathbb{E}_{y\sim q_{\phi_1}(Y|\boldsymbol{x})}[kl(q_{\phi_2}(Z|y,x)\|p(Z))]$, we let $p(Z)$ to be a standard multivariate Gaussian distribution. Since, empirically, $q_{\phi_2}(Z|y,x)$ is the encoders $\hat{q}_{\phi_1^1}(X)$ and $\hat{q}_{\phi_1^2}(X)$, and the two encoders are designed to be deterministic mappings. Therefore, the expectation can be removed, and only the kl term $kl(q_{\phi_2}(Z|y,x)\|p(Z))$ is left. When $p(Z)$ is a Gaussian distribution, the kl term could have a closed form solution [39], i.e., $-\frac{1}{2}\sum_{j=1}^{J}(1+\log((\sigma_j)^2)-(\mu_j)^2-(\sigma_j)^2)$, where J is the dimension of a latent representation z, σ_j and μ_j are the encoder outputs.

For the co-teaching loss, we follow the work of Han et al. [25]. Intuitively, two encoders $\hat{q}_{\phi_1^1}(X)$ and $\hat{q}_{\phi_1^2}(X)$ feed forward all data and select some data of possibly clean labels. Then, two networks communicate with each other to select possible clean data in this mini-batch and use them for training. Finally, each encoder backpropagates over the data selected by its peer network and updates itself by cross-entropy loss.

6.4 Experiments

In this section, we compare the classification accuracy of the proposed method with the popular label-noise learning algorithms [50, 64, 36, 25, 92, 102, 57] on both synthetic and real-world datasets.

6.4.1 Experimental Setup

Datasets We verify the efficacy of our approach on the manually corrupted version of four datasets, i.e., *FashionMNIST* [93], *SVHN* [60], *CIFAR10*, *CIFAR100* [41], and one real-world noisy dataset, i.e., *Clothing1M* [94]. *FashionMNIST* contains 60,000 training images and 10,000 test images with 10 classes; *SVHN* contains 73,257 training images and 26,032 test images with 10 classes. *CIFAR10* contains 50,000 training images and 10,000 test images. *CIFAR10* and *CIFAR100* both contain 50,000 training images and 10,000 test images but the former has 10 classes of images, and the latter has 10 classes of images. The four datasets contain clean data. We add instance-dependent label noise to the training sets manually according to Xia et al. [91]. *Clothing1M* has 1M images with real-world noisy

labels and 10k images with clean labels for testing. For all the synthetic noisy datasets, the experiments have been repeated 5 times.

Network structure and optimization For a fair comparison, all experiments are conducted on NVIDIA Tesla V100, and all methods are implemented by PyTorch. Dimension of the latent representation Z is set to 25 for all synthetic noisy datasets. For encoder networks $\hat{q}_{\phi_1^1}(X)$ and $\hat{q}_{\phi_1^2}(X)$, we use the same network structures with baseline method. Specially, we use a ResNet-18 network for *FashionMNIST*, a ResNet-34 network for *SVHN* and *CIFAR10*. a ResNet-50 network for *CIFAR100* without pretraining. We use the same number of hidden layers and feature maps. Specifically, 1). we model $q_{\phi_2}(Z|Y,X)$ and $p_{\theta_2}(\tilde{Y}|Y,X)$ by two 4-hidden-layer convolutional networks, and the corresponding feature maps are 32, 64, 128 and 256; 2). we model $p_{\theta_1}(X|Y,Z)$ by a 4-hidden-layer transposed-convolutional network, and the corresponding feature maps are 256, 128, 64 and 32. We ran 150 epochs for each experiment on these datasets. For *Clothing1M*, we use ResNet-50 networks pre-trained on ImageNet. We use random crop and horizontal flips for data augmentation. For Clothing1M, we use a ResNet-50 network pre-trained on ImageNet, and the clean training data is not used. Dimension of the latent representation Z is set to 100. We model $q_{\phi_2}(Z|Y,X)$ and $p_{\theta_2}(\tilde{Y}|Y,X)$ by two 5-hidden-layer convolutional networks, and the corresponding feature maps are 32, 64, 128, 256, 512. We model $p_{\theta_1}(X|Y,Z)$ by a 5-hidden-layer transposed-convolutional network, and the corresponding feature maps are 512, 256, 128, 64 and 32. We ran 40 epochs on *Clothing1M*.

Baselines and measurements We compare the proposed method with the following state-of-the-art approaches: (i). CE, which trains the standard deep network with the cross entropy loss on noisy datasets. (ii). Decoupling [57], which trains two networks on samples whose predictions from the two networks are different. (iii). MentorNet [36], Co-teaching [25], which mainly handles noisy labels by training on instances with small loss values. (iv). Forward [64], Reweight [50], and T-Revision [92]. These approaches utilize a class-dependent transition matrix T to correct the loss function. We report average test accuracy on over the last ten epochs of each model on the clean test set. Higher classification accuracy means that

	IDN-20%	IDN-30%	IDN-40%	IDN-45%	IDN-50%
CE	88.54±0.32	88.38±0.42	84.22±0.35	69.72±0.72	52.32±0.68
Co-teaching	91.21±0.31	90.30±0.42	89.10±0.29	86.78±0.90	63.22±1.56
Decoupling	90.70±0.28	90.34±0.36	88.78±0.44	87.54±0.53	68.32±1.77
MentorNet	91.57±0.29	90.52±0.41	88.14±0.76	85.12±0.76	61.62±1.42
Mixup	88.68±0.37	88.02±0.37	85.47±0.55	79.57±0.75	66.02±2.58
Forward	90.05±0.43	88.65±0.43	86.27±0.48	73.35±1.03	58.23±3.14
Reweight	90.27±0.27	89.58±0.37	87.04±0.32	80.69±0.89	64.13±1.23
T-Revision	**91.58**±0.31	90.11±0.61	89.46±0.42	84.01±1.14	68.99±1.04
CausalNL	90.84±0.31	**90.68**±0.37	**90.01**±0.45	**88.75**±0.81	**78.19**±1.01

Table 6.1: Means and standard deviations (percentage) of classification accuracy on *FashionMNIST* with different label noise levels.

	IDN-20%	IDN-30%	IDN-40%	IDN-45%	IDN-50%
CE	91.51±0.45	91.21±0.43	87.87±1.12	67.15±1.65	51.01±3.62
Co-teaching	93.93±0.31	92.06±0.31	91.93±0.81	89.33±0.71	67.62±1.99
Decoupling	90.02±0.25	91.59±0.25	88.27±0.42	84.57±0.89	65.14±2.79
MentorNet	**94.08**±0.12	92.73±0.37	90.41±0.49	87.45±0.75	61.23±2.82
Mixup	89.73±0.37	90.02±0.35	85.47±0.55	82.41±0.62	68.95±2.58
Forward	91.89±0.31	91.59±0.23	89.33±0.53	80.15±1.91	62.53±3.35
Reweight	92.44±0.34	92.32±0.51	91.31±0.67	85.93±0.84	64.13±3.75
T-Revision	93.14±0.53	93.51±0.74	92.65±0.76	88.54±1.58	64.51±3.42
CausalNL	94.06±0.23	**93.86**±0.37	**93.82**±0.45	**93.19**±0.81	**85.41**±2.95

Table 6.2: Means and standard deviations (percentage) of classification accuracy on *CIFAR10* with different label noise levels.

	IDN-20%	IDN-30%	IDN-40%	IDN-45%	IDN-50%
CE	75.81±0.26	69.15±0.65	62.45±0.86	51.72±1.34	39.42±2.52
Co-teaching	80.96±0.31	78.56±0.61	73.41±0.78	71.60±0.79	45.92±2.21
Decoupling	78.71±0.15	75.17±0.58	61.73±0.34	58.61±1.73	50.43±2.19
MentorNet	81.03±0.12	77.22±0.47	71.83±0.49	66.18±0.64	47.89±2.03
Mixup	73.17±0.37	70.02±0.31	61.56±0.71	56.45±0.62	48.95±2.58
Forward	74.64±0.32	69.75±0.56	60.21±0.75	48.81±2.59	46.27±1.30
Reweight	76.23±0.25	70.12±0.72	62.58±0.46	51.54±0.92	45.46±2.56
T-Revision	76.15±0.37	70.36±0.61	64.09±0.37	52.42±1.01	49.02±2.13
CausalNL	**81.47**±0.32	**80.38**±0.37	**77.53**±0.45	**78.60**±0.93	**77.39**±1.24

Table 6.3: Means and standard deviations (percentage) of classification accuracy on *CIFAR100* with different label noise levels.

the algorithm is more robust to the label noise.

	IDN-20%	IDN-30%	IDN-40%	IDN-45%	IDN-50%
CE	30.42±0.44	24.15±0.78	21.45±0.70	15.23±1.32	14.42±2.21
Co-teaching	37.96±0.53	33.43±0.74	28.04±1.43	25.60±0.93	23.97±1.91
Decoupling	36.53±0.49	30.93±0.88	27.85±0.91	23.81±1.31	19.59±2.12
MentorNet	38.91±0.54	34.23±0.73	31.89±1.19	27.53±1.23	24.15±2.31
Mixup	32.92±0.76	29.76±0.87	25.92±1.26	23.13±2.15	21.31±1.32
Forward	36.38±0.92	33.17±0.73	26.75±0.93	21.93±1.29	19.27±2.11
Reweight	36.73±0.72	31.91±0.91	28.39±1.46	24.12±1.41	20.23±1.23
T-Revision	37.24±0.85	36.54±0.79	27.23±1.13	25.53±1.94	22.54±1.95
CausalNL	**41.47**±0.32	**40.98**±0.62	**34.02**±0.95	**33.34**±1.13	**32.129**±2.23

Table 6.4: Means and standard deviations (percentage) of classification accuracy on *CIFAR100* with different label noise levels.

CE	Decoupling	MentorNet	Co-teaching
68.88	54.53	56.79	60.15
Forward	Reweight	T-Revision	caualNL
69.91	70.40	70.97	**72.24**

Table 6.5: Classification accuracy on *Clothing1M*. In the experiments, only noisy samples are exploited to train and validate the deep model.

6.4.2 Classification accuracy Evaluation

Results on synthetic noisy datasets Tables 6.1, 6.2, 6.3, and 6.4 report the classification accuracy on the datasets of *F-MNIST*, *SVHN*, *CIFAR-10*, and *CIFAR100*, respectively. The synthetic experiments reveal that our method is powerful in handling instance-dependent label noise, particularly in situations with high noise rates. For all datasets, the classification accuracy does not drop too much compared with all baselines, and the advantages of our proposed method increase with the increase of the noise rate. Additionally, it shows that for all these datasets Y should be a cause of X, and therefore, the classification accuracy by using our method can be improved.

For noisy *F-MNIST*, *SVHN* and *CIFAR-10*, in the easy case IDN-20%, almost all methods work well. When the noise rate is 30%, the advantages of causalNL begin to show. We surpassed all methods. When the noise rate raises, all the baselines are gradually defeated. Finally, in the hardest case, i.e., IDN-50%, the superiority of causalNL widens the gap of performance.

The classification accuracy of causalNL is at least 10% higher than the best baseline method. For noisy *CIFAR-100*, all the methods do not work well. However, causalNL still overtakes the other methods with clear gaps for all different levels of noise rate.

Results on the real-world noisy dataset On the real-world noisy dataset *Clothing1M*, our method causalNL outperforms all the baselines as shown in Table 6.5. The experimental results also show that the noise type in *Clothing1M* is more likely to be instance-dependent label noise, and making the instance-independent assumption on the transition matrix sometimes can be strong.

6.5 Summary

In this chapter, we have investigated how to use $P(X)$ to help learn instance-dependent label noise. Specifically, the previous assumptions are made on the transition matrix, and the assumptions are hard to be verified and might be violated on real-world datasets. Inspired by a causal perspective, when Y is a cause of X, then $P(X)$ should contain useful information to infer the clean label Y. We propose a novel generative approach called causalNL for instance-dependent label-noise learning. Our model makes use of the causal graph to contribute to the identifiability of the transition matrix, and therefore help learn clean labels. In order to learn $P(X)$, compared to the previous methods, our method contains more parameters. But the experiments on both synthetic and real-world noisy datasets show that a little bit of sacrifice on computational efficiency is worth it, i.e., the classification accuracy of casualNL significantly outperforms all the state-of-the-art methods. Additionally, the results also tell us that in classification problems, Y can usually be considered as a cause of X, and suggest that the understanding and modeling of the data generative process can help leverage additional information that is useful in solving advanced machine learning problems concerning the relationship between different modules of the data joint distribution. In our future work, we will study the theoretical properties of our method and establish the identifiability result under certain assumptions on the data-generative process.

Chapter 7

Conclusion

In this thesis, we have theoretically and empirically shown that the transition matrix plays a crucial role in learning with noisy labels. We have found that the transition matrix not only helps design statistical-consistent methods but also can be leveraged to further improve the performance of SOTA heuristic-based methods. We have also found that the transition matrix can be employed to improve the robustness of learning models on a wide range of datasets that have different data generative processes. As the transition matrix is usually unknown and hard to estimate, we have also proposed two new transition-matrix estimation methods that can actually estimate the instance-independent transition matrix and the instance-dependent transition matrix, respectively. We have also conducted extensive experiments on both synthetic and real-world datasets which demonstrate the effectiveness of our transition-matrix estimation methods.

Appendix A

Poofs in Chapter 3

In this section, we show all the proofs in Chapter 3.

A.1 Proof of Theorem 3.3.1

Proof.

$$
\begin{aligned}
&P(\tilde{Y}=0|\boldsymbol{x}_2)-P(\tilde{Y}=1|\boldsymbol{x}_1)\\
=&P(\tilde{Y}=0|Y=0)P(Y=0|\boldsymbol{x}_2)+P(\tilde{Y}=0|Y=1)P(Y=1|\boldsymbol{x}_2)\\
&-[P(\tilde{Y}=1|Y=0)P(Y=0|\boldsymbol{x}_1)+P(\tilde{Y}=1|Y=1)P(Y=1|\boldsymbol{x}_1)]\\
=&(1-P(\tilde{Y}=1|Y=0))P(Y=0|\boldsymbol{x}_2)+P(\tilde{Y}=0|Y=1)P(Y=1|\boldsymbol{x}_2)\\
&-[P(\tilde{Y}=1|Y=0)P(Y=0|\boldsymbol{x}_1)+(1-P(\tilde{Y}=0|Y=1))P(Y=1|\boldsymbol{x}_1)]\\
=&(1-P(\tilde{Y}=1|Y=0))P(Y=0|\boldsymbol{x}_2)+P(\tilde{Y}=0|Y=1)P(Y=1|\boldsymbol{x}_2)\\
&-[P(\tilde{Y}=1|Y=0)P(Y=0|\boldsymbol{x}_1)+(1-P(\tilde{Y}=0|Y=1))P(Y=1|\boldsymbol{x}_1)]\\
=&P(Y=0|\boldsymbol{x}_2)-P(\tilde{Y}=1|Y=0)P(Y=0|\boldsymbol{x}_2)+P(\tilde{Y}=0|Y=1)P(Y=1|\boldsymbol{x}_2)\\
&-[P(\tilde{Y}=1|Y=0)P(Y=0|\boldsymbol{x}_1)+P(Y=1|\boldsymbol{x}_1)-P(\tilde{Y}=0|Y=1)P(Y=1|\boldsymbol{x}_1)]\\
=&P(Y=1|\boldsymbol{x}_1)-P(\tilde{Y}=1|Y=0)P(Y=1|\boldsymbol{x}_1)+P(\tilde{Y}=0|Y=1)(1-P(Y=1|\boldsymbol{x}_1))\\
&-[P(\tilde{Y}=1|Y=0)(1-P(Y=1|\boldsymbol{x}_1))+P(Y=1|\boldsymbol{x}_1)-P(\tilde{Y}=0|Y=1)P(Y=1|\boldsymbol{x}_1)]\\
=&P(Y=1|\boldsymbol{x}_1)-P(\tilde{Y}=1|Y=0)P(Y=1|\boldsymbol{x}_1)\\
&+P(\tilde{Y}=0|Y=1)-P(\tilde{Y}=0|Y=1)P(Y=1|\boldsymbol{x}_1)\\
&-[P(\tilde{Y}=1|Y=0)-P(\tilde{Y}=1|Y=0)P(Y=1|\boldsymbol{x}_1)\\
&+P(Y=1|\boldsymbol{x}_1)-P(\tilde{Y}=0|Y=1)P(Y=1|\boldsymbol{x}_1)]\\
=&P(\tilde{Y}=0|Y=1)-P(\tilde{Y}=1|Y=0)<0. \qquad \text{(A.1)}
\end{aligned}
$$

Note that f^* is an optimal hypothesis which perfectly learns the noisy class posterior distribution. By employing the cross-entropy loss on f^*, we have

$$\ell(f^*(X), \tilde{Y}) = -\tilde{Y}\log(f^*(X)) - (1-\tilde{Y})\log(1-f^*(X)) = -\log(P(\tilde{Y}|X)), \tag{A.2}$$

which is a non-increasing function. Therefore, the largest noisy class posterior has the minimum loss. Because $\arg\max_{j\in\{0,1\}} P(\tilde{Y}=j|\boldsymbol{x}_2)=0$, $\arg\max_{i\in\{0,1\}} P(\tilde{Y}=i|\boldsymbol{x}_1)=1$, and $P(\tilde{Y}=0|\boldsymbol{x}_2) > P(\tilde{Y}=1|\boldsymbol{x}_1)$ by Eq. (A.1), then

$$\max(P(\tilde{Y}=0|\boldsymbol{x}_2), P(\tilde{Y}=1|\boldsymbol{x}_2), P(\tilde{Y}=0|\boldsymbol{x}_1), P(\tilde{Y}=1|\boldsymbol{x}_1)) = P(\tilde{Y}=1|\boldsymbol{x}_1),$$

which implies that the minimum loss among those four noisy class posteriors is $\ell(f^*(X=\boldsymbol{x}_1), \tilde{Y}=1)$. Therefore $\min_{i\in\{0,1\}} \ell(f^*(\boldsymbol{x}_2), i) > \min_{i\in\{0,1\}} \ell(f^*(\boldsymbol{x}_1), i)$ holds, which completes the proof. □

A.2 Proof of Theorem 3.3.2

Proof.

$$\begin{aligned}
&P(\tilde{Y}=0|\boldsymbol{x}_1) - P(\tilde{Y}=1|\boldsymbol{x}_1)\\
=&P(\tilde{Y}=0|Y=0)P(Y=0|\boldsymbol{x}_1) + P(\tilde{Y}=0|Y=1)P(Y=1|\boldsymbol{x}_1)\\
-&[P(\tilde{Y}=1|Y=0)P(Y=0|\boldsymbol{x}_1) + P(\tilde{Y}=1|Y=1)P(Y=1|\boldsymbol{x}_1)]\\
=&(1-P(\tilde{Y}=1|Y=0))P(Y=0|\boldsymbol{x}_1) + P(\tilde{Y}=0|Y=1)P(Y=1|\boldsymbol{x}_1)\\
-&[P(\tilde{Y}=1|Y=0)P(Y=0|\boldsymbol{x}_1) + (1-P(\tilde{Y}=0|Y=1))P(Y=1|\boldsymbol{x}_1)]\\
=&P(Y=0|\boldsymbol{x}_1) - P(\tilde{Y}=1|Y=0)P(Y=0|\boldsymbol{x}_1) + P(\tilde{Y}=0|Y=1)P(Y=1|\boldsymbol{x}_1)\\
-&[P(\tilde{Y}=1|Y=0)P(Y=0|\boldsymbol{x}_1) + P(Y=1|\boldsymbol{x}_1) - P(\tilde{Y}=0|Y=1)P(Y=1|\boldsymbol{x}_1)]\\
=&(1-2P(\tilde{Y}=1|Y=0))P(Y=0|\boldsymbol{x}_1) + (2P(\tilde{Y}=0|Y=1)-1)P(Y=1|\boldsymbol{x}_1).
\end{aligned} \tag{A.3}$$

Let $P(Y=0|\boldsymbol{x}_1) < \frac{(1-2P(\tilde{Y}=0|Y=1))}{(1-2P(\tilde{Y}=1|Y=0))}P(Y=1|\boldsymbol{x}_1)$, by combining with Eq. (A.3), we have

$$\begin{aligned}
&P(\tilde{Y}=0|\boldsymbol{x}_1) - P(\tilde{Y}=1|\boldsymbol{x}_1) \\
<&(1-2P(\tilde{Y}=1|Y=0))\frac{(1-2P(\tilde{Y}=0|Y=1))}{(1-2P(\tilde{Y}=1|Y=0))}P(Y=1|\boldsymbol{x}_1) \\
&+(2P(\tilde{Y}=0|Y=1)-1)P(Y=1|\boldsymbol{x}_1) \\
<&(1-2P(\tilde{Y}=0|Y=1))P(Y=1|\boldsymbol{x}_1)+(2P(\tilde{Y}=0|Y=1)-1)P(Y=1|\boldsymbol{x}_1) < 0,
\end{aligned} \tag{A.4}$$

which implies that $P(\tilde{Y}=1|\boldsymbol{x}_1) > 0.5$. Let the Bayes label on the clean class-posterior distribution of $\boldsymbol{x}_1$ be 0[1], then $0.5 < P(Y=0|\boldsymbol{x}_1) < \frac{(1-2P(\tilde{Y}=0|Y=1))}{(1-2P(\tilde{Y}=1|Y=0))}P(Y=1|\boldsymbol{x}_1)$, which completes the proof. □

[1]The Bayes label is the label with the largest class posterior. For example, the Bayes label on the clean class-posterior distribution Y^* of an instance $\boldsymbol{x}$ is defined as $Y^* = \arg\max_{i\in\{0,1\}} P(Y=i|x)$[58]

Appendix B

Poofs in Chapter 4

Appendix A

In this section, we show all the proofs in Chapter 4.

B.0.1 Poof of Theorem 4.3.1

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

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

Then similarly,

$$\begin{aligned} P(\tilde{Y} = i|Y' = j) &= \mathbb{E}_{P(X|Y'=j)}\left[\mathbb{1}_{\{\tilde{f}(X)=i\}}\right] \\ &= \int_x \mathbb{1}_{\{\tilde{f}(x)=i\}} P(X = x|Y' = j)\mathrm{d}x \\ &= \int_x \mathbb{1}_{\{\tilde{f}(x)=i\}} \frac{P(Y' = j|X = x)P(X = x)}{P(Y' = j)}\mathrm{d}x \\ &= \mathbb{E}_{P(X)}\left[\mathbb{1}_{\{\tilde{f}(X)=i\}} \frac{P(Y' = j|X)}{P(Y' = j)}\right]. \end{aligned} \tag{B.2}$$

The last equality is obtained by using the reweighting technique [50], 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.

$$
\begin{aligned}
&P(\tilde{Y} = i|Y' = j) - P(\tilde{Y} = i|Y^* = j) \\
=&\mathbb{E}_{P(X)}\left[\mathbb{1}_{\{\tilde{f}(X)=i\}}\frac{P(Y' = j|X)}{P(Y' = j)}\right] - \mathbb{E}_{P(X)}\left[\mathbb{1}_{\{\tilde{f}(X)=i\}}\frac{P(Y^* = j|X)}{P(Y^* = j)}\right] \\
=&\mathbb{E}_{P(X)}\left[\mathbb{1}_{\{\tilde{f}(X)=i\}}\left(\frac{P(Y' = j|X)}{P(Y' = j)} - \frac{P(Y^* = j|X)}{P(Y^* = j)}\right)\right] \\
=&\mathbb{E}_{P(X)}\left[\mathbb{1}_{\{\tilde{f}(X)=i\}}\frac{P(Y' = j|X)P(Y^* = j) - P(Y^* = j|X)P(Y' = j)}{P(Y' = j)P(Y^* = j)}\right] \\
=&\frac{1}{P(Y^* = j)}\mathbb{E}_{P(X)}\left[\mathbb{1}_{\{\tilde{f}(X)=i\}}\frac{P(Y' = j|X)P(Y^* = j) - P(Y^* = j|X)P(Y' = j)}{P(Y' = j)}\right] \\
=&\frac{1}{P(Y^* = j)}\mathbb{E}_{P(X)}\left[\mathbb{1}_{\{\tilde{f}(X)=i\}}\left(P(Y' = j|X)\frac{P(Y^* = j)}{P(Y' = j)} - P(Y^* = j|X)\right)\right]
\end{aligned}
\tag{B.3}
$$

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. □

Appendix C

Poofs in Chapter 5

C.1 Proof of Theorem 5.3.1

Proof. According to Eq. (1) in the main paper, the estimation error for the T-estimator is

$$\epsilon_T = \sum_{i,j} \left| T_{ij} - \hat{T}_{ij} \right| = \sum_{i,j} \left| P(\bar{Y} = j | X = \boldsymbol{x}^i) - \hat{P}(\bar{Y} = j | X = \boldsymbol{x}^i) \right|. \quad \text{(C.1)}$$

As we have assumed, for all instance $\boldsymbol{x} \in \mathcal{X}$, for all $j \in \{1, \ldots, C\}$,

$$\left| P(\bar{Y} = j | X = \boldsymbol{x}) - \hat{P}(\bar{Y} = j | X = \boldsymbol{x}) \right| = \Delta_1. \quad \text{(C.2)}$$

Then, we have

$$\epsilon_T = C^2 \Delta_1. \quad \text{(C.3)}$$

The estimation error for the i, j-the entry of the dual T-estimator is

$$\begin{aligned} &\Bigg| \sum_l P(\bar{Y} = j | Y' = l, Y = i) P(Y' = l | Y = i) \\ &- \sum_l \hat{P}(\bar{Y} = j | Y' = l) P(Y' = l | Y = i) \Bigg| \quad \text{(C.4)} \\ &= \sum_l \left| P(\bar{Y} = j | Y' = l, Y = i) - \hat{P}(\bar{Y} = j | Y' = l) \right| P(Y' = l | Y = i), \end{aligned}$$

where the first equation holds because there is no estimation error for the transition matrix denoting the transition from the clean class to the intermediate class (as we have discussed in Section 3.1). The estimation error for the dual T-estimator comes from the estimation error for fitting the noisy class labels (to eliminate the dependence on the clean label) and the estimation error for $P(\bar{Y}=j|Y'=l)$ by counting discrete labels.

We have assumed that the estimation error for $P(\bar{Y}=j|Y'=l)$ is Δ_2, i.e., $|P(\bar{Y}=j|Y'=l)-\hat{P}(\bar{Y}=j|Y'=l)|=\Delta_2$ and that the estimation error for fitting the noisy class labels is Δ_3, i.e., $\forall \boldsymbol{x}\in\mathcal{X}, P(Y'=\bar{Y}|\boldsymbol{x})=1-\Delta_3$. Note that, to eliminate the dependence on the clean label for $T^{\spadesuit}$, we need to achieve $P(Y'=\bar{Y}|\boldsymbol{x})=1$ for all $\boldsymbol{x}\in\mathcal{X}$. The error Δ_3 will be introduced if there is an error in fitting the noisy class labels. We have that $P(Y'\neq\bar{Y}|\boldsymbol{x})=\Delta_3$.

We have

$$
\begin{aligned}
&\Big|P(\bar{Y}=j|Y'=l,Y=i)-\hat{P}(\bar{Y}=j|Y'=l)\Big| \\
=&\Big|P(\bar{Y}=j|Y'=l,Y=i,\boldsymbol{x})-\hat{P}(\bar{Y}=j|Y'=l,\boldsymbol{x})\Big| \\
=&\Big|P(\bar{Y}=j|Y'=l,Y=i,\boldsymbol{x})P(Y'=\bar{Y}|\boldsymbol{x})+P(\bar{Y}=j|Y'=l,Y=i,\boldsymbol{x})P(Y'\neq\bar{Y}|\boldsymbol{x}) \\
&-\hat{P}(\bar{Y}=j|Y'=l,\boldsymbol{x})P(Y'=\bar{Y}|\boldsymbol{x})-\hat{P}(\bar{Y}=j|Y'=l,\boldsymbol{x})P(Y'\neq\bar{Y}|\boldsymbol{x})\Big| \\
=&\Big|P(\bar{Y}=j|Y'=l,\boldsymbol{x})P(Y'=\bar{Y}|\boldsymbol{x})+P(\bar{Y}=j|Y'=l,Y=i,\boldsymbol{x})P(Y'\neq\bar{Y}|\boldsymbol{x}) \\
&-\hat{P}(\bar{Y}=j|Y'=l,\boldsymbol{x})P(Y'=\bar{Y}|\boldsymbol{x})-\hat{P}(\bar{Y}=j|Y'=l,\boldsymbol{x})P(Y'\neq\bar{Y}|\boldsymbol{x})\Big| \\
\leq&\Big|P(\bar{Y}=j|Y'=l)-\hat{P}(\bar{Y}=j|Y'=l)\Big|\,P(Y'=\bar{Y}|\boldsymbol{x}) \\
&+\Big|P(\bar{Y}=j|Y'=l,Y=i)-\hat{P}(\bar{Y}=j|Y'=l)\Big|\,P(Y'\neq\bar{Y}|\boldsymbol{x}) \\
\leq&\Delta_2(1-\Delta_3)+\Delta_3<\Delta_2+\Delta_3,
\end{aligned}
\tag{C.5}
$$

where the second equation holds because the transition matrices are independent of instances. Hence, the estimation error of $T^{\spadesuit}$ is

$$\begin{aligned}\epsilon_{DT} &= \sum_{i,j,l} \left| P(\bar{Y}=j|Y'=l,Y=i) - \hat{P}(\bar{Y}=j|Y'=l) \right| P(Y'=l|Y=i) \\ &< \sum_{i,j} \sum_{l} (\Delta_2 + \Delta_3) P(Y'=l|Y=i) \\ &= \sum_{i,j} (\Delta_2 + \Delta_3) = C^2(\Delta_2 + \Delta_3). \end{aligned} \tag{C.6}$$

Therefore, under Assumption 5.3.1 in the main paper, the estimation error ϵ_{DT} of the dual T-estimator is smaller than the estimation error ϵ_T the T-estimator. □

Appendix D

Poofs in Chapter 6

D.1 Derivation Details of evidence lower-bound (ELBO)

In this section, we show the derivation details of ELBO$(x, \tilde{y})$.
Recall that the causal decomposition of the instance-dependent label noise is

$$P(X, \tilde{Y}, Y, Z) = P(Y)P(Z)P(X|Y, Z)P(\tilde{Y}|Y, X). \tag{D.1}$$

Our encoders model following distributions

$$q_\phi(Z, Y|X) = q_{\phi_2}(Z|Y, X)q_{\phi_1}(Y|X), \tag{D.2}$$

and decoders model the following distributions

$$p_\theta(X, \tilde{Y}|Y, Z) = p_{\theta_1}(X|Y, Z)p_{\theta_2}(\tilde{Y}|Y, X). \tag{D.3}$$

Now, we start with maximizing the log-likelihood $p_\theta(x, \tilde{y})$ of each datapoint $(x, \tilde{y})$.

$$\begin{aligned}
\log p_\theta(x,\tilde{y}) &= \log \int_z \int_y p_\theta(x,\tilde{y},z,y)\mathrm{d}y\mathrm{d}z \\
&= \log \int_z \int_y p_\theta(x,\tilde{y},z,y)\frac{q_\phi(z,y|x)}{q_\phi(z,y|x)}\mathrm{d}y\mathrm{d}z \\
&= \log \mathbb{E}_{(z,y)\sim q_\phi(Z,Y|x)}\left[\frac{p_\theta(x,\tilde{y},z,y)}{q_\phi(z,y|x)}\right] \\
&\geq \mathbb{E}_{(z,y)\sim q_\phi(Z,Y|x)}\left[\log\frac{p_\theta(x,\tilde{y},z,y)}{q_\phi(z,y|x)}\right] := \mathrm{ELBO}(x,\tilde{y}) \\
&= \mathbb{E}_{(z,y)\sim q_\phi(Z,Y|x)}\left[\log\frac{p(z)p(y)p_{\theta_1}(x|y,z)p_{\theta_2}(\tilde{y}|y,x))}{q_\phi(z,y|x)}\right] \\
&= \mathbb{E}_{(z,y)\sim q_\phi(Z,Y|x)}[\log\left(p_{\theta_1}(x|y,z)\right)] + \mathbb{E}_{(z,y)\sim q_\phi(Z,Y|x)}[\log\left(p_{\theta_2}(\tilde{y}|y,x)\right)] \\
&+ \mathbb{E}_{(z,y)\sim q_\phi(Z,Y|x)}\left[\log\left(\frac{p(z)p(y)}{q_{\phi_2}(z|y,x)q_{\phi_1}(y|x)}\right)\right] \qquad \text{(D.4)}
\end{aligned}$$

The $\mathrm{ELBO}(x, \tilde{y})$ above can be further simplified. Specifically,

$$\begin{aligned}
&\mathbb{E}_{(z,y)\sim q_\phi(Z,Y|x)}[\log\left(p_{\theta_2}(\tilde{y}|y,x)\right)] = \mathbb{E}_{y\sim q_{\phi_1}(Y|x)}\mathbb{E}_{z\sim q_{\phi_2}(Z|y,x)}[\log\left(p_{\theta_2}(\tilde{y}|y,x)\right)] \\
=&\mathbb{E}_{y\sim q_{\phi_1}(Y|x)}[\log\left(p_{\theta_2}(\tilde{y}|y,x)\right)], \qquad \text{(D.5)}
\end{aligned}$$

and similarly,

$$\begin{aligned}
&\mathbb{E}_{(z,y)\sim q_\phi(Z,Y|x)}\left[\log\left(\frac{p(z)p(y)}{q_{\phi_2}(z|y,x)q_{\phi_1}(y|x)}\right)\right] \\
=&\mathbb{E}_{y\sim q_{\phi_1}(Y|x)}\mathbb{E}_{z\sim q_{\phi_2}(Z|y,x)}\left[\log\left(\frac{p(z)p(y)}{q_{\phi_2}(z|y,x)q_{\phi_1}(y|x)}\right)\right] \\
=&\mathbb{E}_{y\sim q_{\phi_1}(Y|x)}\mathbb{E}_{z\sim q_{\phi_2}(Z|y,x)}\left[\log\left(\frac{p(y)}{q_{\phi_1}(y|x)}\right)\right] + \mathbb{E}_{y\sim q_{\phi_1}(Y|x)}\mathbb{E}_{z\sim q_{\phi_2}(Z|y,x)}\left[\log\left(\frac{p(z)}{q_{\phi_2}(z|y,x)}\right)\right] \\
=&\mathbb{E}_{y\sim q_{\phi_1}(Y|x)}\left[\log\left(\frac{p(y)}{q_{\phi_1}(y|x)}\right)\right] + \mathbb{E}_{y\sim q_{\phi_1}(Y|x)}\mathbb{E}_{z\sim q_{\phi_2}(Z|y,x)}\left[\log\left(\frac{p(z)}{q_{\phi_2}(z|y,x)}\right)\right] \\
=&-kl(q_{\phi_1}(Y|x)\|p(Y)) - \mathbb{E}_{y\sim q_{\phi_1}(Y|x)}\left[kl(q_{\phi_2}(Z|y,x)\|p(Z))\right], \qquad \text{(D.6)}
\end{aligned}$$

By combing Eq. (D.4), Eq. (D.5) and Eq. (D.6), we get

$$\begin{aligned}\text{ELBO}(x,\tilde{y}) = &\mathbb{E}_{(z,y)\sim q_\phi(Z,Y|x)}\left[\log p_{\theta_1}(x|y,z)\right] + \mathbb{E}_{y\sim q_{\phi_1}(Y|x)}\left[\log p_{\theta_2}(\tilde{y}|y,x)\right] \\ &- kl(q_{\phi_1}(Y|x)\|p(Y)) - \mathbb{E}_{y\sim q_{\phi_1}(Y|x)}\left[kl(q_{\phi_2}(Z|y,x)\|p(Z))\right],\end{aligned} \tag{D.7}$$

which is the ELBO in our main paper.

D.2 Loss Functions

In this section, we provide the empirical solution of the ELBO and co-teaching loss. Remind that our encoder networks and decoder networks in the first branch are defined as follows

$$Y_1 = \hat{q}_{\phi_1^1}(X), \quad Z_1 \sim \hat{q}_{\phi_2^1}(X, Y_1), \quad X_1 = \hat{p}_{\theta_1^1}(Y_1, Z_1), \quad \tilde{Y}_1 = \hat{p}_{\theta_2^1}(X_1, Y_1),$$

Let S be the noisy training set, and d^2 be the dimension of an instance x. Let y_1 and z_1 be the estimated clean label and latent representation for the instance x, respectively, by the first branch. As mentioned in our main paper (see Section 3.2), the negative ELBO loss is used to minimize 1). a reconstruction loss between each instance x and $\hat{p}_{\theta_1^1}(x, y_1)$; 2). a cross-entropy loss between noisy labels $\hat{p}_{\theta_2^1}(x_1, x_1)$ and $\tilde{y}$; 3). a cross-entropy loss between $\hat{q}_{\phi_2^1}(x, y_1)$ and uniform distribution $P(Y)$; 4). a cross-entropy loss between $\hat{q}_{\phi_2^1}(x, y_1)$ and Gaussian distribution $P(Z)$. Specifically, the empirical version of the ELBO for the first branch is as follows.

$$\begin{aligned}\sum_{(x,\tilde{y})\in S} \hat{\text{ELBO}}^1(x,\tilde{y}) = \sum_{(x,\tilde{y})\in S} \Bigg[&\beta_0 \frac{1}{d^2}\|x - \hat{p}_{\theta_1^1}(y_1, z_1)\|_1 - \beta_1 \tilde{y} \log \hat{p}_{\theta_2^1}(x_1, y_1) \\ &+\beta_2 \hat{q}_{\phi_1^1}(x) \log \hat{q}_{\phi_1^1}(x) + \beta_3 \sum_{j=1}^{J}(1 + \log((\sigma_j)^2) - (\mu_j)^2 - (\sigma_j)^2)\Bigg].\end{aligned} \tag{D.8}$$

The hyper-parameter β_0 and β_1 are set to 0.1, and β_2 is set to $1e-5$ because encouraging the distribution to be uniform on a small min-batch (i.e., 128) could have a large estimation error. The hyper-parameter β_3 is set to 0.01.

The empirical version of the ELBO for the second branch shares the same settings as the first branch.

For co-teaching loss, we directly follow Han et al. [25]. Intuitively, in each mini-batch, both encoders $\hat{q}_{\phi_1^1}(X)$ and $\hat{q}_{\phi_1^2}(X)$ trust small-loss examples, and exchange the examples to each other by a cross-entropy loss. The number of the small-loss instances used for training decays with respect to the training epoch. The experimental settings for co-teaching loss are the same as the settings in the original paper [25].

Bibliography

[1] A. Ali, S. M. Shamsuddin, and A. L. Ralescu. "Classification with class imbalance problem". In: *Int. J. Advance Soft Compu. Appl* 5.3 (2013).

[2] Y. Anzai. *Pattern recognition and machine learning*. Elsevier, 2012.

[3] E. Arazo, D. Ortego, P. Albert, N. O'Connor, and K. McGuinness. "Unsupervised label noise modeling and loss correction". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 312–321.

[4] D. Arpit, S. Jastrzebski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio, et al. "A closer look at memorization in deep networks". In: *ICML*. JMLR. org. 2017, pp. 233–242.

[5] Y. Bai, E. Yang, B. Han, Y. Yang, J. Li, Y. Mao, G. Niu, and T. Liu. "Understanding and Improving Early Stopping for Learning with Noisy Labels". In: *arXiv preprint arXiv:2106.15853* (2021).

[6] H. Bao, G. Niu, and M. Sugiyama. "Classification from Pairwise Similarity and Unlabeled Data". In: *ICML*. 2018, pp. 452–461.

[7] M. Belkin, P. Niyogi, and V. Sindhwani. "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples." In: *Journal of machine learning research* 7.11 (2006).

[8] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. "Variational inference: A review for statisticians". In: *Journal of the American statistical Association* 112.518 (2017), pp. 859–877.

[9] S. Boucheron, G. Lugosi, and P. Massart. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press, 2013.

[10] Z. Chen and L. Chan. "Causality in linear nongaussian acyclic models in the presence of latent gaussian confounders". In: *Neural Computation* 25.6 (2013), pp. 1605–1641.

[11] J. Cheng, T. Liu, K. Ramamohanarao, and D. Tao. "Learning with bounded instance and label-dependent label noise". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 1789–1799.

[12] M. Ciortan, R. Dupuis, and T. Peel. "A framework using contrastive learning for classification with noisy labels". In: *Data* 6.6 (2021), p. 61.

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

[14] I. Csiszár, P. C. Shields, et al. "Information theory and statistics: A tutorial". In: *Foundations and Trends® in Communications and Information Theory* 1.4 (2004), pp. 417–528.

[15] A. Daniely and E. Granot. "Generalization bounds for neural networks via approximate description length". In: *NeurIPS*. 2019, pp. 13008–13016.

[16] C. Elkan and K. Noto. "Learning classifiers from only positive and unlabeled data". In: *SIGKDD*. 2008, pp. 213–220.

[17] E. Englesson and H. Azizpour. "Consistency Regularization Can Improve Robustness to Label Noise". In: *arXiv preprint arXiv:2110.01242* (2021).

[18] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. "Learning object categories from internet image searches". In: *Proceedings of the IEEE* 98.8 (2010), pp. 1453–1466.

[19] D. Geiger and D. Heckerman. "Learning Gaussian networks". In: *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence* (1994).

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

[21] C. Glymour, K. Zhang, and P. Spirtes. "Review of causal discovery methods based on graphical models". In: *Frontiers in genetics* 10 (2019), p. 524.

[22] J. Goldberger and E. Ben-Reuven. "Training deep neural-networks using a noise adaptation layer". In: *ICLR*. 2017.

[23] P. Grünwald and P. Vitányi. "Handbook of the Philosophy of Information". In: ed. by J. W. Dov M. Gabbay Paul Thagard. North Holland, 2008. Chap. Algorithms information theory.

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

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

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

[27] D. Heckerman, D. Geiger, and D. Chickering. "Learning Bayesian networks: the combination of knowledge and statistical data". In: *Machine Learning* 20 (1995).

[28] B. Huang, K. Zhang, Y. Lin, B. Schölkopf, and C. Glymour. "Generalized score functions for causal discovery". In: *KDD* (2018).

[29] A. Hyvärinen and S. Smith. "Pairwise likelihood ratios for estimation of non-Gaussian structural equation models". In: *Journal of Machine Learning Research* 14 (2013), pp. 111–152.

[30] S. Imoto, T. Goto, and S. Miyano. "Estimation of genetic networks and functional structures between genes by using Bayesian networks and nonparametric regression". In: *Pacific Symposium on Biocomputing* 175-186 (2002).

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

[32] E. Jang, S. Gu, and B. Poole. "Categorical reparameterization with gumbel-softmax". In: *arXiv preprint arXiv:1611.01144* (2016).

[33] D. Janzing, J. Mooij, J. Zhang K.and Lemeire, J. Zscheischler, P. Daniusis, B. Steudel, and B. Schölkopf. "Information-geometric approach to inferring causal directions". In: *Artificial Intelligence* 182-183 (2012).

[34] D. Janzing and B. Schölkopf. "Causal inference using the algorithmic Markov condition". In: *IEEE Transactions on Information Theory* 56.10 (2010).

[35] N. Japkowicz and S. Stephen. "The class imbalance problem: A systematic study". In: *Intelligent data analysis* 6.5 (2002), pp. 429–449.

[36] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei. "MentorNet: Learning Data-Driven Curriculum for Very Deep Neural Networks on Corrupted Labels". In: *ICML*. 2018, pp. 2309–2318.

[37] R. Jonker and T. Volgenant. "Improving the Hungarian assignment algorithm". In: *Operations Research Letters* 5.4 (1986), pp. 171–175.

[38] D. Kalainathan, O. Goudet, I. Guyon, D. Lopez-Paz, and M. Sebag. "Structural agnostic modeling: Adversarial learning of causal graphs". In: *arXiv:1803.04929v3* (2020).

[39] D. P. Kingma and M. Welling. "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114* (2013).

[40] J. Kremer, F. Sha, and C. Igel. "Robust Active Label Correction". In: *AISTATS*. 2018, pp. 308–316.

[41] A. Krizhevsky, G. Hinton, et al. "Learning multiple layers of features from tiny images". In: (2009).

[42] S. Laine and T. Aila. "Temporal ensembling for semi-supervised learning". In: *arXiv preprint arXiv:1610.02242* (2016).

[43] Y. LeCun. "The MNIST database of handwritten digits". In: *http://yann. lecun. com/exdb/mnist/* (1998).

[44] Y. LeCun, C. Cortes, and C. Burges. "MNIST handwritten digit database". In: *ATT Labs [Online]. Available: http://yann. lecun. com/exdb/mnist* 2 (2010).

[45] M. S. Lewicki and T. J. Sejnowski. "Learning overcomplete representations". In: *Neural computation* 12.2 (2000), pp. 337–365.

[46] J. Li, R. Socher, and S. C. Hoi. "DivideMix: Learning with Noisy Labels as Semi-supervised Learning". In: *ICLR*. 2019.

[47] J. Li, C. Xiong, and S. C. Hoi. "Mopro: Webly supervised learning with momentum prototypes". In: *arXiv preprint arXiv:2009.07995* (2020).

[48] X. Li, T. Liu, B. Han, G. Niu, and M. Sugiyama. "Provably End-to-end Label-Noise Learning without Anchor Points". In: *arXiv preprint arXiv:2102.02400* (2021).

[49] A. Likas, N. Vlassis, and J. J. Verbeek. "The global k-means clustering algorithm". In: *Pattern recognition* 36.2 (2003), pp. 451–461.

[50] T. Liu and D. Tao. "Classification with noisy labels by importance reweighting". In: *IEEE Transactions on pattern analysis and machine intelligence* 38.3 (2016), pp. 447–461.

[51] Y. Liu. "Understanding instance-level label noise: Disparate impacts and treatments". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 6725–6735.

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

[53] N. Lu, G. Niu, A. K. Menon, and M. Sugiyama. "On the minimal supervision for training any binary classifier from only unlabeled data". In: *ICLR*. 2018.

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

[55] X. Ma, Y. Wang, M. E. Houle, S. Zhou, S. M. Erfani, S.-T. Xia, S. Wijewickrema, and J. Bailey. "Dimensionality-Driven Learning with Noisy Labels". In: *ICML*. 2018, pp. 3361–3370.

[56] D. Mahajan, R. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharambe, and L. Van Der Maaten. "Exploring the limits of weakly supervised pretraining". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 181–196.

[57] E. Malach and S. Shalev-Shwartz. "Decoupling" when to update" from" how to update"". In: *NeurIPS*. 2017, pp. 960–970.

[58] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of machine learning*. MIT press, 2018.

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

[60] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y.Ng. "Reading Digits in Natural Images with Unsupervised Feature Learning". In: *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*. 2011.

[61] D. T. Nguyen, C. K. Mummadi, T. P. N. Ngo, T. H. P. Nguyen, L. Beggel, and T. Brox. "SELF: Learning to Filter Noisy Labels with Self-Ensembling". In: *ICLR*. 2019.

[62] C. Niu, H. Shan, and G. Wang. "Spice: Semantic pseudo-labeling for image clustering". In: *arXiv preprint arXiv:2103.09382* (2021).

[63] C. G. Northcutt, T. Wu, and I. L. Chuang. "Learning with Confident Examples: Rank Pruning for Robust Classification with Noisy Labels". In: *UAI*. 2017.

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

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

[66] J. Peters, J. Mooij, D. Janzing, and B. Schölkopf. "Causal Discovery with Continuous Additive Noise Models". In: *Journal of Machine Learning Research* 15.1 (2014), pp. 2009–2053.

[67] J. Peters, D. Janzing, and B. Scholkopf. *Elements of Causal Inference*. Cambridge, Massachusetts: MIT Press, 2017.

[68] J. Peters, D. Janzing, and B. Scholkopf. "Causal inference on discrete data using additive noise models". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.12 (2011), pp. 2436–2450.

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

[70] H. Ramaswamy, C. Scott, and A. Tewari. "Mixture proportion estimation via kernel embeddings of distributions". In: *ICML*. 2016, pp. 2052–2060.

[71] S. E. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich. "Training Deep Neural Networks on Noisy Labels with Bootstrapping". In: *ICLR, Workshop Track Proceedings*. 2015.

[72] M. Ren, W. Zeng, B. Yang, and R. Urtasun. "Learning to reweight examples for robust deep learning". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 4334–4343.

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

[74] B. Schölkopf. "Causality for machine learning". In: *arXiv preprint arXiv:1911.10500* (2019).

[75] F. Schroff, A. Criminisi, and A. Zisserman. "Harvesting image databases from the web". In: *IEEE transactions on pattern analysis and machine intelligence* 33.4 (2010), pp. 754–766.

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

[77] C. Scott, G. Blanchard, and G. Handy. "Classification with asymmetric label noise: Consistency and maximal denoising". In: *COLT*. 2013, pp. 489–511.

[78] D. Sculley. "Web-scale k-means clustering". In: *Proceedings of the 19th international conference on World wide web*. 2010, pp. 1177–1178.

[79] R. D. Shah and J. Peters. "The hardness of conditional independence testing and the generalised covariance measure". In: *The Annals of Statistics* 48.3 (2020).

[80] S. Shimizu, P. O. Hoyer, A. Hyvärinen, A. Kerminen, and M. Jordan. "A linear non-Gaussian acyclic model for causal discovery." In: *Journal of Machine Learning Research* 7.10 (2006).

[81] S. Shimizu, T. Inazumi, Y. Sogawa, A. Hyvärinen, Y. Kawahara, T. Washio, P. O. Hoyer, and K. Bollen. "DirectLiNGAM: A direct method for learning a linear non-Gaussian structural equation model". In: *Journal of Machine Learning Research* 12.Apr (2011), pp. 1225–1248.

[82] K. Sohn, D. Berthelot, C.-L. Li, Z. Zhang, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang, and C. Raffel. "FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence". In: *arXiv preprint arXiv:2001.07685* 33 (2020), pp. 596–608.

[83] P. Spirtes, C. N. Glymour, R. Scheines, D. Heckerman, C. Meek, G. Cooper, and T. Richardson. *Causation, prediction, and search*. MIT press, 2000.

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

[85] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. "Dropout: a simple way to prevent neural networks from overfitting". In: *The journal of machine learning research* 15.1 (2014), pp. 1929–1958.

[86] S. Sukhbaatar, J. B. Estrach, M. Paluri, L. Bourdev, and R. Fergus. "Training convolutional networks with noisy labels". In: *ICLR*. 2015.

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

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

[89] T. Tashiro, S. Shimizu, A. Hyvärinen, and T. Washio. "ParceLiNGAM: A causal ordering method robust against latent confounders". In: *Neural computation* 26.1 (2014), pp. 57–83.

[90] H. Wei, L. Feng, X. Chen, and B. An. "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*. 2020, pp. 13726–13735.

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

[92] X. Xia, T. Liu, N. Wang, B. Han, C. Gong, G. Niu, and M. Sugiyama. "Are Anchor Points Really Indispensable in Label-Noise Learning?" In: *NeurIPS*. 2019, pp. 6835–6846.

[93] H. Xiao, K. Rasul, and R. Vollgraf. "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms". In: *arXiv preprint arXiv:1708.07747* (2017).

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

[95] Y. Xu, P. Cao, Y. Kong, and Y. Wang. "L_DMI: A Novel Information-theoretic Loss Function for Training Deep Nets Robust to Label Noise". In: *NeurIPS*. 2019, pp. 6222–6233.

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

[97] Y. Yao, T. Liu, M. Gong, B. Han, G. Niu, and K. Zhang. "Instance-dependent Label-noise Learning under a Structural Causal Model". In: vol. 34. 2021.

[98] Y. Yao, T. Liu, B. Han, M. Gong, J. Deng, G. Niu, and M. Sugiyama. "Dual T: Reducing Estimation Error for Transition Matrix in Label-noise Learning". In: *NeurIPS*. 2020.

[99] X. Yu, B. Han, J. Yao, G. Niu, I. Tsang, and M. Sugiyama. "How does Disagreement Help Generalization against Label Corruption?" In: *ICML*. 2019, pp. 7164–7173.

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

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

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

[103] K. Zhang and A. Hyvarinen. "On the Identifiability of the Post-Nonlinear Causal Model". In: *Conference on Uncertainty in Artificial Intelligence*. 2009.

[104] K. Zhang, J. Zhang, and B. Schölkopf. "Distinguishing cause from effect based on exogeneity". In: *arXiv preprint arXiv:1504.05651* (2015).

[105] Y. Zhang, G. Niu, and M. Sugiyama. "Learning Noise Transition Matrix from Only Noisy Labels via Total Variation Regularization". In: *arXiv preprint arXiv:2102.02414* (2021).

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

[107] E. Zheltonozhskii, C. Baskin, A. Mendelson, A. M. Bronstein, and O. Litany. "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*. 2022, pp. 1657–1667.

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