Learning Noise Transition Matrix from Only Noisy Labels via Total Variation Regularization

Yivan Zhang [1 2] **Gang Niu** [2] **Masashi Sugiyama** [2 1]

Abstract

Many weakly supervised classification methods employ a *noise transition matrix* to capture the class-conditional label corruption. To estimate the transition matrix from noisy data, existing methods often need to estimate the noisy class-posterior, which could be unreliable due to the overconfidence of neural networks. In this work, we propose a theoretically grounded method that can estimate the noise transition matrix and learn a classifier simultaneously, without relying on the error-prone noisy class-posterior estimation. Concretely, inspired by the characteristics of the stochastic label corruption process, we propose *total variation regularization*, which encourages the predicted probabilities to be more distinguishable from each other. Under mild assumptions, the proposed method yields a *consistent estimator* of the transition matrix. We show the effectiveness of the proposed method through experiments on benchmark and real-world datasets.

1. Introduction

Can we learn a correct classifier based on possibly incorrect examples? The study of classification in the presence of *label noise* has been of interest for decades (Angluin & Laird, 1988) and is becoming more important in the era of deep learning (Goodfellow et al., 2016). This issue can be caused by the use of imperfect surrogates of clean labels produced by annotation techniques for large-scale datasets such as crowdsourcing and web crawling (Fergus et al., 2005; Jiang et al., 2020). Unfortunately, without proper regularization, deep models could be more vulnerable to overfitting the label noise in the training data (Arpit et al., 2017; Zhang et al., 2017), which affects the classification performance adversely.

[1]The University of Tokyo, Japan [2]RIKEN AIP, Japan. Correspondence to: Yivan Zhang <yivanzhang@ms.k.u-tokyo.ac.jp>.

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

Background. Early studies on learning from noisy labels can be traced back to the *random classification noise* (RCN) model for binary classification (Angluin & Laird, 1988; Long & Servedio, 2010; Van Rooyen et al., 2015). Then, RCN has been extended to the case where the noise rate depends on the classes, called the *class-conditional noise* (CCN) model (Natarajan et al., 2013). The multiclass case is of central interest in recent years (Patrini et al., 2017; Goldberger & Ben-Reuven, 2017; Han et al., 2018a; Xia et al., 2019; Yao et al., 2020), where multiclass labels Y flip into other categories $\widetilde{Y}$ according to probabilities described by a fixed but unknown *noise transition matrix* $\boldsymbol{T}$, where $\boldsymbol{T}_{ij} = p(\widetilde{Y} = j|Y = i)$. In this work, we focus on the multiclass CCN model. Other noise models are discussed in Appendix A.

Methodology. The unknown noise transition matrix in CCN has become a hurdle. In this work, we focus on a line of research that aims to estimate the transition matrix from noisy data. With a consistently estimated transition matrix, consistent estimation of the clean class-posterior is possible (Patrini et al., 2017). To estimate the transition matrix, earlier work mainly relies on a given set of *anchor points* (Liu & Tao, 2015; Patrini et al., 2017; Yu et al., 2018), i.e., instances belonging to the true class deterministically. With anchor points, the transition matrix becomes identifiable based on the noisy class-posterior. Further, recent work has attempted to detect anchor points in noisy data to mitigate the lack of anchor points in real-world settings (Xia et al., 2019; Yao et al., 2020).

Nevertheless, even with a given anchor point set, these two-step methods of first estimating the transition matrix and then using it in neural network training face an inevitable problem — the estimation of the noisy class-posterior. The estimation error could be high due to the overconfidence of neural networks (Guo et al., 2017; Hein et al., 2019; Rahimi et al., 2020) (see also Appendix B).

In this work, we present an alternative methodology that does not rely on the error-prone estimation of the noisy class-posterior. The key idea is as follows: Note the fact that the noisy class-posterior vector $\widetilde{\boldsymbol{p}}$ is given by the product of the noise transition matrix $\boldsymbol{T}$ and the clean class-posterior vector $\boldsymbol{p}$: $\widetilde{\boldsymbol{p}} = \boldsymbol{T}^{\top}\boldsymbol{p}$. However, in the reverse direction, the

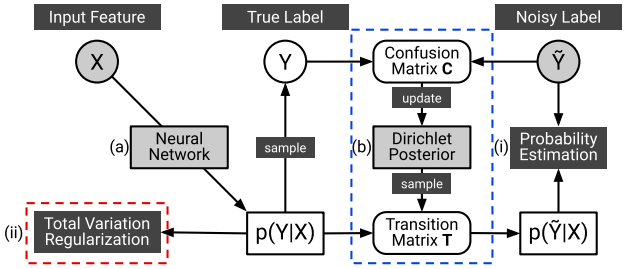

Figure 1. An illustration of the **proposed method**. Our model has two modules: (a) a neural network for predicting $\boldsymbol{p}(Y|X)$; and (b) a *Dirichlet posterior* for the noise transition matrix $\boldsymbol{T}$, whose concentration parameters are updated using the confusion matrix obtained during training. The learning objective in Eq. (12) also contains two parts: (i) the usual cross entropy loss for classification from noisy labels in Eq. (5); and (ii) a *total variation regularization* term for the predicted probability in Eq. (10).

decomposition of the product is not always unique, so $\boldsymbol{T}$ and $\boldsymbol{p}$ are not *identifiable* from $\widetilde{\boldsymbol{p}}$. Thus, without additional assumption, existing methods for estimating $\boldsymbol{T}$ from $\widetilde{\boldsymbol{p}}$ could be unreliable. However, if $\boldsymbol{p}$ has some characteristics, e.g., $\boldsymbol{p}$ is the "cleanest" among all the possibilities, then $\boldsymbol{T}$ and $\boldsymbol{p}$ become identifiable and it is possible to construct consistent estimators. Concretely, we assume that *anchor points exist in the dataset* to guarantee the identifiability, but we do not need to explicitly model or detect them (Theorem 1).

Further, note that the mapping $\boldsymbol{p} \mapsto \boldsymbol{T}^\mathsf{T}\boldsymbol{p}$ is a *contraction* over the probability simplex relative to the *total variation distance* (Del Moral et al., 2003). That is, the "cleanest" $\boldsymbol{p}$ has the property that pairs of $\boldsymbol{p}$ are more distinguishable from each other. Based on this motivation, we propose *total variation regularization* to find the "cleanest" $\boldsymbol{p}$ and consistently estimate $\boldsymbol{T}$ simultaneously (Theorem 2).

Our contribution. In this paper, we study the class-conditional noise (CCN) problem and propose a method that can estimate the noise transition matrix and learn a classifier simultaneously, given only noisy data. The key idea is to regularize the predicted probabilities to be more distinguishable from each other using the pairwise total variation distance. Under mild conditions, the transition matrix becomes identifiable and a consistent estimator can be constructed.

Specifically, we study the characteristics of the class-conditional label corruption process and construct a partial order within the equivalence class of transition matrices in terms of the total variation distance in Section 3, which motivates our proposed method. In Section 4.1, we present the proposed *total variation regularization* and the theorem of consistency (Theorem 2). In Section 4.2, we propose a conceptually novel method based on *Dirichlet distributions* for estimating the transition matrix. Overall, the proposed method is illustrated in Fig. 1.

2. Problem: Class-Conditional Noise

In this section, we review the notation, assumption, and related work in learning with *class-conditional noise* (CCN).

2.1. Noisy Labels

Let $X \in \mathcal{X}$ be the *input feature*, and $Y \in \{1, \dots, K\}$ the *true label*, where K is the number of classes. In fully supervised learning, we can fit a discriminative model for the conditional probability $p(Y|X)$ using an i.i.d. sample of (X, Y)-pairs. However, observed labels may be corrupted in many real-world applications. Treating the corrupted label as correct usually leads to poor performance (Arpit et al., 2017; Zhang et al., 2017). In this work, we explicitly model this label corruption process. We denote the *noisy label* by $\widetilde{Y} \in \{1, \dots, K\}$. The goal is to predict Y from X based on an i.i.d. sample of $(X, \widetilde{Y})$-pairs.

2.2. Class-Conditional Noise (CCN)

Next, we formulate the *class-conditional noise* (CCN) model (Natarajan et al., 2013; Patrini et al., 2017).

In CCN, we have the following assumption:

$$p(\widetilde{Y}|Y, X) = p(\widetilde{Y}|Y). \tag{1}$$

That is, the noisy label $\widetilde{Y}$ only depends on the true label Y but not on X. Then, we can relate the *noisy class-posterior* $p(\widetilde{Y}|X)$ and the *clean class-posterior* $p(Y|X)$ by

$$p(\widetilde{Y}|X) = \textstyle\sum_{Y=1}^{K} p(\widetilde{Y}|Y)p(Y|X). \tag{2}$$

Note that the clean class-posterior $p(Y|X)$ can be seen as a vector-valued function $\boldsymbol{p}(Y|X = x) : \mathcal{X} \to \Delta^{K-1} := [p(Y = 1|X = x), \dots, p(Y = K|X = x)]^\mathsf{T}$, and so can the noisy class-posterior $p(\widetilde{Y}|X)$.[1] Also, $p(\widetilde{Y}|Y)$ can be written in matrix form: $\boldsymbol{T} \in \mathcal{T} \subset [0, 1]^{K \times K}$ with elements $\boldsymbol{T}_{ij} = p(\widetilde{Y} = j|Y = i)$ for $i, j \in \{1, \dots, K\}$, where $\mathcal{T}$ is the set of all full-rank row stochastic matrices. Here, $\boldsymbol{T}$ is called a *noise transition matrix*. Then, with the vector and matrix notation, Eq. (2) can be rewritten as

$$\boldsymbol{p}(\widetilde{Y}|X) = \boldsymbol{T}^\mathsf{T}\boldsymbol{p}(Y|X). \tag{3}$$

Note that multiplying $\boldsymbol{T}$ is a linear transformation from the simplex Δ to the convex hull $\mathrm{Conv}(\boldsymbol{T})$ induced by rows of $\boldsymbol{T}$,[2] which is illustrated in Fig. 2.

In the context of learning from noisy labels, we further assume $\boldsymbol{T}$ to be *diagonally dominant* in the sense that $\boldsymbol{T}_{ii} > \boldsymbol{T}_{ij}$ for $i \neq j$. Although this formulation can be also used for learning from *complementary labels*, where $\boldsymbol{T}_{ii} = 0$ or $\boldsymbol{T}_{ii} < \boldsymbol{T}_{ij}$ for $i \neq j$ (Ishida et al., 2017; Yu et al., 2018).

[1] Δ^{K-1} denotes the $(K-1)$-dimensional probability simplex. The superscript in Δ^{K-1} is omitted hereafter.

[2] Here, $\mathrm{Conv}(\boldsymbol{T})$ is a shorthand for the *convex hull* of the set of vectors $\{\boldsymbol{T}_i | i = 1, \dots, K\}$ within the simplex Δ.

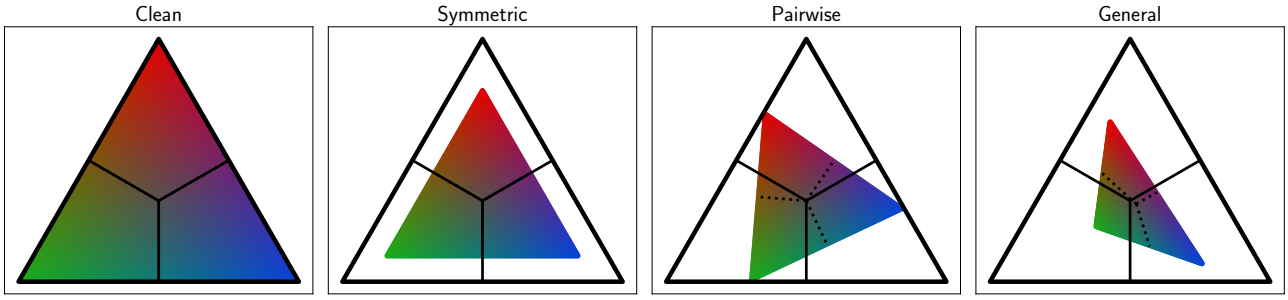

Figure 2. Illustrations of the **noise transition matrix** as a linear transformation $\Delta \to \mathrm{Conv}(\boldsymbol{T})$ when $K = 3$, including symmetric, pair flipping, and general noises. The outer black triangle is the simplex Δ and the inner colored triangle is the convex hull $\mathrm{Conv}(\boldsymbol{T})$. Solid lines are the decision boundaries of argmax and dotted lines are the ones after the transformation. Note that different transition matrices affect the decision boundary differently. Adding symmetric noise does not change the decision boundary.

2.3. Learning with Known Noise Transition Matrix

If the ground-truth noise transition matrix $\boldsymbol{T}$ is known, $\boldsymbol{p}(Y|X)$ is *identifiable* based on observations of $\boldsymbol{p}(\widetilde{Y}|X)$, which means that different $\boldsymbol{p}(Y|X)$ must generate distinct $\boldsymbol{p}(\widetilde{Y}|X)$. Therefore, we can consistently recover $\boldsymbol{p}(Y|X)$ by estimating $\boldsymbol{p}(\widetilde{Y}|X)$ using the relation in Eq. (3) (Patrini et al., 2017; Yu et al., 2018). However, if $\boldsymbol{T}$ and $\boldsymbol{p}(Y|X)$ are both unknown, then they are both *partially identifiable* because there could be multiple observationally equivalent $\boldsymbol{T}$ and $\boldsymbol{p}(Y|X)$ whose product equals $\boldsymbol{p}(\widetilde{Y}|X)$. Thus, it is impossible to estimate both $\boldsymbol{T}$ and $\boldsymbol{p}(Y|X)$ from $\boldsymbol{p}(\widetilde{Y}|X)$ without any additional assumption.

Concretely, let $\widehat{\boldsymbol{p}}(Y|X;W)$ parameterized by $W \in \mathcal{W}$ be a differentiable model for the true label,[3] and $\widehat{\boldsymbol{T}} \in \mathcal{T}$ be an estimator for the noise transition matrix. We consider a sufficiently large function class of $\widehat{\boldsymbol{p}}(Y|X;W)$ that contains the ground-truth $\boldsymbol{p}(Y|X)$, i.e., $\exists W^* \in \mathcal{W}, \widehat{\boldsymbol{p}}(Y|X;W^*) = \boldsymbol{p}(Y|X)$ a.e. In practice, we use an expressive deep neural network (Goodfellow et al., 2016) as $\widehat{\boldsymbol{p}}(Y|X;W)$.

Then, let us consider the *expected Kullback–Leibler (KL) divergence* as the learning objective:

$$L_0(W, \widehat{\boldsymbol{T}}) := \mathop{\mathbb{E}}_{X\sim p(X)} \Big[D_{\mathrm{KL}} \Big(\boldsymbol{p}(\widetilde{Y}|X) \,\Big\|\, \widehat{\boldsymbol{T}}^{\mathsf{T}} \widehat{\boldsymbol{p}}(Y|X;W) \Big) \Big], \tag{4}$$

which is related to the *expected negative log-likelihood* or the *cross-entropy loss* in the following way:

$$L(W, \widehat{\boldsymbol{T}}) := \mathop{\mathbb{E}}_{X,\widetilde{Y}\sim p(X,\widetilde{Y})} \Big[-\log(\widehat{\boldsymbol{T}}_{\widetilde{Y}}^{\mathsf{T}} \widehat{\boldsymbol{p}}(Y|X;W)) \Big] \tag{5}$$

$$= L_0(W, \widehat{\boldsymbol{T}}) + H(\widetilde{Y}|X), \tag{6}$$

where the second term $H(\widetilde{Y}|X)$ is the conditional entropy of $\widetilde{Y}$ given X, which is a constant w.r.t. W and $\widehat{\boldsymbol{T}}$. Note that $L(W, \widehat{\boldsymbol{T}})$ is minimized if and only if $L_0(W, \widehat{\boldsymbol{T}}) = 0$.

When $L(W, \widehat{\boldsymbol{T}})$ is empirically estimated and optimized based on a finite sample of $(X, \widetilde{Y})$-pairs, we can ensure that $\widehat{\boldsymbol{T}}^{\mathsf{T}} \widehat{\boldsymbol{p}}(Y|X) \xrightarrow{\mathrm{d}} \boldsymbol{T}^{\mathsf{T}} \boldsymbol{p}(Y|X)$ as the sample size increases, but we can not guarantee that $\widehat{\boldsymbol{p}}(Y|X) \xrightarrow{\mathrm{d}} \boldsymbol{p}(Y|X)$ due to non-identifiability.[4] The latter holds only when the ground-truth $\boldsymbol{T}$ is used as $\widehat{\boldsymbol{T}}$ (Patrini et al., 2017).

2.4. Learning with Unknown Noise Transition Matrix

In real-world applications, the ground-truth $\boldsymbol{T}$ is usually unknown. Some existing two-step methods attempted to transform this problem to the previously solved one by first estimating the noise transition matrix and then using it in neural network training. Since it is rare to have both clean labels Y and noisy labels $\widetilde{Y}$ for the same instance X, several methods are proposed to estimate $\boldsymbol{T}$ from only noisy data.

Existing methods usually rely on a separate set of *anchor points* (Liu & Tao, 2015; Patrini et al., 2017; Yu et al., 2018; Xia et al., 2019; Yao et al., 2020), which are defined as follows:

Definition 1 (Anchor point)**.** An instance x is called an anchor point for class i if $p(Y = i|X = x) = 1$.

Based on an anchor point x for class i, we have

$$\boldsymbol{p}(\widetilde{Y}|X = x) = \boldsymbol{T}^{\mathsf{T}} \boldsymbol{p}(Y|X = x) = \boldsymbol{T}_i. \tag{7}$$

Thus, we can first estimate $\boldsymbol{p}(\widetilde{Y}|X)$ and then calculate the value on anchor points to obtain an estimate of $\boldsymbol{T}$. However, if we cannot find such anchor points in real-world datasets easily, the aforementioned method can not be applied.

A workaround is to detect anchor points from all noisy data, assuming that they exist in the dataset. Further revision of the transition matrix before (Yao et al., 2020) or during (Xia et al., 2019) the second stage of training can be adopted to improve the performance.

[3] W is sometimes omitted to keep the notation uncluttered.

[4] $\xrightarrow{\mathrm{d}}$ denotes the convergence in distribution.

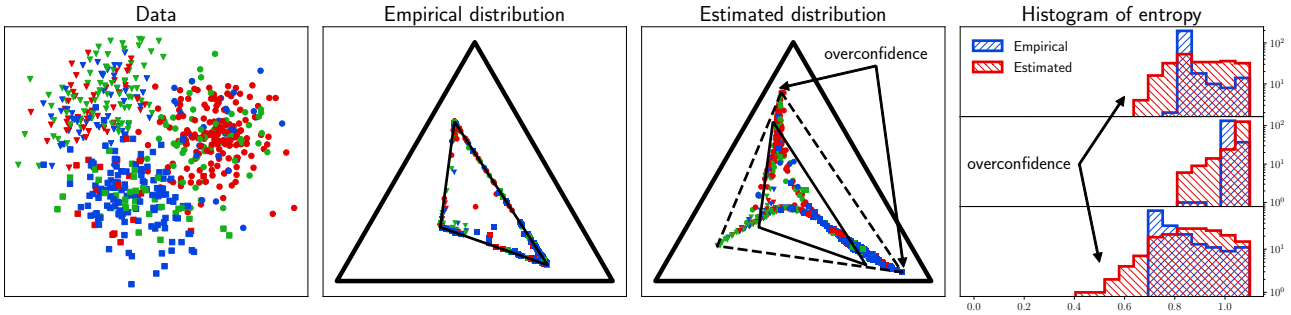

Figure 3. An example of **overconfident predictions** yield from neural networks. Notation: • Shape: true labels; • Color: observed labels; • Location in the simplex: ground-truth/estimated probabilities; • Solid triangle: convex hull $\mathrm{Conv}(\boldsymbol{T})$; • Dashed triangle: convex hull of an estimated transition matrix; • Vertices of the triangle: anchor points. Without knowing $\boldsymbol{T}$ and the constraint that $\boldsymbol{p}(\widetilde{Y}|X)$ should be within $\mathrm{Conv}(\boldsymbol{T})$, a neural network trained with noisy labels tends to output high-confident (low-entropy) predictions outside of $\mathrm{Conv}(\boldsymbol{T})$. Therefore, $\boldsymbol{T}$ may be poorly estimated based on the overconfident noisy class-posterior.

Nevertheless, such two-step methods based on anchor points face an inevitable problem — the estimation of the noisy class-posterior $\boldsymbol{p}(\widetilde{Y}|X)$ using possibly over-parameterized neural networks trained with noisy labels. We point out that the estimation error could be high in this step because of the overconfidence problem of deep neural networks (Guo et al., 2017; Hein et al., 2019). If no revision is made, errors in the first stage may lead to major errors in the second stage.

Figure 3 illustrates an example of the overconfidence. As discussed in Section 2.2, $\boldsymbol{p}(\widetilde{Y}|X)$ should be within the convex hull $\mathrm{Conv}(\boldsymbol{T})$. However, without knowing $\boldsymbol{T}$ and this constraint, a neural network trained with noisy labels tends to output overconfident probabilities that are outside of $\mathrm{Conv}(\boldsymbol{T})$. Note that over-parameterized neural networks trained with clean labels could also be overconfident and several re-calibration methods are developed to alleviate this issue (Guo et al., 2017; Kull et al., 2019; Hein et al., 2019; Rahimi et al., 2020). However, in Appendix B we demonstrate that estimating the noise class-posterior causes a significantly worse overconfidence issue than estimating the clean one. Consequently, transition matrix estimation may suffer from poorly estimated noisy class-posteriors, which leads to performance degradation.

In contrast to existing methods, our proposed method only uses the product $\widehat{\boldsymbol{T}}^{\mathsf{T}}\widehat{\boldsymbol{p}}(Y|X)$ as an estimate of $\boldsymbol{p}(\widetilde{Y}|X)$ and never estimates $\boldsymbol{p}(\widetilde{Y}|X)$ directly using neural networks.

3. Motivation

In this section, we take a closer look at the class-conditional label corruption process and construct an equivalence class and a partial order for the noise transition matrix, which motivates our proposed method. Concretely, we show that the *contraction property* of the stochastic matrices leads to a partial order of the transition matrices, which can be used to find the "cleanest" clean class-posterior.

3.1. Transition Matrix Equivalence

Recall that $\mathcal{T}$ is the set of full-rank row stochastic matrices, which is *closed under multiplication*. Based on this, we first define an *equivalence relation* of an ordered pair of transition matrices induced by the product:

Definition 2 (Transition matrix equivalence)**.**

$$(\boldsymbol{U}, \boldsymbol{V}) \sim (\boldsymbol{U}', \boldsymbol{V}') \Leftrightarrow \boldsymbol{U}\boldsymbol{V} = \boldsymbol{U}'\boldsymbol{V}'.$$

The corresponding equivalence class with a product $\boldsymbol{W}$ is denoted by $[\boldsymbol{W}]$. Specially, for the identity matrix $\boldsymbol{I}$, $[\boldsymbol{I}]$ contains pairs of permutation matrices $(\boldsymbol{P}, \boldsymbol{P}^{-1})$; for a non-identity matrix $\boldsymbol{W}$, $[\boldsymbol{W}]$ contains at least two distinct elements $(\boldsymbol{W}, \boldsymbol{I})$ and $(\boldsymbol{I}, \boldsymbol{W})$ and possibly infinitely many other elements.

Now, consider the equivalence class $[\boldsymbol{T}]$ for the ground-truth noise transition matrix $\boldsymbol{T}$ in our problem. Then, any element $(\boldsymbol{U}, \boldsymbol{V}) \in [\boldsymbol{T}]$ corresponds to a possible optimal solution of Eq. (5): $\widehat{\boldsymbol{T}} = \boldsymbol{V}$ and $\widehat{\boldsymbol{p}}(Y|X; W) = \boldsymbol{U}^{\mathsf{T}}\boldsymbol{p}(Y|X)$, given that such a parameter W exists. Among possibly infinitely many possibilities, only $(\boldsymbol{I}, \boldsymbol{T})$ is of our central interest. However, it is possible to get infinitely many other wrong ones, such as $(\boldsymbol{T}, \boldsymbol{I})$, which corresponds to a model that predicts the noisy class-posterior and chooses the transition matrix to be the identity matrix $\boldsymbol{I}$.

3.2. Transition Matrix Decomposition

Next, consider the reverse direction: if we obtain an optimal solution of Eq. (5), $\widehat{\boldsymbol{T}}$ and $\widehat{\boldsymbol{p}}(Y|X)$, is there a $\boldsymbol{U}$ such that $\widehat{\boldsymbol{p}}(Y|X) = \boldsymbol{U}^{\mathsf{T}}\boldsymbol{p}(Y|X)$? The answer is yes if there are anchor points for each class in the dataset, which can be proved using the following theorem:

Theorem 1 (Transition matrix decomposition)**.** For two row stochastic matrices $\boldsymbol{W}, \boldsymbol{V} \in \mathcal{T}$, if $\forall \boldsymbol{p} \in \Delta$, $\exists \boldsymbol{q} \in \Delta$, s.t. $\boldsymbol{W}^{\mathsf{T}}\boldsymbol{p} = \boldsymbol{V}^{\mathsf{T}}\boldsymbol{q}$, then $\exists$ a row stochastic matrix $\boldsymbol{U} \in \mathcal{T}$, s.t. $\boldsymbol{W} = \boldsymbol{U}\boldsymbol{V}$ and $\forall \boldsymbol{p} \in \Delta$, $\boldsymbol{q} = \boldsymbol{U}^{\mathsf{T}}\boldsymbol{p}$.

Proof. Let $\boldsymbol{p}$ be $\boldsymbol{e}_i$ and denote the corresponding $\boldsymbol{q}$ by $\boldsymbol{q}_i$ for $i = 1, \ldots, K$. Then $\boldsymbol{U} = [\boldsymbol{q}_1, \ldots, \boldsymbol{q}_K]^\top$. □

Here, $\boldsymbol{e}_i$ is the i-th standard basis and $\boldsymbol{p}(Y|X = x) = \boldsymbol{e}_i$ means that x is an anchor point for the class i. Consequently, we can derive that if there are anchor points for each class in the dataset, given an estimated transition matrix $\widehat{\boldsymbol{T}}$ and an estimated clean class-posterior $\widehat{\boldsymbol{p}}(Y|X)$ from an optimal solution of Eq. (5), we know that there is an *implicit* row stochastic matrix $\boldsymbol{U}$ such that $\widehat{\boldsymbol{p}}(Y|X) = \boldsymbol{U}^\top \boldsymbol{p}(Y|X)$. In other words, the estimate $\widehat{\boldsymbol{p}}(Y|X)$ may still contain class-conditional label noise, which is described by $\boldsymbol{U}$.

We point out that the existence of anchor points is a *sufficient but not necessary* condition for the existence of the $\boldsymbol{U}$ above. If anchor points do not exist, we may or may not find such a $\boldsymbol{U}$. Also note that we will not try to detect anchor points from noisy data.

More importantly, we have no intention of estimating $\boldsymbol{U}$ explicitly. In this work, we only use the fact that there is a one-to-one correspondence between optimal solutions of Eq. (5) and elements in the equivalence class $[\boldsymbol{T}]$ under the above assumption. Based on this fact, we can study the equivalence class $[\boldsymbol{T}]$ or the properties of the implicit $\boldsymbol{U}$ instead, which is easier to deal with.

3.3. Transition Matrix as a Contraction Mapping

Next, we attempt to *break the equivalence* introduced above by examining the characteristics of this consecutive class-conditional label corruption process.

We start with the definition of the *total variation distance* $d_{\mathrm{TV}}(\cdot, \cdot)$ between pairs of categorical probabilities:

$$d_{\mathrm{TV}}(\boldsymbol{p}, \boldsymbol{q}) := \frac{1}{2}\|\boldsymbol{p} - \boldsymbol{q}\|_1, \tag{8}$$

where $\|\cdot\|_1$ denotes the ℓ_1 norm. Then, from the theory of Markov chains, we know that the mapping $\Delta \to \mathrm{Conv}(\boldsymbol{U})$ defined by $\boldsymbol{p} \mapsto \boldsymbol{U}^\top \boldsymbol{p}$ is a *contraction mapping* over the simplex Δ relative to the total variation distance (Del Moral et al., 2003), which means that $\forall \boldsymbol{U} \in \mathcal{T}, \forall \boldsymbol{p}, \boldsymbol{q} \in \Delta$,

$$d_{\mathrm{TV}}(\boldsymbol{U}^\top \boldsymbol{p}, \boldsymbol{U}^\top \boldsymbol{q}) \leq d_{\mathrm{TV}}(\boldsymbol{p}, \boldsymbol{q}). \tag{9}$$

3.4. Transition Matrix Partial Order

Finally, based on this contraction property of the stochastic matrices, we can introduce a *partial order* induced by the total variation distance within the equivalence class $[\boldsymbol{T}]$:

Definition 3 (Transition matrix partial order)**.**

$$(\boldsymbol{U}, \boldsymbol{V}) \preceq (\boldsymbol{U}', \boldsymbol{V}') \Leftrightarrow$$
$$\forall \boldsymbol{p}, \boldsymbol{q} \in \Delta, d_{\mathrm{TV}}(\boldsymbol{U}^\top \boldsymbol{p}, \boldsymbol{U}^\top \boldsymbol{q}) \leq d_{\mathrm{TV}}(\boldsymbol{U}'^\top \boldsymbol{p}, \boldsymbol{U}'^\top \boldsymbol{q}).$$

Note that $(\boldsymbol{I}, \boldsymbol{T})$ is the *unique greatest element* because of Eq. (9). Despite the fact that there could be incomparable elements, we may gradually increase the total variation to find $(\boldsymbol{I}, \boldsymbol{T})$. Then, with the help of this partial order, it is possible to estimate both $\boldsymbol{p}(Y|X)$ and $\boldsymbol{T}$ simultaneously, which is discussed in the following section.

4. Proposed Method

In this section, we present our proposed method. Overall, the proposed method is illustrated in Fig. 1.

Summarizing our motivation discussed in Section 3, we found that if anchor points exist in the dataset, estimating both the transition matrix $\boldsymbol{T}$ and the clean class-posterior $\boldsymbol{p}(Y|X)$ by training with the cross-entropy loss in Eq. (5) results in a solution in the form $\widehat{\boldsymbol{p}}(Y|X) = \boldsymbol{U}^\top \boldsymbol{p}(Y|X)$, where $\boldsymbol{U}$ is an unknown transition matrix (Theorem 1). Then, we pointed out that the stochastic matrices have the contraction property shown in Eq. (9) so that the "cleanest" clean class-posterior has the highest pairwise total variation defined in Eq. (8). Based on this fact, we can regularize the predicted probabilities to be more distinguishable from each other to find the optimal solution, as discussed below.

4.1. Total Variation Regularization

First, we discuss how to enforce our preference of more distinguishable predictions in terms of the total variation distance. We start with defining the *expected pairwise total variation distance*:

$$\begin{aligned} R(W) &:= \mathop{\mathbb{E}}_{x_1 \sim p(X)} \mathop{\mathbb{E}}_{x_2 \sim p(X)} [d_{\mathrm{TV}}(\widehat{\boldsymbol{p}}_1, \widehat{\boldsymbol{p}}_2)], \\ \text{where } \widehat{\boldsymbol{p}}_i &:= \widehat{\boldsymbol{p}}(Y|X = x_i; W), \quad i = 1, 2. \end{aligned} \tag{10}$$

Note that this *data-dependent* term depends on X but not on Y nor on $\widetilde{Y}$.

Then, we adopt the learning objective in the KL-divergence form in Eq. (4), combine it with the expected pairwise total variation distance in Eq. (10), and formulate our approach in the form of *constrained optimization*, as stated in the following theorem:

Theorem 2 (Consistency)**.** Given a finite i.i.d. sample of $(X, \widetilde{Y})$-pairs of size N, where anchor points (Definition 1) for each class exist in the sample, let $\widetilde{L}_0(W, \widehat{\boldsymbol{T}})$ and $\widetilde{R}(W)$ be the empirical estimates of $L_0(W, \widehat{\boldsymbol{T}})$ in Eq. (4) and $R(W)$ in Eq. (10), respectively. Assume that the parameter space $\mathcal{W}$ is compact. Let $(W^\circ, \widehat{\boldsymbol{T}}^\circ)$ be an optimal solution of the following constrained optimization problem:

$$\max_W \widetilde{R}(W) \text{ s.t. } \widetilde{L}_0(W, \widehat{\boldsymbol{T}}) = 0. \tag{11}$$

Then, $\widehat{\boldsymbol{T}}^\circ$ is a consistent estimator of the transition matrix $\boldsymbol{T}$; and $\widehat{\boldsymbol{p}}(Y|X; W^\circ) \xrightarrow{\mathrm{d}} \boldsymbol{p}(Y|X)$ a.e. as $N \to \infty$.

The proof is given in Appendix C. Informally, we make use of Theorem 1, the property of the KL-divergence, and the contraction property of the transition matrix.

In practice, the constrained optimization in Eq. (11) can be solved via the following Lagrangian (Kuhn et al., 1951):

$$\mathcal{L}(W, \widehat{\boldsymbol{T}}) := \widetilde{L}_0(W, \widehat{\boldsymbol{T}}) - \gamma \widetilde{R}(W), \tag{12}$$

where $\gamma \in \mathbb{R}_{>0}$ is a parameter controlling the importance of the regularization term. We call such a regularization term a *total variation regularization*. This Lagrangian technique has been widely used in the literature (Cortes & Vapnik, 1995; Kloft et al., 2009; Higgins et al., 2017; Li et al., 2021). When the total variation regularization term is empirically estimated and optimized, we can sample a fixed number of pairs to reduce the additional computational cost.

4.2. Transition Matrix Estimation

Next, we discuss the estimation of the transition matrix $\boldsymbol{T}$. In contrast to existing methods (Patrini et al., 2017; Xia et al., 2019; Yao et al., 2020), we adopt a one-step training procedure to obtain both $\widehat{\boldsymbol{p}}(Y|X)$ and $\widehat{\boldsymbol{T}}$ simultaneously.

Gradient-based estimation. First, note that the learning objective Eq. (12) is differentiable w.r.t. $\widehat{\boldsymbol{T}}$. As a baseline, it is sufficient to use gradient-based optimization for $\widehat{\boldsymbol{T}}$. In practice, we apply softmax to an unconstrained matrix in $\mathbb{R}^{K\times K}$ to ensure that $\widehat{\boldsymbol{T}} \in \mathcal{T}$. Then, $\widehat{\boldsymbol{T}}$ is estimated by optimizing $\mathcal{L}(W, \widehat{\boldsymbol{T}})$ using stochastic gradient descent (SGD) or its variants (e.g., Kingma & Ba (2015)).

Dirichlet posterior update. The additional total variation regularization term Eq. (10) is irrelevant to $\widehat{\boldsymbol{T}}$ so we are free to use other optimization methods besides gradient-based methods. To capture the uncertainty of the estimation of $\boldsymbol{T}$ during different stages of training, we propose an alternative *derivative-free* approach that uses Dirichlet distributions to model $\boldsymbol{T}$. Concretely, let the posterior of $\boldsymbol{T}$ be

$$\widehat{\boldsymbol{T}}_i \sim \text{Dirichlet}(\boldsymbol{A}_i) \quad (i = 1, \dots, K), \tag{13}$$

where $\boldsymbol{A}_i \in \mathbb{R}^K_{>0}$ is the concentration parameter. Denote the *confusion matrix* by $\boldsymbol{C} \in \mathbb{N}^{K\times K}_{\geq 0}$, where its element $\boldsymbol{C}_{ij}$ is the number of instances that are predicted to be $\widehat{Y} = i$ via sampling $\widehat{Y} \sim \widehat{\boldsymbol{p}}(Y|X; W)$ but are labeled as $\widetilde{Y} = j$ in the noisy dataset. In other words, we use a posterior of $p(\widetilde{Y}|\widehat{Y})$ to approximate $p(\widetilde{Y}|Y)$ during training.

Then, inspired by the closed-form posterior update rule for the Dirichlet-multinomial conjugate (Diaconis & Ylvisaker, 1979):

$$\boldsymbol{A}^{(\text{posterior})} = \boldsymbol{A}^{(\text{prior})} + \boldsymbol{C}^{(\text{observation})}, \tag{14}$$

we update the concentration parameters $\boldsymbol{A}$ during training using the confusion matrix $\boldsymbol{C}$ via the following update rule:

$$\boldsymbol{A} \leftarrow \beta_1 \boldsymbol{A} + \beta_2 \boldsymbol{C}, \tag{15}$$

where $\boldsymbol{\beta} = (\beta_1, \beta_2)$ are fixed hyperparameters that control the convergence of $\boldsymbol{A}$. We initialize $\boldsymbol{A}$ with an appropriate diagonally dominant matrix to reflect our prior knowledge of noisy labels. For each batch of data, we sample a noise transition matrix $\widehat{\boldsymbol{T}}$ from the Dirichlet posterior and use it in our learning objective in Eq. (12).

The idea is that the confusion matrix $\boldsymbol{C}$ at any stage during training is a crude estimator of the true noise transition matrix $\boldsymbol{T}$, then we can improve this estimator based on information obtained during training. Because at earlier stage of training, this estimator is very crude and may be deviated from the true one significantly, we use a decaying factor β_1 close to 1 (e.g., 0.999) to let the model gradually "forget" earlier information. Meanwhile, β_2 controls the variance of the Dirichlet posterior during training, which is related to the learning rate and batch size. At early stages, the variance is high so the model is free to explore various transition matrices; as the model converges, the estimation of the transition matrix also becomes more precise so the posterior would concentrate around the true one.

5. Related Work

In addition to methods using the noise transition matrix explicitly and two-step methods detecting anchor points from noisy data (Patrini et al., 2017; Yu et al., 2018; Xia et al., 2019; Yao et al., 2020) introduced in Sections 1 and 2, in this section we review related work in learning from noisy labels in a broader sense.

First, in CCN, is it possible to learn a correct classifier *without* the noise transition matrix? Existing studies in *robust loss functions* (Ghosh et al., 2017; Zhang & Sabuncu, 2018; Wang et al., 2019; Charoenphakdee et al., 2019; Ma et al., 2020; Feng et al., 2020; Lyu & Tsang, 2020; Liu & Guo, 2020) showed that it is possible to alleviate the label noise issue even without estimating the noise rate/transition matrix, under various conditions such as the noise being symmetric (the RCN model in binary classification (Angluin & Laird, 1988)). Further, it is proven that the accuracy metric itself can be robust (Chen et al., 2021). However, if the noise is heavy and complex, robust losses may perform poorly. This motivates us to evaluate our method under various types of label noises beyond the symmetric noise.

Another direction is to learn a classifier that is robust against label noise, including *training sample selection* (Malach & Shalev-Shwartz, 2017; Jiang et al., 2018; Han et al., 2018b; Wang et al., 2018; Yu et al., 2019; Wei et al., 2020; Mirzasoleiman et al., 2020; Wu et al., 2020) that selects training

Table 1. **Accuracy** (%) on the MNIST, CIFAR-10, and CIFAR-100 datasets. We reported "mean (standard deviation)" of 10 trials.

		(a) Clean	(b) Symm.	(c) Pair	(d) Pair^2	(e) Trid.	(f) Rand.
MNIST	MAE	98.72(0.09)	98.00(0.14)	91.46(7.40)	89.79(6.11)	96.22(3.87)	34.07(31.98)
	CCE	**99.21(0.04)**	98.13(0.16)	94.70(0.64)	94.86(0.67)	96.78(0.22)	95.68(1.31)
	GCE	99.12(0.06)	98.41(0.12)	93.79(1.04)	94.06(0.63)	96.60(0.14)	96.28(0.93)
	Forward	**99.18(0.05)**	98.00(0.24)	94.37(1.00)	94.84(0.53)	96.54(0.29)	95.95(1.49)
	T-Revision	**99.20(0.06)**	98.01(0.14)	94.19(0.78)	95.24(0.74)	96.76(0.15)	96.62(0.70)
	Dual-T	99.16(0.05)	**98.58(0.12)**	**99.06(0.07)**	**99.03(0.06)**	**99.04(0.05)**	**98.79(0.17)**
	TVG	99.16(0.06)	**98.55(0.09)**	94.26(0.59)	95.42(0.44)	97.78(0.56)	97.67(0.84)
	TVD	**99.18(0.07)**	**98.56(0.08)**	**99.09(0.08)**	**99.00(0.07)**	**99.03(0.08)**	**98.82(0.11)**
CIFAR10	MAE	66.47(4.76)	57.23(4.15)	44.29(2.23)	42.43(1.66)	43.43(2.69)	26.95(5.45)
	CCE	**91.87(0.19)**	75.71(0.57)	65.54(0.66)	65.23(0.85)	76.07(0.61)	70.44(1.98)
	GCE	89.25(0.17)	**83.68(0.29)**	71.49(1.18)	69.66(0.57)	82.14(0.41)	78.07(2.16)
	Forward	**91.87(0.15)**	76.18(0.63)	65.42(0.92)	65.65(1.11)	76.41(0.50)	70.86(2.19)
	T-Revision	91.72(0.18)	75.51(0.59)	65.49(0.97)	65.70(0.66)	76.18(0.80)	71.22(1.62)
	Dual-T	**91.75(0.18)**	82.85(0.42)	80.86(1.03)	79.61(1.20)	**88.11(0.28)**	84.33(2.11)
	TVG	91.61(0.14)	82.60(0.38)	**89.78(0.16)**	**88.36(0.24)**	**88.07(0.25)**	**86.19(0.52)**
	TVD	91.00(0.13)	83.03(0.24)	88.47(0.29)	86.96(0.35)	87.44(0.16)	**85.86(0.46)**
CIFAR100	MAE	11.23(1.02)	7.89(0.67)	6.94(1.11)	6.60(0.74)	7.45(0.55)	7.15(0.98)
	CCE	**70.58(0.29)**	42.94(0.47)	44.00(0.71)	41.37(0.27)	46.55(0.54)	42.41(0.48)
	GCE	57.10(0.85)	48.66(0.58)	45.27(0.85)	43.67(0.94)	50.98(0.33)	48.66(0.63)
	Forward	**70.58(0.28)**	44.32(0.64)	44.17(0.57)	42.07(0.55)	47.48(0.40)	43.15(0.53)
	T-Revision	**70.47(0.26)**	46.52(0.57)	44.08(0.42)	42.01(0.52)	47.59(0.60)	45.33(0.40)
	Dual-T	**70.56(0.28)**	55.92(0.60)	46.22(0.72)	44.74(0.65)	61.68(0.51)	57.92(0.50)
	TVG	70.02(0.30)	**57.33(0.42)**	45.68(0.85)	44.38(0.72)	54.23(0.53)	**59.85(0.61)**
	TVD	69.93(0.21)	52.54(0.45)	**56.02(0.82)**	**49.18(0.53)**	**62.45(0.44)**	53.95(0.47)

examples during training, *learning with rejection* (El-Yaniv & Wiener, 2010; Thulasidasan et al., 2019; Mozannar & Sontag, 2020; Charoenphakdee et al., 2021) that abstains from using confusing instances, *meta-learning* (Shu et al., 2019; Li et al., 2019), and *semi-supervised learning* (Nguyen et al., 2020; Li et al., 2020). These methods exploit the training dynamics, characteristics of loss distribution, or information of data itself instead of the class-posteriors. Then, the CCN assumption in Eq. (1) might not be needed but accordingly these methods usually have limited consistency guarantees. Moreover, the computational cost and model complexity of these methods could be higher.

For the CCN model and noise transition matrix estimation, recently, the idea of solving the class-conditional label noise problem using a one-step method was concurrently used by Li et al. (2021), aiming to relax the anchor point assumption. They adopt a different approach based on the characteristics of the noise transition matrix, instead of the properties of the clean class-posterior used in our work. Li et al. (2021) has the advantage that their assumption is weaker than ours. However, the additional term on the transition matrix might be incompatible with derivative-free optimization, such as the Dirichlet posterior update method proposed in our work.

6. Experiments

In this section, we present experimental results to show that the proposed method achieves lower estimation error of the transition matrix and consequently better classification accuracy for the true labels, confirming Theorem 2.

6.1. Benchmark Datasets

We evaluated our method on three image classification datasets, namely **MNIST** (LeCun et al., 1998), **CIFAR-10**, and **CIFAR-100** (Krizhevsky, 2009). We used various noise types besides the common symmetric noise and pair flipping noise.

Concretely, noise types include: (a) (**Clean**) no additional synthetic noise, which serves as a baseline for the dataset and model; (b) (**Symm.**) symmetric noise 50% (Patrini et al., 2017); (c) (**Pair**) pair flipping noise 40% (Han et al., 2018b); (d) (**Pair**^{2}) a product of two pair flipping noise matrices with noise rates 30% and 20%. Because the multiplication of pair flipping noise matrices is commutative, it is guaranteed to have multiple ways of decomposition of the transition matrix; (e) (**Trid.**) tridiagonal noise (see also Han et al., 2018a), which corresponds to a spectral of classes where

Table 2. **Average total variation** ($\times 100$) on the MNIST, CIFAR-10, and CIFAR-100 datasets. We reported "mean (standard deviation)" of 10 trials.

		(a) Clean	(b) Symm.	(c) Pair	(d) Pair^2	(e) Trid.	(f) Rand.
MNIST	Forward	**0.00(0.00)**	34.14(3.03)	39.71(0.15)	41.98(0.82)	38.33(0.93)	30.45(2.16)
	T-Revision	0.03(0.02)	32.94(3.22)	39.87(0.08)	41.50(0.50)	38.39(1.34)	29.35(1.85)
	Dual-T	0.12(0.02)	7.12(0.99)	3.90(0.66)	3.59(0.58)	3.11(0.88)	10.63(0.90)
	TVG	2.36(0.01)	**1.47(0.13)**	39.29(0.03)	32.17(0.93)	14.11(5.21)	7.33(4.25)
	TVD	2.06(0.12)	1.96(0.17)	**2.12(0.21)**	**2.12(0.10)**	**1.92(0.11)**	**2.13(0.22)**
CIFAR-10	Forward	**0.00(0.00)**	47.63(0.35)	39.09(0.28)	41.70(0.32)	35.63(0.81)	45.52(0.65)
	T-Revision	0.03(0.03)	43.05(0.36)	39.13(0.22)	40.80(0.30)	34.82(0.67)	43.05(0.52)
	Dual-T	0.81(0.04)	**2.99(0.23)**	19.37(0.45)	16.84(0.61)	4.60(0.31)	8.80(1.57)
	TVG	0.64(0.01)	3.17(0.19)	**1.56(0.13)**	**2.16(0.22)**	**1.94(0.18)**	**2.24(0.26)**
	TVD	7.87(0.10)	6.90(0.18)	8.46(0.17)	8.70(0.24)	7.06(0.14)	7.98(0.38)
CIFAR-100	Forward	**0.00(0.00)**	48.62(0.11)	39.81(0.03)	43.57(0.04)	40.92(0.07)	49.06(0.10)
	T-Revision	0.46(0.05)	31.58(0.46)	39.45(0.03)	42.77(0.06)	40.01(0.09)	39.49(0.26)
	Dual-T	3.10(0.08)	17.10(0.18)	33.26(0.20)	33.79(0.26)	**23.56(0.43)**	22.59(0.23)
	TVG	1.59(0.02)	**13.11(0.10)**	37.79(0.30)	38.83(0.34)	30.80(0.51)	**16.47(0.18)**
	TVD	21.98(0.11)	26.46(0.15)	**29.47(0.26)**	**31.34(0.30)**	23.86(0.22)	35.37(0.30)

adjacent classes are easier to be *mutually* mislabeled, unlike the *unidirectional* pair flipping; and (f) (**Rand.**) random noise constructed by sampling a Dirichlet distribution and mixing with the identity matrix to a specified noise rate. See Appendix E for details.

Methods. We compared the following methods: (1) (**MAE**) mean absolute error (Ghosh et al., 2017) as a robust loss; (2) (**CCE**) categorical cross-entropy loss; (3) (**GCE**) generalized cross-entropy loss (Zhang & Sabuncu, 2018); (4) (**Forward**) forward correction (Patrini et al., 2017) based on anchor points detection; (5) (**T-Revision**) transition-revision (Xia et al., 2019) where the transition matrix is further revised during the second stage of training; (6) (**Dual-T**) dual-T estimator (Yao et al., 2020) that uses the normalized confusion matrix to correct the transition matrix; (7) (**TVG**) total variation regularization with the gradient-based estimation of $\boldsymbol{T}$; and (8) (**TVD**) the one with the Dirichlet posterior update.

Models. For MNIST, we used a sequential convolutional neural network (CNN) and an Adam optimizer (Kingma & Ba, 2015). For both CIFAR-10 and CIFAR-100, we used a residual network model ResNet-18 (He et al., 2016) and a stochastic gradient descent (SGD) optimizer with momentum (Sutskever et al., 2013).

Hyperparameters. For the gradient-based estimation, we initialized the unconstrained matrix with diagonal elements of $\log(0.5)$ and off-diagonal elements of $\log(0.5/(K-1))$, so after applying softmax the diagonal elements are 0.5. For the Dirichlet posterior update method, we initialized the concentration matrix with diagonal elements of 10 for MNIST and 100 otherwise and off-diagonal elements of 0. We set $\boldsymbol{\beta} = (0.999, 0.01)$ and $\gamma = 0.1$. We sampled 512 (the same as the batch size) pairs in each batch to calculate the pairwise total variation distance. Other hyperparameters are provided in Appendix E.

Evaluation metrics. In addition to the test *accuracy*, we reported the *average total variation* to evaluate the transition matrix estimation, which is defined as follows:

$$\frac{1}{K}\sum_{i=1}^{K} d_{\mathrm{TV}}(\boldsymbol{T}_i, \widehat{\boldsymbol{T}}_i) = \frac{1}{K}\sum_{i=1}^{K}\frac{1}{2}\sum_{j=1}^{K}\left|\boldsymbol{T}_{ij} - \widehat{\boldsymbol{T}}_{ij}\right| \in [0,1].$$

Results. We ran 10 trials for each experimental setting and reported "mean (standard deviation)" of the accuracy and average total variation in Tables 1 and 2, respectively. Outperforming methods are highlighted in boldface using one-tailed t-tests with a significance level of 0.05.

In Table 1, we observed that the proposed methods performs well in terms of accuracy. Note that a baseline method Dual-T also showed superiority in some settings, which sheds light on the benefits of using the confusion matrix. However, as a two-step method, their computational cost is at least twice ours. In Table 2, we can confirm that in most settings, our methods have lower estimation error of the transition matrix than baselines, sometimes by a large margin. For better reusability, we fixed the initial transition matrix/concentration parameters across all different noise types. If we have more prior knowledge about the noise, a better initialization may further improve the performance.

Table 3. **Accuracy** (%) on the Clothing1M dataset.

CCE	Forward	T-Revision	Dual-T	TVD
69.91	69.96	69.97	70.67	71.65

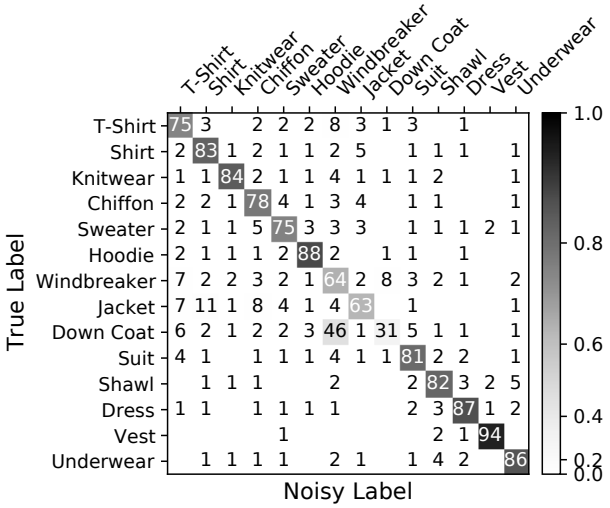

Figure 4. **Estimated transition matrix** ($\times 100$) on Clothing1M.

6.2. Real-World Dataset

We also evaluated our method on a real-world noisy label dataset, Clothing1M (Xiao et al., 2015). Unlike some previous work that also used a small set of clean training data (Patrini et al., 2017; Xia et al., 2019), we only used the $1M$ noisy training data. We followed previous work for other settings such as the model and optimization. We implemented data-parallel distributed training on 64 NVIDIA Tesla P100 GPUs by PyTorch (Paszke et al., 2019). See Appendix E for details.

Results. In Table 3, we reported the test accuracy. The transition matrix estimated by our proposed method was plotted in Fig. 4.

We can see that our method outperformed the baselines in terms of accuracy, which demonstrated the effectiveness of our method in real-world settings. Although there is no ground-truth transition matrix for evaluation, we can observe the *similarity relationship* between categories from the estimated transition matrix, which itself could be of great interest. For example, if two categories are relatively easy to be mutually mislabeled, they may be visually similar; if one category can be mislabeled as another, but not vice versa, we may get a semantically meaningful hierarchy of categories. Further investigation is left for future work.

7. Conclusion

We have introduced a novel method for estimating the noise transition matrix and learning a classifier simultaneously, given only noisy data. In this problem, the supervision is insufficient to identify the true model, i.e., we have a class of observationally equivalent models. We address this issue by finding characteristics of the true model under realistic assumptions and introducing a partial order as a regularization. As a result, the proposed *total variation regularization* is theoretically guaranteed to find the optimal transition matrix under mild conditions, which is reflected in experimental results on benchmark datasets.

Acknowledgements

We thank Xuefeng Li, Tongliang Liu, Nontawat Charoenphakdee, Zhenghang Cui, and Nan Lu for insightful discussion. We also would like to thank the Supercomputing Division, Information Technology Center, the University of Tokyo, for providing the Reedbush supercomputer system. YZ was supported by Microsoft Research Asia D-CORE program and RIKEN's Junior Research Associate (JRA) program. GN and MS were supported by JST AIP Acceleration Research Grant Number JPMJCR20U3, Japan. MS was also supported by the Institute for AI and Beyond, UTokyo.

References

Angluin, D. and Laird, P. Learning from noisy examples. *Machine Learning*, 2(4):343–370, 1988. 1, 5, A

Arpit, D., Jastrzębski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., et al. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, 2017. 1, 2.1

Bartlett, P. L., Jordan, M. I., and McAuliffe, J. D. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006. B

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

Blanchard, G. and Scott, C. Decontamination of mutually contaminated models. In *Artificial Intelligence and Statistics*, pp. 1–9, 2014. A

Charoenphakdee, N., Lee, J., and Sugiyama, M. On symmetric losses for learning from corrupted labels. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 961–970, 2019. 5

Charoenphakdee, N., Cui, Z., Zhang, Y., and Sugiyama, M. Classification with rejection based on cost-sensitive classification. In *Proceedings of the 38th International Conference on Machine Learning*, 2021. 5

Chen, P., Ye, J., Chen, G., Zhao, J., and Heng, P.-A. Robustness of accuracy metric and its inspirations in learning with noisy labels. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence*, 2021. 5

Cheng, J., Liu, T., Ramamohanarao, K., and Tao, D. Learning with bounded instance-and label-dependent label noise. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 1789–1799, 2020. A

Cortes, C. and Vapnik, V. Support-vector networks. *Machine learning*, 20(3):273–297, 1995. 4.1

Del Moral, P., Ledoux, M., and Miclo, L. On contraction properties of markov kernels. *Probability theory and related fields*, 126(3):395–420, 2003. 1, 3.3

Diaconis, P. and Ylvisaker, D. Conjugate priors for exponential families. *The Annals of statistics*, pp. 269–281, 1979. 4.2

du Plessis, M. C., Niu, G., and Sugiyama, M. Analysis of learning from positive and unlabeled data. In *Advances in neural information processing systems*, pp. 703–711, 2014. A

El-Yaniv, R. and Wiener, Y. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11(5), 2010. 5

Feng, L., Shu, S., Lin, Z., Lv, F., Li, L., and An, B. Can cross entropy loss be robust to label noise? In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pp. 2206–2212, 2020. 5

Fergus, R., Fei-Fei, L., Perona, P., and Zisserman, A. Learning object categories from Google's image search. In *Tenth IEEE International Conference on Computer Vision*, volume 2, pp. 1816–1823, 2005. 1

Ghosh, A., Kumar, H., and Sastry, P. Robust loss functions under label noise for deep neural networks. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pp. 1919–1925, 2017. 5, (1)

Goldberger, J. and Ben-Reuven, E. Training deep neural-networks using a noise adaptation layer. In *International Conference on Learning Representations*, 2017. 1, A

Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. *Deep learning*. MIT press Cambridge, 2016. 1, 2.3

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 1321–1330, 2017. 1, 2.4, B

Han, B., Yao, J., Niu, G., Zhou, M., Tsang, I., Zhang, Y., and Sugiyama, M. Masking: A new perspective of noisy supervision. In *Advances in Neural Information Processing Systems*, pp. 5836–5846, 2018a. 1, (e), A, 5

Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I., and Sugiyama, M. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in Neural Information Processing Systems*, pp. 8527–8537, 2018b. 5, (c), 3

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016. 6.1, E.1, E.2

Hein, M., Andriushchenko, M., and Bitterwolf, J. Why ReLU networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 41–50, 2019. 1, 2.4, B

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017. 4.1

Ishida, T., Niu, G., Hu, W., and Sugiyama, M. Learning from complementary labels. In *Advances in neural information processing systems*, pp. 5639–5649, 2017. 2.2

Jiang, L., Zhou, Z., Leung, T., Li, L.-J., and Fei-Fei, L. MentorNet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 2304–2313, 2018. 5

Jiang, L., Huang, D., Liu, M., and Yang, W. Beyond synthetic noise: Deep learning on controlled noisy labels. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 4804–4815, 2020. 1

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations*, 2015. 4.2, 6.1, B, E.1

Kloft, M., Brefeld, U., Sonnenburg, S., Laskov, P., Müller, K., and Zien, A. Efficient and accurate ℓ_p-norm multiple kernel learning. In *Advances in Neural Information Processing Systems*, pp. 997–1005, 2009. 4.1

Krizhevsky, A. Learning multiple layers of features from tiny images, 2009. 6.1, 6

Kuhn, H., Tucker, A., et al. Nonlinear programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*. The Regents of the University of California, 1951. 4.1

Kull, M., Nieto, M. P., Kängsepp, M., Silva Filho, T., Song, H., and Flach, P. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. In *Advances in neural information processing systems*, pp. 12316–12326, 2019. 2.4, B

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition, 1998. 6.1, 5

Li, J., Wong, Y., Zhao, Q., and Kankanhalli, M. S. Learning to learn from noisy labeled data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5051–5059, 2019. 5

Li, J., Socher, R., and Hoi, S. C. DivideMix: Learning with noisy labels as semi-supervised learning. In *International Conference on Learning Representations*, 2020. 5

Li, X., Liu, T., Han, B., Niu, G., and Sugiyama, M. Provably end-to-end label-noise learning without anchor point. In *Proceedings of the 38th International Conference on Machine Learning*, 2021. 4.1, 5, D

Liu, T. and Tao, D. Classification with noisy labels by importance reweighting. *IEEE Transactions on pattern analysis and machine intelligence*, 38(3):447–461, 2015. 1, 2.4

Liu, Y. and Guo, H. Peer loss functions: Learning from noisy labels without knowing noise rates. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 6226–6236, 2020. 5

Long, P. M. and Servedio, R. A. Random classification noise defeats all convex potential boosters. *Machine learning*, 78(3):287–304, 2010. 1, A

Lu, N., Niu, G., Menon, A. K., and Sugiyama, M. On the minimal supervision for training any binary classifier from only unlabeled data. In *International Conference on Learning Representations*, 2019. A

Lyu, Y. and Tsang, I. W. Curriculum loss: Robust learning and generalization against label corruption. In *International Conference on Learning Representations*, 2020. 5

Ma, X., Huang, H., Wang, Y., Romano, S., Erfani, S., and Bailey, J. Normalized loss functions for deep learning with noisy labels. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 6543–6553, 2020. 5

Malach, E. and Shalev-Shwartz, S. Decoupling "when to update" from "how to update". In *Advances in Neural Information Processing Systems*, pp. 960–970, 2017. 5

Menon, A., Van Rooyen, B., Ong, C. S., and Williamson, B. Learning from corrupted binary labels via class-probability estimation. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 125–134, 2015. A

Menon, A. K., Van Rooyen, B., and Natarajan, N. Learning from binary labels with instance-dependent noise. *Machine Learning*, 107(8-10):1561–1595, 2018. A

Mirzasoleiman, B., Cao, K., and Leskovec, J. Coresets for robust training of deep neural networks against noisy labels. In *Advances in Neural Information Processing Systems*, 2020. 5

Mozannar, H. and Sontag, D. Consistent estimators for learning to defer to an expert. In *International Conference on Machine Learning*, pp. 7076–7087, 2020. 5

Natarajan, N., Dhillon, I. S., Ravikumar, P. K., and Tewari, A. Learning with noisy labels. In *Advances in neural information processing systems*, pp. 1196–1204, 2013. 1, 2.2, A

Nguyen, D. T., Mummadi, C. K., Ngo, T. P. N., Nguyen, T. H. P., Beggel, L., and Brox, T. SELF: Learning to filter noisy labels with self-ensembling. In *International Conference on Learning Representations*, 2020. 5

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. PyTorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pp. 8026–8037, 2019. 6.2, E.2

Patrini, G., Rozza, A., Krishna Menon, A., Nock, R., and Qu, L. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1944–1952, 2017. 1, 2.2, 2.3, 2.3, 2.4, 4.2, 5, (b), (4), 6.2, A, B, 2, E.2

Rahimi, A., Shaban, A., Cheng, C.-A., Hartley, R., and Boots, B. Intra order-preserving functions for calibration of multi-class neural networks. In *Advances in Neural Information Processing Systems*, 2020. 1, 2.4, B

Ramaswamy, H., Scott, C., and Tewari, A. Mixture proportion estimation via kernel embeddings of distributions. In *Proceedings of The 33rd International Conference on Machine Learning*, pp. 2052–2060, 2016. A

Scott, C. A rate of convergence for mixture proportion estimation, with application to learning from noisy labels. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, pp. 838–846, 2015. A

Scott, C., Blanchard, G., and Handy, G. Classification with asymmetric label noise: Consistency and maximal denoising. In *Proceedings of the 26th Annual Conference on Learning Theory*, pp. 489–511, 2013. A

Shu, J., Xie, Q., Yi, L., Zhao, Q., Zhou, S., Xu, Z., and Meng, D. Meta-Weight-Net: Learning an explicit mapping for sample weighting. In *Advances in Neural Information Processing Systems*, 2019. 5

Sutskever, I., Martens, J., Dahl, G., and Hinton, G. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pp. 1139–1147, 2013. 6.1

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

Thulasidasan, S., Bhattacharya, T., Bilmes, J., Chennupati, G., and Mohd-Yusof, J. Combating label noise in deep learning using abstention. In *International Conference on Machine Learning*, pp. 6234–6243, 2019. 5

Van Rooyen, B., Menon, A., and Williamson, R. C. Learning with symmetric label noise: The importance of being unhinged. In *Advances in Neural Information Processing Systems*, pp. 10–18, 2015. 1, A

Wang, Y., Liu, W., Ma, X., Bailey, J., Zha, H., Song, L., and Xia, S.-T. Iterative learning with open-set noisy labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8688–8696, 2018. 5

Wang, Y., Ma, X., Chen, Z., Luo, Y., Yi, J., and Bailey, J. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 322–330, 2019. 5

Wei, H., Feng, L., Chen, X., and An, B. Combating noisy labels by agreement: A joint training method with co-regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13726–13735, 2020. 5

Wu, P., Zheng, S., Goswami, M., Metaxas, D., and Chen, C. A topological filter for learning with label noise. In *Advances in Neural Information Processing Systems*, 2020. 5

Xia, X., Liu, T., Wang, N., Han, B., Gong, C., Niu, G., and Sugiyama, M. Are anchor points really indispensable in label-noise learning? In *Advances in Neural Information Processing Systems*, pp. 6838–6849, 2019. 1, 2.4, 2.4, 4.2, 5, (5), 6.2, A, B, E.2

Xia, X., Liu, T., Han, B., Wang, N., Gong, M., Liu, H., Niu, G., Tao, D., and Sugiyama, M. Part-dependent label noise: Towards instance-dependent label noise. In *Advances in Neural Information Processing Systems*, 2020. A

Xiao, T., Xia, T., Yang, Y., Huang, C., and Wang, X. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2691–2699, 2015. 6.2, A, E.2

Yao, Y., Liu, T., Han, B., Gong, M., Deng, J., Niu, G., and Sugiyama, M. Dual T: Reducing estimation error for transition matrix in label-noise learning. In *Advances in Neural Information Processing Systems*, 2020. 1, 2.4, 2.4, 4.2, 5, (6), A, E.1

Yu, X., Liu, T., Gong, M., and Tao, D. Learning with biased complementary labels. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 68–83, 2018. 1, 2.2, 2.3, 2.4, 5

Yu, X., Han, B., Yao, J., Niu, G., Tsang, I., and Sugiyama, M. How does disagreement help generalization against label corruption? In *Proceedings of the 36th International Conference on Machine Learning*, pp. 7164–7173, 2019. 5

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017. 1, 2.1

Zhang, Z. and Sabuncu, M. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Advances in neural information processing systems*, pp. 8778–8788, 2018. 5, (3)

A. Beyond Class-Conditional Noise

In this section, we provide an overview of several selected noise models.

As introduced in Section 1, early studies focused on the most simple case — the *random classification noise* (RCN) model for binary classification (Angluin & Laird, 1988; Long & Servedio, 2010; Van Rooyen et al., 2015), where binary labels are flipped independently with a fixed noise rate $\rho \in [0, 0.5)$. Here, $Y, \widetilde{Y} \in \{\pm 1\}$ and $\rho = p(\widetilde{Y} = -1|Y = +1) = p(\widetilde{Y} = -1|Y = +1)$. Then, still for binary classification, the *class-conditional noise* (CCN) model (Natarajan et al., 2013) extended the RCN model to the case where the noise rate depends on the class: $\rho_{+1} = p(\widetilde{Y} = -1|Y = +1)$, $\rho_{-1} = p(\widetilde{Y} = +1|Y = -1)$, $\rho_{+1} + \rho_{-1} < 1$. These noise models are special cases of the multiclass CCN model (Patrini et al., 2017; Goldberger & Ben-Reuven, 2017; Han et al., 2018a; Xia et al., 2019; Yao et al., 2020), which is the main focus of our work.

A more general framework for learning with label noise is the *mutual contamination* (MC) model (Scott et al., 2013; Blanchard & Scott, 2014; du Plessis et al., 2014; Menon et al., 2015; Lu et al., 2019), where examples of each class are drawn separately. That is, $p(X|Y)$ is corrupted but not $p(Y|X)$. Consequently, the marginal distribution of data may not match the true marginal distribution. It is known that CCN is a special case of MC (Menon et al., 2015). For the binary case, there is a related problem of the transition matrix estimation, called *mixture proportion estimation* (MPE) (du Plessis et al., 2014; Scott, 2015; Ramaswamy et al., 2016), which has more technical difficulties. Our method may not work well in the MC setting because it explicitly relies on the i.i.d. assumption.

Further, the *instance-dependent noise* (IDN) model (Menon et al., 2018; Cheng et al., 2020; Berthon et al., 2021) has been assessed to only a limited extent but is of great interest recently. IDN still explicitly models the label corruption process as CCN bu removes the CCN assumption in Eq. (1). Therefore, the noise transition matrix could be instance-dependent and thus harder to estimate. In other words, $\boldsymbol{T}$ is not a fixed matrix anymore but a matrix-valued function of the instance $\boldsymbol{T}(X) : \mathcal{X} \to \mathcal{T}$. Owing to its complexity, IDN has not been investigated extensively.

One simple way is to estimate the matrix-valued function $\boldsymbol{T}(X)$ and the clean class-posterior $p(Y|X)$ directly, assuming a certain level of smoothness of $\boldsymbol{T}(X)$ (Goldberger & Ben-Reuven, 2017). However, there is no theoretical guarantee and the estimation error could be very high. Another direction is to restrict the problem so we could provide some theoretical guarantees under certain conditions (Menon et al., 2018; Cheng et al., 2020). It is also a promising way to approximate IDN using a simpler dependency structure (Xiao et al., 2015; Xia et al., 2020), which works well in practice. Our method may also serve as a practical approximation of IDN without theoretical guarantee, which is reflected in the experiment on the Clothing1M dataset (Section 6). The use of regularization techniques in our work may inspire practical algorithm design for IDN.

Learning from noisy labels has a rich literature and there are several other noise models, e.g., capturing the uncertainty of labels without explicitly modeling the label corruption process. Although they are out of scope of our discussion.

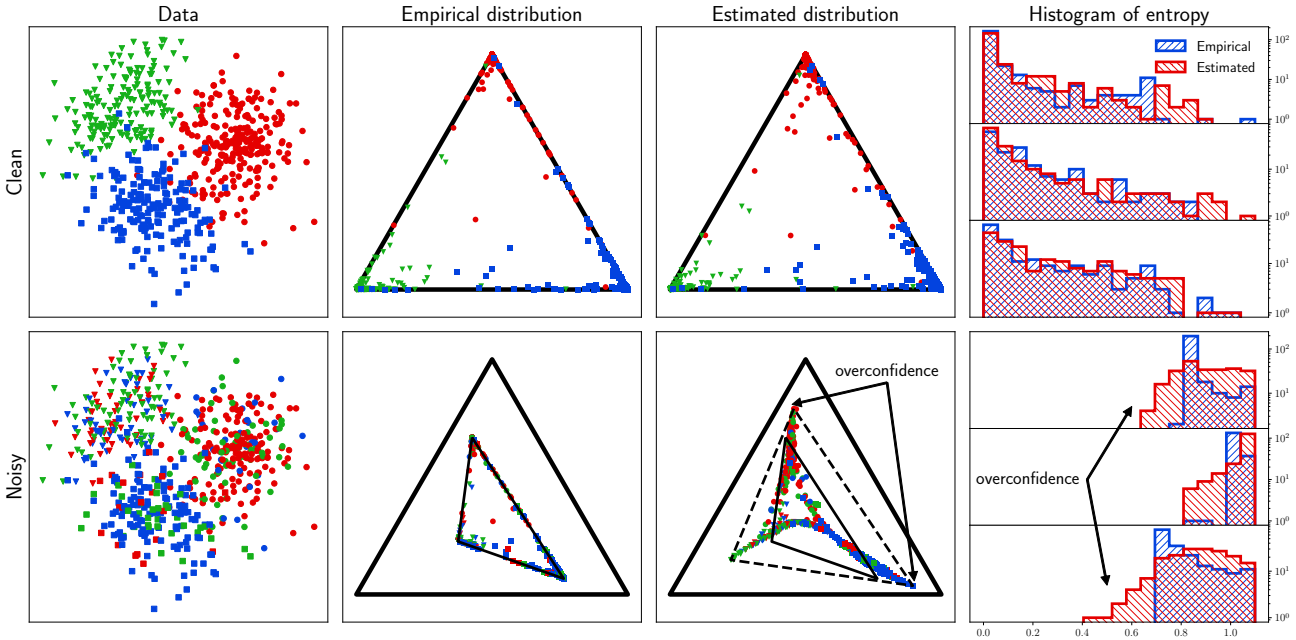

Figure 5. An example of **overconfident predictions** yield from neural networks, comparing the clean/noisy class-posterior estimation. See Fig. 3 for the notation.

B. Overconfidence in Neural Networks

In this section, we discuss the overconfidence phenomenon in neural networks, which is partially presented in Section 2.4.

In neural network training, if we only care the classification accuracy, the overconfidence is not a problem. We could use any *classification-calibrated* loss (Bartlett et al., 2006; Tewari & Bartlett, 2007), which only guarantees that the accuracy (0-1 risk) is asymptotically optimal. The class-posterior might not be recovered from the output of the classifier.

However, the transition matrix estimation based on anchor points (Patrini et al., 2017; Xia et al., 2019) — the method shown in Section 2.4 — heavily relies on a *confidence-calibrated* estimation of the noisy class-posterior using neural networks. The overconfidence issue of possibly over-parameterized neural networks has been discovered, and several re-calibration methods are developed to alleviate this issue (Guo et al., 2017; Kull et al., 2019; Hein et al., 2019; Rahimi et al., 2020).

In learning with class-conditional label noise, we can demonstrate that estimating the noise class-posterior causes a significantly worse overconfidence issue than estimating the clean class-posterior. Fig. 5 shows a comparison between the clean class-posterior estimation and the noisy class-posterior estimation (also shown in Fig. 3). We used a Gaussian mixture with 3 components as the training data, a 3-layer multilayer perceptron (MLP) with hidden layer size of 32 and rectified linear unit (ReLU) activations as the model, and an Adam optimizer (Kingma & Ba, 2015) with batch size of 64 and learning rate of 1×10^{-3}.

As discussed in Section 2.2, $\boldsymbol{p}(\widetilde{Y}|X)$ should be within the convex hull $\mathrm{Conv}(\boldsymbol{T})$. However, without knowing $\boldsymbol{T}$ and this constraint, a neural network trained with noisy labels tends to output overconfident probabilities that are outside of $\mathrm{Conv}(\boldsymbol{T})$. The lack of the constraint $\mathrm{Conv}(\boldsymbol{T})$ aggravates the overconfidence problem and might make it harder to re-calibrate the confidence. Consequently, transition matrix estimation may suffer from poorly estimated noisy class-posteriors, which leads to performance degradation of the aforementioned two-step methods.

This is the motivation of using the product $\widehat{\boldsymbol{T}}^{\mathsf{T}}\widehat{\boldsymbol{p}}(Y|X)$ as an estimate of $\boldsymbol{p}(\widetilde{Y}|X)$ and avoiding estimating $\boldsymbol{p}(\widetilde{Y}|X)$ directly using neural networks. However, it is important to note that the neural network may still suffer from the overconfidence issue, especially after we enforce the predicted probabilities to be more distinguishable from each other in terms of the pairwise total variation distance. In such a case, if the confidence of $\boldsymbol{p}(Y|X)$ is needed to make decisions, post-hoc re-calibration methods can be applied (Guo et al., 2017; Kull et al., 2019; Hein et al., 2019; Rahimi et al., 2020). Nevertheless, if we only use the accuracy as the evaluation metric, the overconfidence issue in the clean class-posterior estimation is much less harmful than it in the noisy class-posterior estimation, which affects the transition matrix estimation significantly.

C. Proof

In this section, we provide the proof of Theorem 2:

Theorem 2 (Consistency)**.** Given a finite i.i.d. sample of $(X, \widetilde{Y})$-pairs of size N, where anchor points (Definition 1) for each class exist in the sample, let $\widetilde{L}_0(W, \widehat{\boldsymbol{T}})$ and $\widetilde{R}(W)$ be the empirical estimates of $L_0(W, \widehat{\boldsymbol{T}})$ in Eq. (4) and $R(W)$ in Eq. (10), respectively. Assume that the parameter space $\mathcal{W}$ is compact. Let $(W^\circ, \widehat{\boldsymbol{T}}^\circ)$ be an optimal solution of the following constrained optimization problem:

$$\max_{W} \widetilde{R}(W) \text{ s.t. } \widetilde{L}_0(W, \widehat{\boldsymbol{T}}) = 0. \tag{11}$$

Then, $\widehat{\boldsymbol{T}}^\circ$ is a consistent estimator of the transition matrix $\boldsymbol{T}$; and $\widehat{\boldsymbol{p}}(Y|X; W^\circ) \xrightarrow{\text{d}} \boldsymbol{p}(Y|X)$ a.e. as $N \to \infty$.

First, recall Theorem 1:

Theorem 1 (Transition matrix decomposition)**.** For two row stochastic matrices $\boldsymbol{W}, \boldsymbol{V} \in \mathcal{T}$, if $\forall \boldsymbol{p} \in \Delta$, $\exists \boldsymbol{q} \in \Delta$, s.t. $\boldsymbol{W}^\mathsf{T}\boldsymbol{p} = \boldsymbol{V}^\mathsf{T}\boldsymbol{q}$, then $\exists$ a row stochastic matrix $\boldsymbol{U} \in \mathcal{T}$, s.t. $\boldsymbol{W} = \boldsymbol{U}\boldsymbol{V}$ and $\forall \boldsymbol{p} \in \Delta$, $\boldsymbol{q} = \boldsymbol{U}^\mathsf{T}\boldsymbol{p}$.

Proof. Let $\boldsymbol{p}$ be $\boldsymbol{e}_i$ and denote the corresponding $\boldsymbol{q}$ by $\boldsymbol{q}_i$ for $i = 1, \dots, K$. Then $\boldsymbol{U} = [\boldsymbol{q}_1, \dots, \boldsymbol{q}_K]^\mathsf{T}$. □

and the definitions of $L_0(W, \widehat{\boldsymbol{T}})$ and $R(W)$:

$$L_0(W, \widehat{\boldsymbol{T}}) := \mathop{\mathbb{E}}_{X \sim p(X)} \Big[D_{\text{KL}} \Big(\boldsymbol{p}(\widetilde{Y}|X) \,\Big\|\, \widehat{\boldsymbol{T}}^\mathsf{T} \widehat{\boldsymbol{p}}(Y|X; W) \Big) \Big], \tag{4}$$

$$R(W) := \mathop{\mathbb{E}}_{x_1 \sim p(X)} \mathop{\mathbb{E}}_{x_2 \sim p(X)} [d_{\text{TV}}(\widehat{\boldsymbol{p}}_1, \widehat{\boldsymbol{p}}_2)], \text{ where } \widehat{\boldsymbol{p}}_i := \widehat{\boldsymbol{p}}(Y|X = x_i; W), \quad i = 1, 2. \tag{10}$$

Also, recall that we considered a sufficiently large function class of $\widehat{\boldsymbol{p}}(Y|X; W)$ that contains the ground-truth $\boldsymbol{p}(Y|X)$, i.e., $\exists W^* \in \mathcal{W}, \widehat{\boldsymbol{p}}(Y|X; W^*) = \boldsymbol{p}(Y|X)$ a.e. Although there could be a set of W^* satisfying this condition, without loss of generality, we assume that W^* is unique.

Denote the set of W and $\widehat{\boldsymbol{T}}$ s.t. $L_0(W, \widehat{\boldsymbol{T}}) = 0$ by $(\mathcal{W} \times \mathcal{T})_0 \subset \mathcal{W} \times \mathcal{T}$. By definition, $(W^*, \boldsymbol{T}) \in (\mathcal{W} \times \mathcal{T})_0$.

Then, we have the following lemmas:

Lemma 1. $\forall (W, \widehat{\boldsymbol{T}}) \in (\mathcal{W} \times \mathcal{T})_0$, $\exists \boldsymbol{U} \in \mathcal{T}$, $\boldsymbol{p}(\widetilde{Y}|X) = \boldsymbol{T}^\mathsf{T}\boldsymbol{p}(Y|X) = \widehat{\boldsymbol{T}}^\mathsf{T}\widehat{\boldsymbol{p}}(Y|X; W) = \widehat{\boldsymbol{T}}^\mathsf{T}(\boldsymbol{U}^\mathsf{T}\boldsymbol{p}(Y|X))$ a.e.

Proof. This is the due to the *identity of indiscernibles* property of the KL-divergence and Theorem 1. □

Lemma 2. $\forall (W, \widehat{\boldsymbol{T}}) \in (\mathcal{W} \times \mathcal{T})_0$, $R(W) \leq R(W^*)$.

Proof. This is a direct consequence of Lemma 1, the contraction Eq. (9), and our assumption of the existence of W^*:

$$\begin{aligned}
R(W) &= \mathop{\mathbb{E}}_{x_1 \sim p(X)} \mathop{\mathbb{E}}_{x_2 \sim p(X)} [d_{\text{TV}}(\widehat{\boldsymbol{p}}(Y|X = x_1; W), \widehat{\boldsymbol{p}}(Y|X = x_2; W))] && (16)\\
&= \mathop{\mathbb{E}}_{x_1 \sim p(X)} \mathop{\mathbb{E}}_{x_2 \sim p(X)} [d_{\text{TV}}(\boldsymbol{U}^\mathsf{T}\boldsymbol{p}(Y|X = x_1), \boldsymbol{U}^\mathsf{T}\boldsymbol{p}(Y|X = x_2))] && (17)\\
&\leq \mathop{\mathbb{E}}_{x_1 \sim p(X)} \mathop{\mathbb{E}}_{x_2 \sim p(X)} [d_{\text{TV}}(\boldsymbol{p}(Y|X = x_1), \boldsymbol{p}(Y|X = x_2))] && (18)\\
&= \mathop{\mathbb{E}}_{x_1 \sim p(X)} \mathop{\mathbb{E}}_{x_2 \sim p(X)} [d_{\text{TV}}(\widehat{\boldsymbol{p}}(Y|X = x_1; W^*), \widehat{\boldsymbol{p}}(Y|X = x_2; W^*))] = R(W^*). && (19)
\end{aligned}$$

□

Lemma 3. $\sup_{W \in \mathcal{W}} \Big| R(W) - \widetilde{R}(W) \Big| \xrightarrow{\text{p}} 0$ and $\sup_{\substack{W \in \mathcal{W} \\ \widehat{\boldsymbol{T}} \in \mathcal{T}}} \Big| L_0(W, \widehat{\boldsymbol{T}}) - \widetilde{L}_0(W, \widehat{\boldsymbol{T}}) \Big| \xrightarrow{\text{p}} 0$ as $N \to \infty$.

Proof. This is due to the i.i.d. assumption, the compactness of $\mathcal{W}$ and $\mathcal{T}$, the continuity of $R(W)$ and $L_0(W, \widehat{\boldsymbol{T}})$, and the *uniform law of large numbers*. □

Finally, by the definition and Lemma 3, we have $\mathbb{P}\Big[L_0(W^\circ, \widehat{\boldsymbol{T}}^\circ) = 0 \Big] \to 1$, and by Lemmas 2 and 3, we have $W^\circ \to W^*$ as $N \to \infty$. Therefore, $\widehat{\boldsymbol{p}}(Y|X; W^\circ) \xrightarrow{\text{d}} \boldsymbol{p}(Y|X)$ a.e., which means that the corresponding $\boldsymbol{U} \to \boldsymbol{I}$ and thus $\widehat{\boldsymbol{T}}^\circ \to \boldsymbol{T}$ as $N \to \infty$. ■

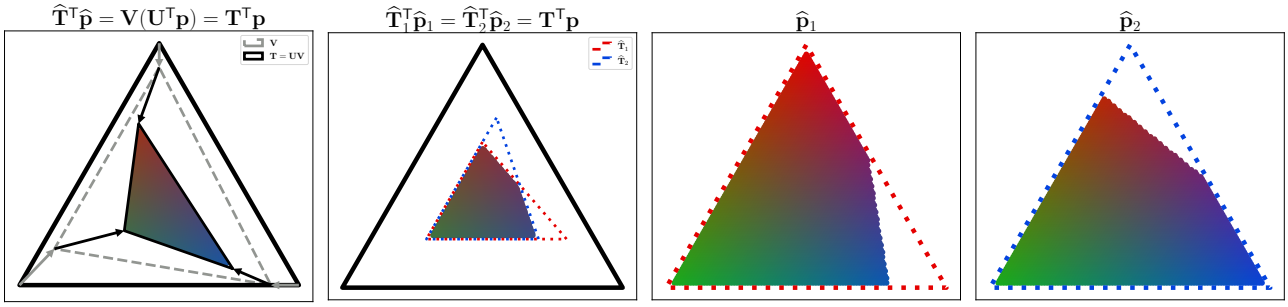

Figure 6. The intuition behind the theoretical results.

D. Intuition

In this section, we discuss the intuition behind our theoretical results.

Given the only constraint of $\boldsymbol{p}(\widetilde{Y}|X) = \widehat{\boldsymbol{T}}^\top\widehat{\boldsymbol{p}}(Y|X)$, a feasible $\widehat{\boldsymbol{T}}$ can be any matrix that satisfies

$$\forall x \in \mathcal{X}, \boldsymbol{p}(\widetilde{Y}|X = x) \in \mathrm{Conv}(\widehat{\boldsymbol{T}}), \tag{20}$$

and we can find $\widehat{\boldsymbol{p}}(Y|X)$ accordingly, if the function class is sufficiently large. This is the partial identifiability problem. On the other hand, in real-world problems, we hope that the clean class-posterior $\boldsymbol{p}(Y|X)$ is the "cleannest", so at least

$$\nexists \boldsymbol{S} \in \mathcal{T}, \text{ s.t. } \boldsymbol{p}(Y|X) = \boldsymbol{S}^\top\boldsymbol{p}'(Y|X). \tag{21}$$

Otherwise, $\boldsymbol{p}(\widetilde{Y}|X) = (\boldsymbol{S}\boldsymbol{T})^\top\boldsymbol{p}'(Y|X)$ also holds and $\boldsymbol{p}'(Y|X)$ might be a better solution. Thus, we become less ambitious, ignore all intermediate possible solutions and only aim to find the "cleannest" one.

However, there could be multiple "cleannest" ones in this sense. An example is illustrated in Fig. 6. There could be $\widehat{\boldsymbol{T}}_1$, $\widehat{\boldsymbol{T}}_2$, and $\widehat{\boldsymbol{p}}_1(Y|X)$, $\widehat{\boldsymbol{p}}_2(Y|X)$, such that

$$\boldsymbol{p}(\widetilde{Y}|X) = \widehat{\boldsymbol{T}}_1^\top\widehat{\boldsymbol{p}}_1(Y|X) = \widehat{\boldsymbol{T}}_2^\top\widehat{\boldsymbol{p}}_2(Y|X), \tag{22}$$

and both $\widehat{\boldsymbol{p}}_1(Y|X)$ and $\widehat{\boldsymbol{p}}_2(Y|X)$ satisfy the condition above. In this sense, we still cannot distinguish $\widehat{\boldsymbol{p}}_1(Y|X)$ and $\widehat{\boldsymbol{p}}_2(Y|X)$. Either of them can be the true clean class-posterior.

To avoid such cases, in this work, we made the assumption that anchor points exist, i.e., there are instances for each class that we are absolutely sure which class they belong to. Such instances are considered prototypes of each class, and we believe that they exist in many real-world noisy datasets. In this way, we can guarantee the uniqueness of the "cleannest" clean class-posterior and the transition matrix, and consequently construct consistent estimators to find them, as explained in this paper.

If anchor points for all classes do not exist, the proposed algorithm may still work in practice but there is no theoretical guarantee yet. As mentioned in Section 5, Li et al. (2021) aims to relax the anchor point assumption. From the perspective of the geometric property of the transition matrix, it is possible to solve this problem in a weaker condition, which is, however, not the focus of this work.

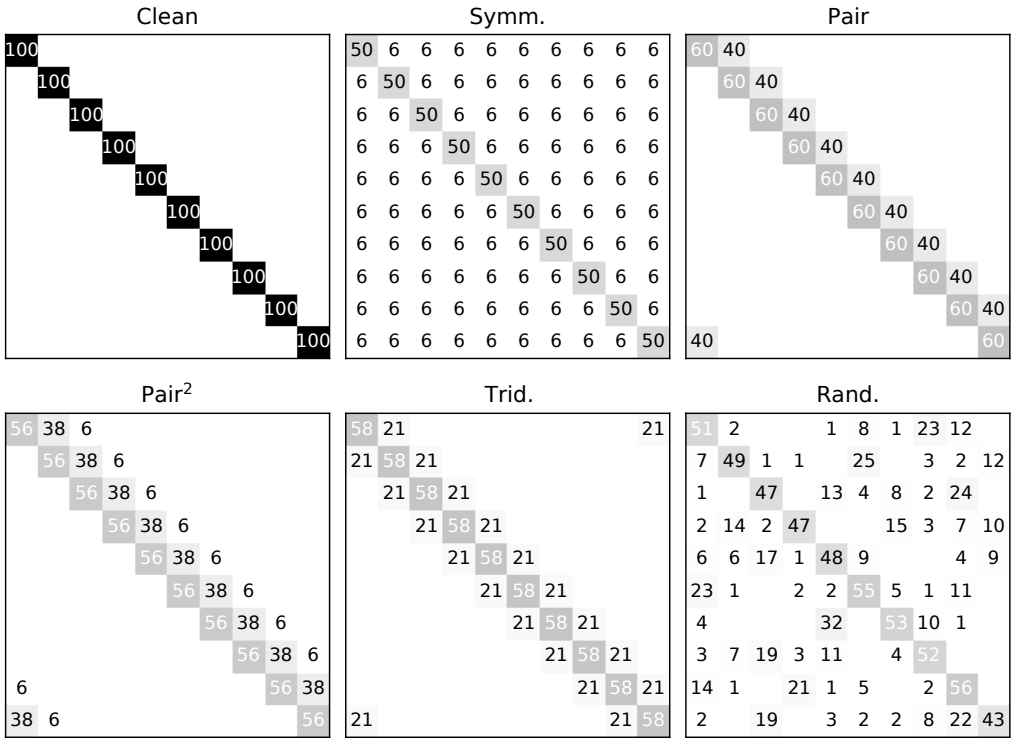

Figure 7. **Synthetic transition matrices** ($\times 100$) used in our experiments when $K = 10$ (MNIST and CIFAR-10). The random noise matrix (Rand.) is an example while other matrices are fixed.

E. Experiments

In this section, we provide missing details of the experimental settings used in Section 6.

E.1. Benchmark Datasets

Data. We used the MNIST,[5] CIFAR-10, and CIFAR-100[6] datasets. The MNIST dataset contains 28×28 grayscale images in 10 classes. The size of the training set is 60000 and the size of the test set is 10000. The CIFAR-10 and CIFAR-100 datasets contain 32×32 colour images in 10 classes and in 100 classes, respectively. The size of the training set is 50000 and the size of the test set is 10000.

Data preprocessing. For MNIST, We did not use any data augmentation. For CIFAR-10 and CIFAR-100, we used random crop and random horizontal flip. We added synthetic label noise into the training sets. The test sets were not modified. Overall, the transition matrices are plotted in Fig. 7. More specifically, we used:

1. (**Clean**) no additional synthetic noise, which serves as a baseline for the dataset and model.
2. (**Symm.**) symmetric noise with noise rate 50% (Patrini et al., 2017).
3. (**Pair**) pair flipping noise with noise rate 40% (Han et al., 2018b).
4. (**Pair^2**) a product of two pair flipping noise matrices with noise rates 30% and 20%. Because the multiplication of pair flipping noise matrices is commutative, it is guaranteed to have multiple ways of decomposition of the transition matrix, e.g., $\boldsymbol{T}_{\text{pair}}(30\%)\boldsymbol{T}_{\text{pair}}(20\%) = \boldsymbol{T}_{\text{pair}}(20\%)\boldsymbol{T}_{\text{pair}}(30\%)$. The overall noise rate is 44%.
5. (**Trid.**) tridiagonal noise (see also Han et al., 2018a), which corresponds to a spectral of classes where adjacent classes are easier to be *mutually* mislabeled, unlike the *unidirectional* pair flipping. It can be implemented by two consecutive

[5] MNIST (LeCun et al., 1998) http://yann.lecun.com/exdb/mnist/
[6] CIFAR-10, CIFAR-100 (Krizhevsky, 2009) https://www.cs.toronto.edu/~kriz/cifar.html

pair flipping transformations in the opposite direction. We used $\boldsymbol{T}_{\text{pair}}(30\%)\boldsymbol{T}_{\text{pair}}(30\%)^{\mathsf{T}}$ in the experiment. The overall noise rate is 42%. Strictly, the matrix is not a tridiagonal matrix in the conventional sense because $\boldsymbol{T}_{1,K}$ and $\boldsymbol{T}_{K,1}$ are non-zero.

6. (**Rand.**) random noise constructed by sampling a Dirichlet distribution and mixing with the identity matrix to a specified noise rate. The higher the concentration parameter of the Dirichlet distribution is, the more uniform the off-diagonal elements of the transition matrix are. We used 0.5 in the experiment. Then, we mixed the sampled matrix with the identity matrix linearly to make the overall noise rate 50%. The transition matrix is sampled for each trial.

Models. For MNIST, we used a sequential convolutional neural network with the following structure: `Conv2d`(channel=32) $\times 2$, `Conv2d`(channel=64) $\times 2$, `MaxPool2d`(size=2), `Linear`(dim=128), `Dropout`(p=0.5), `Linear`(dim=10). The kernel size of convolutional layers is 3, and rectified linear unit (ReLU) is applied after the convolutional layers and linear layers except the last one. For both CIFAR-10 and CIFAR-100, we used a ResNet-18 model (He et al., 2016).

Optimization. For MNIST, we used an Adam optimizer (Kingma & Ba, 2015) with batch size of 512 and learning rate of 1×10^{-3}. The model was trained for 2000 iterations (17.07 epochs) and the learning rate decayed exponentially to 1×10^{-4}. For CIFAR-10 and CIFAR-100, we used a stochastic gradient descent (SGD) optimizer with batch size of 512, momentum of 0.9, and weight decay of 1×10^{-4}. The learning rate increased from 0 to 0.1 linearly for 400 iterations and decreased to 0 linearly for 3600 iterations (4000 iterations/40.96 epochs in total).

For the gradient-based estimation, we used an Adam optimizer (Kingma & Ba, 2015). The learning rate increased from 0 to 5×10^{-3} linearly for 400 iterations and creased to 0 linearly for the rest iterations. This is helpful because at earlier stage, the model was not sufficiently trained yet and changing the transition matrix too much may destabilize the training of the model.

We tuned hyperparameters using grid search on a small experiment and fixed them in all experimental settings. For better reusability, we assumed that we are noise-agnostic and did not fine-tune hyperparameters for each noise type. If we have more prior knowledge about the noise, a better initialization may further improve the performance.

Infrastructure. The experiments were conducted on NVIDIA Tesla P100 GPUs. We used a single GPU for MNIST and data-parallel on 2 GPUs for CIFAR-10 and CIFAR-100.

Results. In addition to the accuracy and average total variation presented in Tables 1 and 2, we also provide the heat maps of the estimated transition matrices in Figs. 8 to 10. Extremely small numbers are hided for better demonstration.

We can observe that our proposed method, especially the Dirichlet posterior update method, usually has better estimation of the transition matrix under various noise types. Dual-T (Yao et al., 2020) also performs well in some settings, which is also reflected in Table 2.

E.2. Clothing1M

Data. Clothing1M (Xiao et al., 2015) is a real-world noisy label dataset. It contains 47570 clean training images, 1×10^6 ($1M$) noisy training images, 14313 clean validation images, and 10526 clean test images in 14 classes. We only used the noisy training data and clean test data.

Model and optimization. We followed previous work (Patrini et al., 2017; Xia et al., 2019). We used a ResNet-50 model (He et al., 2016) pretrained on ImageNet and a SGD optimizer with momentum of 0.9, weight decay of 1×10^{-3}, and batch size of 32. We trained the model on 64 GPUs for 5000 iterations (10.24 epochs in total). The learning rate was 1×10^{-3} for the first half and 1×10^{-4} for the second half.

Other hyperparameters. We initialized the concentration matrix with diagonal elements of 1 and off-diagonal elements of 0. We set $\boldsymbol{\beta} = (0.999, 0.01)$ and $\gamma = 0.1$.

Infrastructure. We implemented data-parallel distributed training on 64 NVIDIA Tesla P100 GPUs by PyTorch (Paszke et al., 2019). The average runtime is about 15 (without `SyncBatchNorm`) to 25 (with `SyncBatchNorm`) minutes.

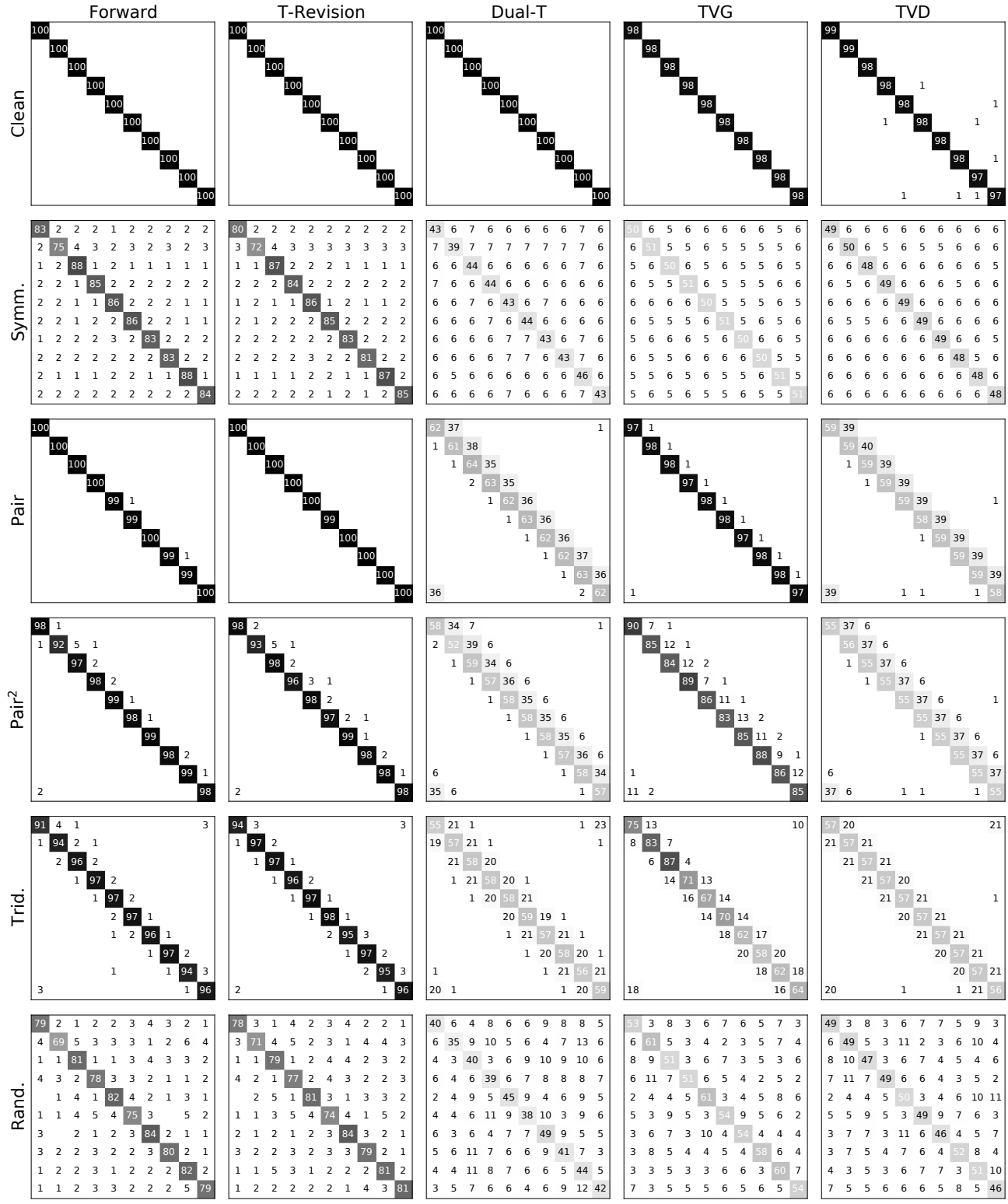

Figure 8. **Estimated transition matrices** ($\times 100$) on MNIST.

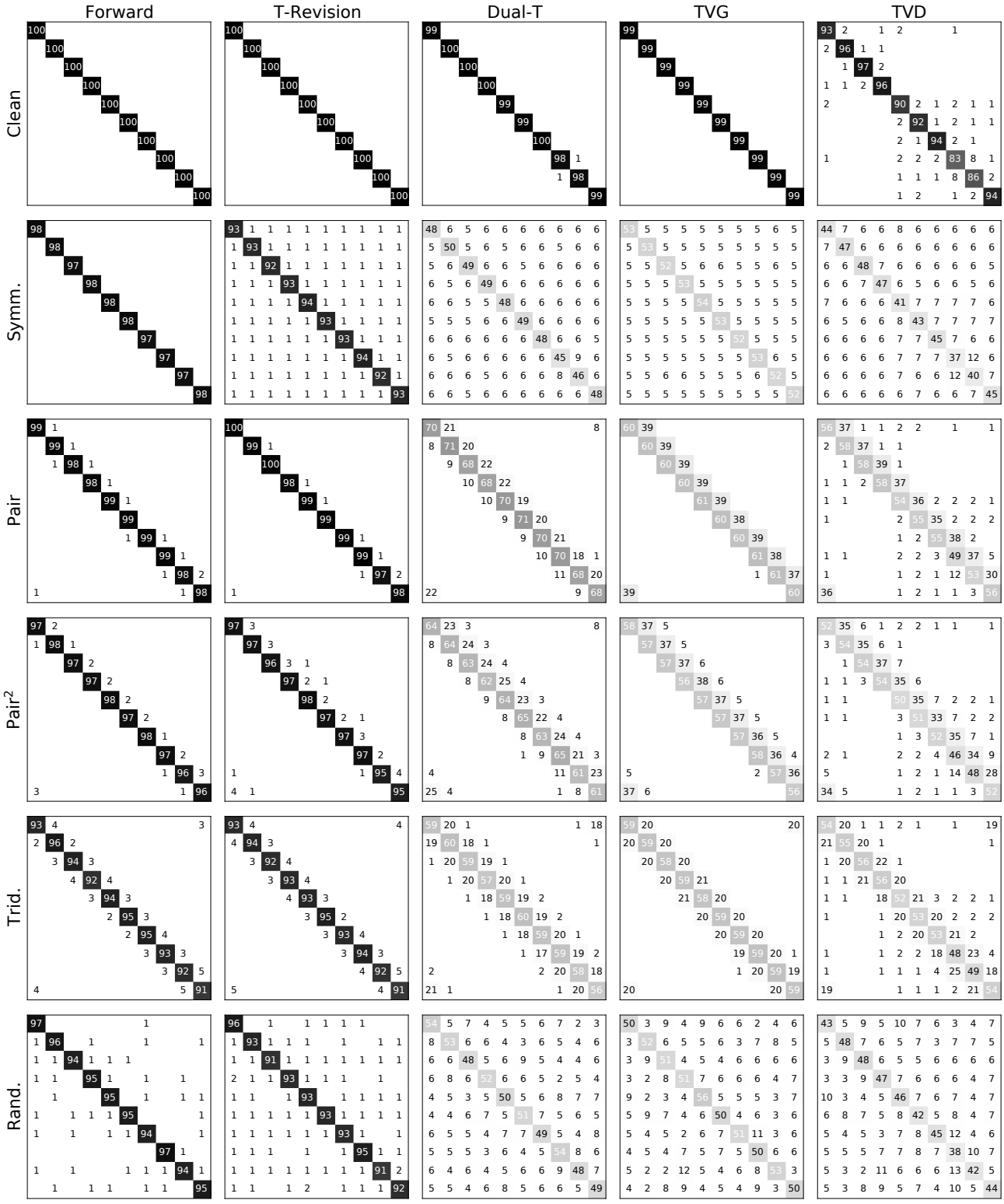

Figure 9. **Estimated transition matrices** ($\times 100$) on CIFAR10.

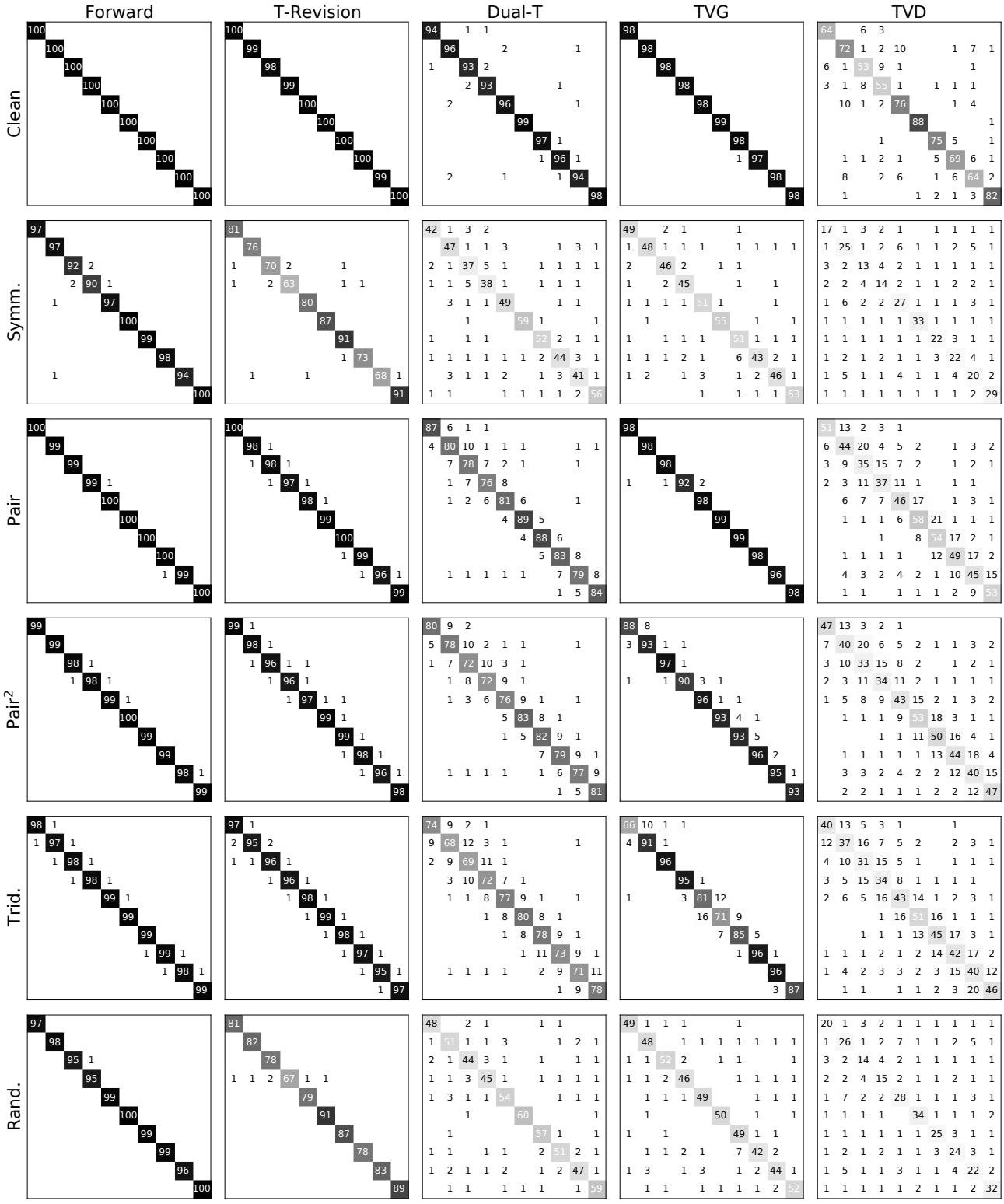

Figure 10. **Estimated transition matrices** ($\times 100$) on CIFAR100 (first 10 classes).