Reduction from Complementary-Label Learning to Probability Estimates

Wei-I Lin and Hsuan-Tien Lin
National Taiwan University, Taipei, Taiwan
{r10922076,htlin}@csie.ntu.edu.tw

Abstract. Complementary-Label Learning (CLL) is a weakly-supervised learning problem that aims to learn a multi-class classifier from only complementary labels, which indicate a class to which an instance does not belong. Existing approaches mainly adopt the paradigm of reduction to ordinary classification, which applies specific transformations and surrogate losses to connect CLL back to ordinary classification. Those approaches, however, face several limitations, such as the tendency to overfit. In this paper, we sidestep those limitations with a novel perspective—reduction to probability estimates of complementary classes. We prove that accurate probability estimates of complementary labels lead to good classifiers through a simple decoding step. The proof establishes a reduction framework from CLL to probability estimates. The framework offers explanations of several key CLL approaches as its special cases and allows us to design an improved algorithm that is more robust in noisy environments. The framework also suggests a validation procedure based on the quality of probability estimates, offering a way to validate models with only CLs. The flexible framework opens a wide range of unexplored opportunities in using deep and non-deep models for probability estimates to solve CLL. Empirical experiments further verified the framework’s efficacy and robustness in various settings.

Keywords: complementary-label learning · weakly-supervised learning

1 Introduction

In real-world machine learning applications, high-quality labels may be hard or costly to collect. To conquer the problem, researchers turn to the weakly-supervised learning (WSL) framework, which seeks to learn a good classifier with incomplete, inexact, or inaccurate data [14]. This paper focuses on a very weak type of WSL, called complementary-label learning (CLL) [3]. For the multi-class classification task, a complementary label (CL) designates a class to which a specific instance does not belong. The CLL problem assumes that the learner receives complementary labels rather than ordinary ones during training, while wanting the learner to correctly predict the ordinary labels of the test instances. Complementary labels can be cheaper to obtain. For example, when

1 The full paper can be accessed at https://arxiv.org/abs/2209.09500
labeling with many classes, selecting the correct label is time-consuming for data annotators, while selecting a complementary label would be less costly \[3\]. In this case, fundamental studies on CLL models can potentially upgrade multi-class classification models and make machine learning more realistic. CLL’s usefulness also attracts researchers to study its interaction with other tasks, such as generative-discriminative learning \[7,10\] and domain-adaptation \[13\].

\[3,4\] proposed a pioneering model for CLL based on replacing the ordinary classification error with its unbiased risk estimator (URE) computed from only complementary labels assuming that the CLs are generated uniformly. \[1\] unveiled the overfitting tendency of URE and proposed the surrogate complementary loss (SCL) as an alternative design. \[11\] studied the situation where the CLs are not generated uniformly, and proposed a loss function that includes a transition matrix for representing the non-uniform generation. \[2\] argued that the non-uniform generation shall be tackled by being agnostic to the transition matrix instead of including the matrix in the loss function.

The methods mentioned above mainly focused on applying transformation and specific loss functions to the ordinary classifiers. Such a “reduction to ordinary classification” paradigm, however, faces some limitations and is not completely analyzed. For instance, so far most of the methods in the paradigm require differentiable models such as neural networks in their design. It is not clear whether non-deep models could be competitive or even superior to deep ones. It remains critical to correct the overfitting tendency caused by the stochastic relationship between complementary and ordinary labels, as repeatedly observed on URE-related methods \[1\]. More studies are also needed to understand the potential of and the sensitivity to the transition matrix in the non-uniform setting, rather than only fixing the matrix in the loss function \[11\] or dropping it \[2\].

The potential limitations from reduction to ordinary classification motivate us to sidestep them by taking a different perspective—reduction to complementary probability estimates. Our contribution can be summarized as follows.

1. We propose a framework that only relies on the probability estimates of CLs, and prove that a simple decoding method can map those estimates back to correct ordinary labels with theoretical guarantees.
2. The proposed framework offers explanations of several key CLL approaches as its special cases and allows us to design an improved algorithm that is more robust in noisy environments.
3. We propose a validation procedure based on the quality of probability estimates, providing a novel approach to validate models with only CLs along with theoretical justifications.
4. We empirically verify the effectiveness of the proposed framework under broader scenarios than previous works that cover various assumptions on the CL generation (uniform/non-uniform; clean/noisy) and models (deep/non-deep). The proposed framework improves the SOTA methods in those scenarios, demonstrating the effectiveness and robustness of the framework.
2 Problem Setup

In this section, we first introduce the problem of ordinary multi-class classification, then formulate the CLL problem, and introduce some common assumption.

2.1 Ordinary-label learning

We start by reviewing the problem formulation of ordinary multi-class classification. In this problem, we let $K$ with $K > 2$ denote the number of classes to be classified, and use $\mathcal{Y} = [K] = \{1, 2, \ldots, K\}$ to denote the label set. Let $\mathcal{X} \subset \mathbb{R}^d$ denote the feature space. Let $D$ be an unknown joint distribution over $\mathcal{X} \times \mathcal{Y}$ with density function $p_D(x, y)$. Given $N$ i.i.d. training samples $\{(x_i, y_i)\}_{i=1}^N$ and a hypothesis set $\mathcal{H}$, the goal of the learner is to select a classifier $f: \mathcal{X} \rightarrow \mathbb{R}^K$ from the hypothesis set $\mathcal{H}$ that predicts the correct labels on unseen instances.

The prediction $\hat{y}$ of an unseen instance $x$ is determined by taking the argmax function on $f$, i.e. $\hat{y} = \text{argmax}_i f_i(x)$, where $f_i(x)$ denote the $i$-th output of $f(x)$. The goal of the learner is to learn an $f$ from $\mathcal{H}$ that minimizes the following classification risk: $E_{(x,y) \sim D}[\ell(f(x), e_y)]$, where $\ell: \mathbb{R}^K \times \mathbb{R}^K \rightarrow \mathbb{R}^+$ denotes the loss function, and $e_y$ denote the one-hot vector of label $y$.

2.2 Complementary-label learning

In complementary-label learning, the goal for the learner remains to find an $f$ that minimizes the ordinary classification risk. The difference lies in the dataset to learn from. The complementary learner does not have access to the ground-truth labels $y_i$. Instead, for each instance $x_i$, the learner is given a complementary label $\bar{y}_i$. A complementary label is a class that $x_i$ does not belong to; that is, $\bar{y}_i \in [K]\{y_i\}$. In CLL, it is assumed that the complementary dataset is generated according to an unknown distribution $\bar{D}$ over $\mathcal{X} \times \mathcal{Y}$ with density function $\bar{p}_D(x, y)$. Given access to i.i.d. samples $\{(x_i, \bar{y}_i)\}_{i=1}^N$ from $\bar{D}$, the complementary-label learner aims to find a hypothesis that classifies the correct ordinary labels on unseen instances.

Next, we introduce the class-conditional complementary transition assumption, which is used by many existing work [2][4][11]. It assumes that the generation of complementary labels only depends on the ordinary labels; that is, $P(\bar{y} \mid y, x) = P(\bar{y} \mid y)$. The transition probability $P(\bar{y} \mid y)$ is often represented by a $K \times K$ matrix, called transition matrix, with $T_{ij} = P(\bar{y}_j = y = i)$. It is commonly assumed to be all-zeros on the diagonals, i.e., $T_{ii} = 0$ for all $i \in [K]$ in CLL because complementary labels are not ordinary. The transition matrix is further classified into two categories: (a) Uniform: In uniform complementary generation, each complementary label is sampled uniformly from all labels except the ordinary one. The transition matrix in this setting is accordingly $T = \frac{1}{K-1}(1_k - I_k)$. This is the most widely researched and benchmarked setting in CLL. (b) Biased: A biased complementary generation is one that is not uniform. Biased transition matrices could be further classified as invertible ones and noninvertible ones based on its invertibility. The invertibility of a transition matrix comes with less...
Table 1. Comparison of recent approaches to CLL. $f(x)$ is the probability estimates of $x$, and $\ell$ is an arbitrary multi-class loss.

| Method | Transformation | Loss Function |
|--------|----------------|---------------|
| URE \[3,4\] | $\phi = I$ | $-(K-1)\ell(f(x), \bar{y}) + \sum_{k=1}^{K} \ell(f(x), k)$ |
| SCL-NL \[1\] | $\phi = I$ | $-\log(1 - f_\bar{y}(x))$ |
| Fwd \[11\] | $\phi(f)(x) = T^T f(x)$ | $\ell(\phi(f)(x), \bar{y})$ |
| DM \[2\] | $\phi(f)(x) = \text{sm}(1 - f(x))$ | $\ell(\phi(f)(x), \bar{y})$ |

physical meaning in the context of CLL; however, it plays an important role in some theoretical analysis in previous work \[1,11\].

Following earlier approaches, we assume that the generation of complementary labels follows class-conditional transition in the rest of the paper and that the transition matrix is given to the learning algorithms. What is different is that we do not assume the transition matrix to be uniform nor invertible. This allows us to make comparison in broader scenarios. In real-world scenarios, the true transition matrix may be impossible to access. To loosen the assumption that the true transition matrix is given, we will analyze the case that the given matrix is inaccurate later. This analysis can potentially help us understand the CLL in a more realistic environment.

3 Proposed Framework

In this section, we propose a framework for CLL based on complementary probability estimates (CPE) and decoding. We first motivate the proposed CPE framework in Section 3.1. Then, we describe the framework and derive its theoretical properties in Section 3.2. In Section 3.3, we explain how earlier approaches can be viewed as special cases in CPE. We further draw insights for earlier approaches through CPE and propose improved algorithms based on those insights.

3.1 Motivation

To conquer CLL, recent approaches \[1,4,11\] mainly focus on applying different transformation and surrogate loss functions to the ordinary classifier, as summarized in Table 1. This paradigm of reduction to ordinary, however, faces some limitations. For instance, as \[1\] points out, the URE approach suffers from the large variance in the gradients. Besides, it remains unclear how some of them behave when the transition matrix is biased. Also, those methods only studied using neural networks and linear models as base models. It is unclear how to easily cast other traditional models for CLL. These limitations motivate us to sidestep them with a different perspective—reduction to complementary probability estimates.

3.2 Methodology

Overview The proposed method consists of two steps: In training phase, we aim to find a hypothesis $\hat{f}$ that predicts the distribution of complementary labels
well, i.e., an $\tilde{f}$ that approximates $P(\bar{y} \mid x)$. This step is motivated by [2,11], which involve modeling the conditional distribution of the complementary labels $P(\bar{y} \mid x)$, and [12], which uses similar idea on noisy-label learning. What is different in our framework is the decoding step during prediction. In inference phase, we propose to predict the label with the closest transition vector to the predicted complementary probability estimates. Specifically, we propose to predict $\hat{y} = \arg\min_{k \in [K]} d(\tilde{f}(x), T_k)$ for an unseen instance $x$, where $d$ denotes a loss function. It is a natural choice to decode with respect to $T_k$ because the transition vector $T_k = (P(\bar{y} = 1 \mid y = k), \ldots, P(\bar{y} = K \mid y = k))^\top$ is the ground-truth distribution of the complementary labels if the ordinary label is $k$. In the following paragraph, we provide further details of our framework.

**Training Phase: Probability Estimates** In this phase, we aim to find a hypothesis $\tilde{f}$ that predicts $P(\bar{y} \mid x)$ well. To do so, given a hypothesis $\tilde{f}$ from hypothesis set $\tilde{H}$, we set the following complementary estimation loss to optimize:

$$R(\tilde{f}; \ell) = \mathbb{E}_{(x,y) \sim \mathcal{D}} (\ell(\tilde{f}(x), P(\bar{y} \mid x, y)))$$  \hspace{1cm} (1)

where $\ell$ can be any loss function defined between discrete probability distributions. By the assumption that complementary labels are generated with respect to the transition matrix $T$, the ground-truth distribution for $P(\bar{y} \mid x, y)$ is $T_y$, so we can rewrite Equation (1) as follows:

$$R(\tilde{f}; \ell) = \mathbb{E}_{(x,y) \sim \mathcal{D}} (\ell(\tilde{f}(x), T_y))$$  \hspace{1cm} (2)

The loss function above is still hard to optimize for two reasons: First, the presence of ordinary label $y$ suggests that it cannot be accessed from the complementary dataset. Second, as we only have one complementary label per instance, it becomes questionable to directly use the empirical density, i.e., the one-hot vector of the complementary label $e_{\bar{y}}$ to approximate $T_y$ as it may change the objective.

Here we propose to use the Kullback-Leibler divergence for the loss function to solve the two issues mentioned above with the following property:

**Proposition 1.** There is a constant $C$ such that

$$\mathbb{E}_{(x,\bar{y}) \sim \mathcal{D}} \ell(\tilde{f}(x), e_{\bar{y}}) + C = \mathbb{E}_{(x,y) \sim \mathcal{D}} \ell(\tilde{f}(x), T_y)$$  \hspace{1cm} (3)

holds for all hypothesis $\tilde{f} \in \tilde{H}$ if $\ell$ is the KL divergence, i.e., $\ell(\hat{y}, y) = \sum_{k=1}^{K} -y_k (\log \hat{y}_k - \log y_k)$.

The result is well-known in the research of proper scoring rules [4,9]. It allows us to replace the $T_y$ by $e_{\bar{y}}$ in Equation (2) because the objective function only differs by a constant after the replacement. This suggests that minimizing the two objectives is equivalent. Moreover, the replacement makes the objective function accessible through the complementary dataset because it only depends on the complementary label $\bar{y}$ rather than the ordinary one.
Formally speaking, minimizing Equation (2) becomes equivalent to minimizing the following surrogate complementary estimation loss (SCEL):

\[
\bar{R}(\bar{f}; \ell) = \mathbb{E}_{(x, \bar{y}) \sim \bar{D}} \left( \ell(\bar{f}(x), e_{\bar{y}}) \right)
\]

(4)

By using KL divergence as the loss function, we have that

\[
\bar{R}(\bar{f}; \ell) = \mathbb{E}_{(x, \bar{y}) \sim \bar{D}} \left( - \log \bar{f}_{\bar{y}}(x) \right)
\]

(5)

with \( \bar{f}_{\bar{y}}(x) \) being the \( \bar{y} \)-th output of \( \bar{f}(x) \). Next, we can use the following empirical version as the training objective:

\[
\frac{1}{N} \sum_{i=1}^{N} - \log \bar{f}_{\bar{y}_i}(x_i)
\]

According to the empirical risk minimization (ERM) principle, we can estimate the distribution of complementary labels \( P(\bar{y} | x) \) by minimizing the log loss on the complementary dataset. That is, by choosing \( \bar{f}^* \) with \( \bar{f}^* = \arg\min_{\bar{f} \in \bar{H}} \frac{1}{N} \sum_{i=1}^{N} - \log \bar{f}_{\bar{y}_i}(x_i) \), we can get an estimate of \( P(\bar{y} | x) \) with \( \bar{f}^* \).

In essence, we reduce the task of learning from complementary labels into learning probability estimates for multi-class classification (on the complementary label space). As the multi-class probability estimates is a well-researched problem, our framework becomes flexible on the choice of the hypothesis set. For instance, one can use K-Nearest Neighbor or Gradient Boosting with log loss to estimate the distribution of complementary labels. The flexibility becomes superior to the previous methods, who mainly focus on using neural networks to minimize specific surrogate losses. It makes them hard to optimize for non-differentiable models. In contrast, the proposed methods directly enable existing ordinary models to learn from complementary labels.

**Inference Phase: Decoding** After finding a complementary probability estimator \( \bar{f}^* \) during the training phase, we propose to predict the ordinary label by decoding: Given an unseen example \( x \), we predict the label \( \hat{y} \) whose transition vector \( T_y \) is closest to the predicted complementary probability estimates. That is, the label is predicted by

\[
\hat{y} = \arg\min_{k \in [K]} d \left( \bar{f}^*(x), T_k \right)
\]

(6)

where \( d \) could be an arbitrary loss function on the probability simplex and \( T_k \) is the \( k \)-th row vector of \( T \). We use \( \text{dec}(\bar{f}; d) \) to denote the function that decodes the output from \( \bar{f} \) according to the loss function \( d \). The next problem is whether the prediction of the decoder can guarantee a small out-sample classification error \( R_{01}(f) = \mathbb{E}_{(x,y) \sim D} I_{f(x) \neq y} \).

We propose to use a simple decoding step by setting \( L_1 \) distance as the loss function for decoding:

\[
\text{dec}(\bar{f}; L_1)(x) = \arg\min_{y \in [K]} \| T_y - \bar{f}(x) \|_1
\]

(7)

This choice of \( L_1 \) distance makes the decoding step easy to perform and provides the following bound that quantifies the relationship between the error rate and the quality of probability estimator:
Proposition 2. For any $\bar{f} \in \mathcal{H}$, and distance function $d$ defined on the probability simplex $\Delta^K$, it holds that

$$R_{01}(\text{dec}(\bar{f};d)) \leq \frac{2}{\gamma_d} R(\bar{f};d)$$  \hspace{1cm} (8)

where $\gamma_d = \min_{i \neq j} d(T_i, T_j)$ is the minimal distance between any pair of transition vectors. Moreover, if $d$ is the $L_1$ distance and $\ell$ is the KL divergence, then with $\gamma = \min_{i \neq j} \|T_i - T_j\|_1$, it holds that

$$R_{01}(\text{dec}(\bar{f};L_1)) \leq \frac{4\sqrt{2}}{\gamma} \sqrt{R(\bar{f};\ell)}$$  \hspace{1cm} (9)

The proof is in Appendix A.2. In the realizable case, where there is a target function $g$ that satisfies $g(x) = y$ for all instances, the term $R(\bar{f};\ell_{KL})$ can be minimized to zero with $\bar{f}^* : x \mapsto T_{g(x)}$. This indicates that for a sufficiently rich complementary hypothesis set, if the complementary probability estimator is consistent ($\bar{f} \to \bar{f}^*$) then the $L_1$ decoded prediction is consistent ($R_{01}(\text{dec}(\bar{f};L_1)) \to 0$). The result suggests that the performance of the $L_1$ decoder can be bounded by the accuracy of the probability estimates of complementary labels measured by the KL divergence. In other words, to obtain an accurate ordinary classifier, it suffices to find an accurate complementary probability estimator followed by the $L_1$ decoding. Admittedly, in the non-realizable case, $R(\bar{f};\ell_{KL})$ contains irreducible error. We leave the analysis of the error bound in this case for the future research.

Another implication of the Proposition 2 is related to the inaccurate transition matrix. Suppose the complementary labels are generated with respect to the transition matrix $T'$, which may be different from $T$, the one provided to the learning algorithm. In the proposed framework, the only affected component is the decoding step. This allows us to quantify the effect of inaccuracy as follows:

Corollary 1. For any $\bar{f} \in \mathcal{H}$, if $d$ is the $L_1$ distance and $\ell$ is the KL divergence, then

$$R_{01}(\text{dec}(\bar{f};L_1)) \leq \frac{4\sqrt{2}}{\gamma} \sqrt{R(\bar{f};\ell)} + \frac{2\epsilon}{\gamma}$$  \hspace{1cm} (10)

where $\gamma = \min_{i \neq j} \|T_i - T_j\|_1$ is the minimal $L_1$ distance between pairs of transition vectors, and $\epsilon = \max_{k \in [K]} \|T'_k - T_k\|_1$ denotes the difference between $T'$ and $T$.

Validation Phase: Quality of Probability Estimates The third implication of Proposition 2 is an alternative validation procedure to the unbiased risk estimation (URE) [3]. According to Proposition 2, selecting the best-performing parameter minimizes the right hand side of Eq. (9) among all hyper-parameter choices minimizes the ordinary classification error. This suggests an alternative metric for parameter selection: using the surrogate complementary estimation loss (SCEL) on the validation dataset.

Although the proposed validation procedure does not directly estimate the ordinary classification error, it provides benefits in the scenarios where URE does
A softmax function was applied to each row of Table 2. We propose to allow the transition layer to be trainable. This technique is also used in label-noise learning, such as [6]. Specifically, we propose three methods in our Complementary Probability Estimates framework: (a) CPE-I denotes a model without a transition layer (b) CPE-F denotes a model with a fixed additional layer to T (c) CPE-T denotes a model with a trainable transition layer. To make the transition layer trainable, we considered a $K \times K$ matrix $W$. A softmax function was applied to each row of $W$ to transform it into a valid transition matrix.

### 3.3 Connection to Previous Methods

The proposed framework also explains several earlier approaches as its special cases, including (1) Forward Correction (FWD) [11], (2) Surrogate Complementary Loss (SCL) with log loss [1], and (3) Discriminative Model (DM) [2], which are explained in Table 2 and Appendix B. By viewing those earlier approaches in the proposed framework, we provide additional benefits for them. First, the novel validation process can be applied for parameter selection. This provides an alternative to validate those approaches. Also, we fill the gap on the theoretical explanation to help understand those approaches in the realizable case.

On the other hand, the success of FWD inspires us to reconsider the role of transition layers in the framework. As the base model’s output $f(x; \theta)$ is in the probability simplex $\Delta^K$, the model’s output $T^\top f(x; \theta)$ lies in the convex hull formed by the row vectors of $T$. If the transition matrix $T$ provided to the learning algorithm is accurate, then such transformation helps control the model’s complexity by restricting its output. The restriction may be wrong, however, when the given transition matrix $T$ is inaccurate. To address this issue, we propose to allow the transition layer to be trainable.

Table 2. A unifying view of earlier approaches and proposed algorithms through the lens of reduction to probability estimates, where $U$ denote the uniform transition matrix.

| Method          | Hypothesis set | Decoder                |
|-----------------|----------------|------------------------|
| Fwd (general T) | $\{x \mapsto T^\top f(x; \theta) : \theta \in \Theta\}$ | $\text{argmax}_\theta (T^{-1})^\top f(x)_K$ |
| Fwd (uniform)   | $\{x \mapsto U^\top f(x; \theta) : \theta \in \Theta\}$ | $\text{argmin}_\theta \|f(x) - U \|_1$ |
| SCL             | $\{x \mapsto U^\top f(x; \theta) : \theta \in \Theta\}$ | $\text{argmin}_\theta \|f(x) - U \|_1$ |
| DM              | $\{x \mapsto \text{sm}(1 - f(x; \theta)) : \theta \in \Theta\}$ | $\text{argmin}_\theta \|f(x) - U \|_1$ |
| CPE-I (no transition) | $\{x \mapsto f(x; \theta) : \theta \in \Theta\}$ | $\text{argmin}_\theta \|f(x) - T \|_1$ |
| CPE-F (fixed transition) | $\{x \mapsto T^\top f(x; \theta) : \theta \in \Theta\}$ | $\text{argmin}_\theta \|f(x) - T \|_1$ |
| CPE-T (trainable transition) | $\{x \mapsto T(W)^\top f(x; \theta) : \theta \in \Theta, W \in \mathbb{R}^{K \times K}\}$ | $\text{argmin}_\theta \|f(x) - T \|_1$ |
Reduction from Complementary-Label Learning to Probability Estimates

transition matrix \( T(W) = (\text{sm}(W_1), \text{sm}(W_2), \ldots, \text{sm}(W_K))^\top \). For a base model \( f \), the complementary probability estimates of CPE-T for a given instance \( x \) would be \( T(W)^\top f(x; \theta) \). Note that we use the \( L_1 \) decoder for CPE-I, CPE-F, and CPE-T.

4 Experiments

In this section, we benchmark the proposed framework to the state-of-the-art baselines and discuss the following questions: (a) Can the transition layers improve the model’s performance? (b) Is the proposed \( L_1 \) decoding competitive to MAX? (c) Does the transition matrix provide information to the learning algorithms even if it is inaccurate? We further demonstrate the flexibility of incorporating traditional models in CPE in Section 4.3 and verify the effectiveness of the proposed validation procedure in the Appendix.

4.1 Experiment Setup

Baseline and setup We first evaluate CPE with the following state-of-the-art methods: (a) URE-GA: Gradient Ascent applied on the unbiased risk estimator \([3,4]\), (b) Fwd: Forward Correction \([11]\), (c) SCL: Surrogate Complementary Loss with negative log loss \([1]\), and (d) DM: Discriminative Models with Weighted Loss \([2]\). Following the previous work, we test those methods on MNIST, Fashion-MNIST, and Kuzushiji-MNIST, and use one-layer mlp model (d-500-c) as base models. All models are optimized using Adam with learning rate selected from \{1e-3, 5e-4, 1e-4, 5e-5, 1e-5\} and a fixed weight decay 1e-4 for 300 epochs. The learning rate for CPE is selected with the Surrogate Complementary Estimation Loss (SCEL) on the validation dataset. For the baseline method, it is selected with unbiased risk estimator (URE) of the zero-one loss. It is worth noting that the validation datasets consist of only complementary labels, which is different from some previous works.

Transition matrices In the experiment of clean transition matrices, three types of transition matrices are benchmarked in the experiment. Besides the uniform transition matrix, following \([2,11]\), we generated two biased ones as follows: For each class \( y \), the complementary classes \( \mathcal{Y} \setminus \{y\} \) are first randomly split into three subsets. Within each subset, the probabilities are set to \( p_1, p_2 \) and \( p_3 \), respectively. We consider two cases for \((p_1, p_2, p_3)\): (a) Strong: \((0.75, 0.24, 0.01)\) to model stronger deviation from uniform transition matrices, (b) Weak: \((0.45, 0.30, 0.25)\) to model milder deviation from uniform transition matrices. In the experiment of noisy transition matrices, we consider the Strong deviation transition matrix \( T_{\text{strong}} \) to be the ground-truth transition matrix, and a uniform noise transition matrix \( \frac{1}{K} \mathbf{1}_K \) to model the noisy complementary label generation. We generated complementary labels with the transition matrix \((1 - \lambda)T_{\text{strong}} + \lambda \frac{1}{K} \mathbf{1}_K\), but provided \( T_{\text{strong}} \) and the generated complementary dataset to the learners. The parameter \( \lambda \) controls the proportion of the uniform noise in the complementary labels. The results are reported in Table 3.
The difference makes suggests that $L_CPE-F$ only differ in the decoding step, with the former using is restricted to a wrong output space, making it underperform $CPE-F$ of complementary labels better by altering the transition layer. In contrast, making transition layers trainable, the model can potentially fit the distribution setting, especially when the noise factor clean or not. Also, we observe that help achieve higher performances, no matter the provided transition matrix is in both clean and noisy experiments. We observe that the answer is positive.

| MNIST     | Fashion-MNIST | Kuzushiji-MNIST |
|-----------|---------------|-----------------|
| Unif. | Weak | Strong | Unif. | Weak | Strong | Unif. | Weak | Strong |
| URE-GA | 90.3±2.2 | 87.8±2.9 | 70.3±3.1 | 75.7±2.0 | 53.3±4.5 | 65.6±2.8 | 62.5±1.1 | 23.3±5.4 |
| SCL | 94.3±0.4 | 95.8±0.4 | 27.5±1.9 | 82.6±0.4 | 61.2±0.1 | 28.5±10.8 | 73.7±1.4 | 71.2±2.9 | 20.7±4.8 |
| DM | 91.9±0.6 | 90.2±0.3 | 26.7±4.6 | 82.5±0.3 | 80.3±1.1 | 28.8±5.9 | 63.6±2.0 | 64.5±2.7 | 20.1±3.2 |
| Fwd | 94.4±0.2 | 91.9±0.3 | 95.3±0.4 | 82.6±0.6 | 83.0±1.0 | 85.2±0.3 | 75.2±1.6 | 63.1±2.6 | 74.1±4.8 |
| CPE-I | 90.2±0.2 | 88.4±0.3 | 92.7±0.8 | 81.1±0.3 | 79.2±0.5 | 81.9±1.4 | 66.2±1.0 | 62.5±0.9 | 73.7±1.0 |
| CPE-F | 94.4±0.2 | 92.0±0.2 | 95.5±0.3 | 83.0±0.1 | 83.0±0.3 | 85.8±0.3 | 73.5±1.6 | 64.6±0.5 | 75.3±2.6 |
| CPE-T | 92.8±0.6 | 92.1±0.2 | 95.2±0.5 | 83.0±0.1 | 80.3±0.3 | 85.8±0.3 | 63.6±0.4 | 64.6±0.4 | 74.2±2.8 |

Comparison of testing classification accuracies with different transition matrices (upper part) and different noise levels (lower part).

Table 3.

| Unif. | Weak | Strong | Unif. | Weak | Strong | Unif. | Weak | Strong |
|-------|------|--------|------|------|--------|------|------|--------|
| A ≈ 0.1 | A = 0.2 | A = 0.5 | A = 0.1 | A = 0.2 | A = 0.5 | A = 0.1 | A = 0.2 | A = 0.5 |
| URE-GA | 31.8±6.4 | 27.8±8.2 | 28.1±4.1 | 27.3±5.5 | 28.6±4.1 | 28.3±2.0 | 24.5±4.6 | 21.1±2.2 | 19.8±2.1 |
| SCL | 25.1±11.7 | 24.7±8.9 | 23.8±2.7 | 26.6±2.7 | 23.2±6.7 | 23.2±2.7 | 20.4±2.4 | 17.3±2.0 | 16.8±2.0 |
| DM | 26.5±9.1 | 24.6±6.5 | 22.6±1.3 | 24.1±5.1 | 23.6±6.7 | 22.6±2.9 | 20.0±2.6 | 18.2±3.1 | 18.2±1.6 |
| Fwd | 88.3±8.7 | 83.9±10.7 | 71.6±18.4 | 84.8±0.6 | 80.0±2.6 | 62.9±20.1 | 72.8±5.6 | 67.6±7.5 | 54.7±12.4 |
| CPE-I | 92.4±0.7 | 92.0±0.8 | 87.6±1.4 | 81.7±1.4 | 81.3±1.4 | 78.6±1.5 | 73.0±0.7 | 71.6±0.9 | 62.7±1.6 |
| CPE-F | 94.3±0.5 | 93.6±0.5 | 89.9±1.4 | 84.1±0.8 | 83.0±1.1 | 78.4±2.5 | 76.1±1.8 | 73.7±1.8 | 63.7±1.5 |
| CPE-T | 94.4±0.5 | 93.7±0.5 | 89.6±0.9 | 84.1±0.8 | 83.2±1.1 | 78.9±2.0 | 76.1±1.3 | 73.9±1.6 | 64.2±1.2 |

Table 4.

Comparison of testing accuracies of decoders when the baseline models use fixed transition layers. The parameters are selected from the one with smallest SCEL on the validation dataset.

4.2 Discussion

Can Transition Layers Improve Performance? The answer is positive in both clean and noisy experiments. We observe that CPE-F and CPE-T outperform CPE-I in both settings, demonstrating that the transition layer help achieve higher performances, no matter the provided transition matrix is clean or not. Also, we observe that CPE-T outperforms CPE-F in the noisy setting, especially when the noise factor $\lambda$ is large. It demonstrates that by making transition layers trainable, the model can potentially fit the distribution of complementary labels better by altering the transition layer. In contrast, CPE is restricted to a wrong output space, making it underperform CPE-T. The difference makes CPE-T a better choice for noisy environment.

Is $L_1$ competitive with MAX? As analyzed in Section 3.3, Fwd and CPE-F only differ in the decoding step, with the former using MAX and the latter using $L_1$. We provide the testing accuracies of these decoders when the base models are CPE-F in Table 3. It is displayed that the MAX decoder outperform $L_1$ in most noiseless settings; however, when the transition matrix is highly inaccurate ($\lambda = 0.5$), we observe that the $L_1$ decoder outperform the MAX decoder. This suggests that $L_1$ could be more tolerant to an inaccurate transition matrix.
Table 5. Comparison of testing accuracies of CPE with traditional models. **Boldfaced** ones outperform the baseline methods based on single-layer deep models.

| Model       | MNIST Unif. | Weak | Strong | MNIST Unif. | Weak | Strong | Kuzushiji-MNIST Unif. | Weak | Strong |
|-------------|-------------|------|--------|-------------|------|--------|--------------------|------|--------|
| CPE-KNN     | 93.1 ± 0.1  | 92.6 ± 0.1 | 94.5 ± 0.4  | 79.1 ± 0.4  | 77.8 ± 0.6  | 79.0 ± 1.7  | 74.0 ± 0.8  | 73.7 ± 0.8  | 80.4 ± 1.3 |
| CPE-GBDT    | 86.9 ± 0.3  | 86.0 ± 0.2  | 90.3 ± 0.9  | 79.2 ± 0.3  | 76.8 ± 0.4  | 81.4 ± 1.1  | 60.6 ± 0.4  | 76.6 ± 1.8  | 68.4 ± 2.1 |

These results reveal that a deeper sensitivity analysis of different decoders, both empirically and theoretically, would be desired. We leave this as future studies.

**Discussion of T-agnostic models** Among the baseline methods, URE-GA, SCL and DM are ones that does not take $T$ as inputs or assumes $T$ is uniform, which we called $T$-agnostic models. Those models perform well when the transition matrix is just slightly deviated from the uniform one, but their performances all dropped when the deviation from uniform becomes larger. As we discussed in Section 3.3 the result can be interpreted to be caused by their implicit assumption on uniform transition matrices, which brings great performance on uniform transition matrices but worse performance on biased ones. In contrast, we observed that all variations of CPE have similar testing accuracies across different transition matrices, demonstrating that CPE does exploit the information from the transition matrix that helps the models deliver better performance.

### 4.3 Learn from CL with Traditional Methods

As discussed in Section 3, the proposed framework is not constrained by deep models. We explored the possibility of applying traditional methods to learn from CL, including (a) $k$-Nearest Neighbor ($k$-NN) and (b) Gradient Boosting Decision Tree (GBDT). We benchmarked those models in the same settings and reported the results in Table 5. It displays that traditional models, specifically, $k$-NN, outperform all the methods using deep models in Kuzushiji-MNIST, indicating the benefit of the proposed CPE’s flexibility in using non-deep models.

### 5 Conclusion

In this paper, we view the CLL problem from a novel perspective, reduction to complementary probability estimates. Through this perspective, we propose a framework that only requires complementary probability estimates and prove that a simple decoding step can map the estimates to ordinary labels. The framework comes with a theoretically justified validation procedure, provable tolerance in noisy environment, and flexibility of incorporating non-deep models. Empirical experiments further verify the effectiveness and robustness of the proposed framework under broader scenarios, including non-uniform and noisy complementary label generation. We expect the realistic elements of the framework to keep inspiring future research towards making CLL practical.
References

1. Chou, Y.T., Niu, G., Lin, H.T., Sugiyama, M.: Unbiased risk estimators can mislead: A case study of learning with complementary labels. In: International Conference on Machine Learning. pp. 1929–1938. PMLR (2020)

2. Gao, Y., Zhang, M.L.: Discriminative complementary-label learning with weighted loss. In: International Conference on Machine Learning. pp. 3587–3597. PMLR (2021)

3. Ishida, T., Niu, G., Hu, W., Sugiyama, M.: Learning from complementary labels. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. pp. 5644–5654 (2017)

4. Ishida, T., Niu, G., Menon, A., Sugiyama, M.: Complementary-label learning for arbitrary losses and models. In: International Conference on Machine Learning. pp. 2971–2980. PMLR (2019)

5. Kull, M., Flach, P.: Novel decompositions of proper scoring rules for classification: Score adjustment as precursor to calibration. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 68–85. Springer (2015)

6. Li, X., Liu, T., Han, B., Niu, G., Sugiyama, M.: Provably end-to-end label-noise learning without anchor points. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 6403–6413. PMLR (18–24 Jul 2021)

7. Liu, J., Hang, H., Wang, B., Li, B., Wang, H., Tian, Y., Shi, Y.: Gan-cl: Generative adversarial networks for learning from complementary labels. IEEE Transactions on Cybernetics (2021)

8. Wang, D.B., Feng, L., Zhang, M.L.: Learning from complementary labels via partial-output consistency regularization. In: IJCAI. pp. 3075–3081 (2021)

9. Williamson, R.C., Vernet, E., Reid, M.D.: Composite multiclass losses. Journal of Machine Learning Research 17(222), 1–52 (2016)

10. Xu, Y., Gong, M., Chen, J., Liu, T., Zhang, K., Batmanghelich, K.: Generative-discriminative complementary learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 6526–6533 (2020)

11. Yu, X., Liu, T., Gong, M., Tao, D.: Learning with biased complementary labels. In: Proceedings of the European conference on computer vision (ECCV). pp. 68–83 (2018)

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

13. Zhang, Y., Liu, F., Fang, Z., Yuan, B., Zhang, G., Lu, J.: Learning from a complementary-label source domain: Theory and algorithms. IEEE Transactions on Neural Networks and Learning Systems (2021)

14. Zhou, Z.H.: A brief introduction to weakly supervised learning. National science review 5(1), 44–53 (2018)

Acknowledgements. We thank the anonymous reviewers and the members of NTU CLLab for valuable suggestions. The work is partially supported by the National Science and Technology Council via the grants 110-2628-E-002-013 and 111-2628-E-002-018. We also thank the National Center for High-performance Computing (NCHC) of National Applied Research Laboratories (NARLabs) in Taiwan for providing computational resources.
A Proofs

This section provides the proofs for the propositions, theorems claimed in the main text.

A.1 Proof of Proposition 1

First, set \( C = \mathbb{E}_{(x,y) \sim D} \sum_{k=1}^{K} T_{yk} \log(T_{yk}) \), then

\[
\mathbb{E}_{(x,y) \sim D} \ell(\bar{f}(x), T_y) = \mathbb{E}_{(x,y) \sim D} \sum_{k=1}^{K} -T_{yk} \log \left( \frac{\bar{f}_k(x)}{T_{yk}} \right) = C + \mathbb{E}_{(x,y) \sim D} \sum_{k=1}^{K} -T_{yk} \log(\bar{f}_k(x)) \tag{11}
\]

Next, as \( P(\bar{y} | y) = \frac{T_y \bar{y}}{T_y \bar{y}} \), then

\[
\mathbb{E}_{(x,y) \sim D} \sum_{k=1}^{K} -T_{yk} \log(\bar{f}_k(x)) = \mathbb{E}_{(x,y) \sim \bar{D}} \left( \mathbb{E}_{\bar{y} | y} - \log(\bar{f}_{\bar{y}}(x)) \right) = \mathbb{E}_{(x,y) \sim \bar{D}} \ell(\bar{f}(x), e_{\bar{y}}) \tag{12}
\]

Hence, \( \mathbb{E}_{(x,y) \sim D} \ell(\bar{f}(x), T_y) = C + \mathbb{E}_{(x,y) \sim \bar{D}} \ell(\bar{f}(x), e_{\bar{y}}) \).

A.2 Proof of Proposition 2

Let \( I_A \) denote the indicator function of event \( A \), then using Markov’s inequality on the random variable \( d(\bar{f}(x), T_y) \), we have

\[
R_{01}(\text{dec}(\bar{f}; d)) \leq P(\text{dec}(\bar{f}(x), T_y) \geq \frac{\gamma_d}{2}) \leq \frac{2}{\gamma_d} \mathbb{E} \left[ d(\bar{f}(x), T_y) \right] = \frac{2}{\gamma_d} R(\bar{f}; d) \tag{13}
\]

To see the first inequality holds, note that if \( d(\bar{f}(x), T_y) < \frac{\gamma_d}{2} \), then for any incorrect class \( y' \neq y \), we have

\[
d(\bar{f}(x), T_{y'}) \geq d(T_y, T_{y'}) - d(T_y, \bar{f}(x)) \geq \frac{\gamma_d}{2} \tag{14}
\]

by triangular inequality and the definition of \( \gamma_d \). As a result, the decoder decodes \( \bar{f}(x) \) to the correct class \( y \) if \( d(\bar{f}(x), T_y) < \frac{\gamma_d}{2} \). This completes the first part of the Proposition.

Next, by Pinsker’s inequality and Jensen’s inequality, we have that

\[
R(\bar{f}; L_1) = \mathbb{E}_{(x,y) \sim D} \left\| \bar{f}(x) - T_y \right\|_1 \leq 2 \mathbb{E}_{(x,y) \sim D} \sqrt{2\ell_{\text{KL}}(\bar{f}(x), T_y)} \leq 2 \sqrt{2 \mathbb{E}_{(x,y) \sim D} \ell_{\text{KL}}(\bar{f}(x), T_y)} = 2\sqrt{2 R(\bar{f}; \ell_{\text{KL}})} \tag{17}
\]

According to the above inequality and the results of the first part, the proof for the second part is now complete.
A.3 Proof of Corollary 1

The decoding step remains the same when \( T' \neq T \) because the decoder uses the same transition matrix \( T \) to decode. The only difference is in the complementary probability estimates. Specifically, we have that the complementary estimation loss becomes \( R(f; \ell) = \mathbb{E}_{(x,y) \sim D} (\ell(f(x), T'_y)) \) as the complementary labels are generated with respect to \( T' \).

Hence, the last equality in Equation (13) is no longer correct. Instead, we use the following:

\[
\mathbb{E} \left[ d(\bar{f}(x), T_y) \right] \leq \mathbb{E} \left[ d(\bar{f}(x), T'_y) + d(T'_y, T_y) \right] \leq \mathbb{E} \left[ d(\bar{f}(x), T'_y) \right] + \epsilon
\]  

(18)

to obtain that \( R_{01}(\text{dec}(\bar{f}; L_1)) \leq \frac{4 \sqrt{2}}{\gamma} \sqrt{R(f; \ell)} + \frac{2\epsilon}{\gamma} \) . Then, we can use Pinsker’s inequality and Jensen’s inequality as in (15) to get

\[
R_{01}(\text{dec}(f; L_1)) \leq \frac{4 \sqrt{2}}{\gamma} \sqrt{R(f; \ell)} + \frac{2\epsilon}{\gamma} .
\]  

(19)

B Details of the Connections between Proposed Framework and Previous Methods

In this section, we provide further details about how our framework can explain several previous methods as its special cases. Across this section, we let \( f(x; \theta) \) denote the base model parametrized by \( \theta \in \Theta \). We also provide some insights drawn from viewing these previous methods using the proposed framework.

**Forward Correction** In the training phase, Forward Correction optimizes the following loss functions:

\[
L_{\text{Fwd}}(\theta) = \frac{1}{N} \sum_{i=1}^{N} -\log (T^\top f(x_i; \theta))_{\hat{y}_i}
\]  

(20)

In the inference phase, Forward Correction predicts \( \hat{y} = \arg\max_k f_k(x) \) for an unseen instance \( x \). We claim that Forward Correction is equivalent to CPE with the following parameters when \( T \) is invertible:

- Hypothesis Set: \( \{ x \mapsto T^\top f(x; \theta) : \theta \in \Theta \} \)
- Decoder: \( \arg\max_k ((T^\top)^{-1} f(x; \theta))_k \).

**Proof.** First, by setting the hypothesis set as above and plugging in the surrogate complementary estimation loss, we get the training objective function for CPE:

\[
L_{\text{CPE}}(\theta) = \frac{1}{N} \sum_{i=1}^{N} -\log (T^\top f(x_i; \theta))_{\bar{y}_i}
\]

(21)

Equation (21) matches Equation (20), implying that in the training phase they select the same parameter \( \theta \). Next, in the inference phase, it is clear that \( (T^\top)^{-1} f(x; \theta) = (T^\top)^{-1} T^\top f(x; \theta) = f(x; \theta) \), so both methods predict the same label for an instance \( x \).
Next, we further show that when \( T \) is the uniform transition matrix \( U \), the decoder is equivalent to the \( L_1 \) decoder, i.e., \( \arg\max_k ((U^\top)^{-1}f(x))_k = \arg\min_k \| U_k - f(x) \|_1 \).

**Proof.** First, as

\[
((U^\top)^{-1}f(x))_k = -(K - 1)f_k(x) + \sum_{k=1}^K f_k(x) = -(K - 1)f_k(x) + 1,
\]

we have that \( \arg\max_k ((U^\top)^{-1}f(x))_k = \arg\min_k \bar{f}_k(x) \). Next, set \( \bar{y} = \arg\min_k \bar{f}_k(x) \).

For any \( y \neq \bar{y} \), we want to show

\[
|U_{\bar{y}\bar{y}} - \bar{f}_{\bar{y}}(x)| + |U_{yy} - \bar{f}_y(x)| \geq |U_{\bar{y}y} - \bar{f}_{\bar{y}}(x)| + |U_{y\bar{y}} - \bar{f}_{\bar{y}}(x)|.
\]

As \( \bar{f}_{\bar{y}}(x) \leq \frac{1}{K} \leq \frac{1}{K-1} = U_{\bar{y}y} \),

\[
|U_{\bar{y}\bar{y}} - \bar{f}_{\bar{y}}(x)| + |U_{yy} - \bar{f}_y(x)| = |U_{\bar{y}y} - \bar{f}_{\bar{y}}(x)| + \bar{f}_y(x) + |U_{y\bar{y}} - \bar{f}_{\bar{y}}(x)| - \bar{f}_y(x) \tag{23}
\]

\[
= |U_{\bar{y}y} - \bar{f}_{\bar{y}}(x)| + \frac{1}{K - 1} - \bar{f}_y(x) + \bar{f}_y(x) - \bar{f}_y(x) \tag{24}
\]

If \( \bar{f}_y(x) \leq \frac{1}{K - 1} \), as \( \bar{f}_{\bar{y}}(x) \leq \bar{f}_y(x) \),

\[
\frac{1}{K - 1} - \bar{f}_y(x) + \bar{f}_y(x) - \bar{f}_y(x) \geq \frac{1}{K - 1} - \bar{f}_y(x) \geq \frac{1}{K - 1} - \bar{f}_y(x) = |U_{\bar{y}y} - \bar{f}_{\bar{y}}(x)|
\]

Otherwise, as \( \bar{f}_{\bar{y}}(x) \leq \frac{1}{K} \),

\[
\frac{1}{K - 1} - \bar{f}_y(x) + \bar{f}_y(x) - \bar{f}_y(x) \geq \bar{f}_y(x) - \bar{f}_y(x) \geq \frac{1}{K - 1} - \bar{f}_y(x) = |U_{\bar{y}y} - \bar{f}_{\bar{y}}(x)|
\]

Hence, Equation (22) holds. Now,

\[
\sum_{k=1}^K |U_{yk} - \bar{f}_k(x)| = |U_{\bar{y}\bar{y}} - \bar{f}_{\bar{y}}(x)| + |U_{yy} - \bar{f}_y(x)| + \sum_{k \neq \bar{y}, \bar{y}}^K |U_{yk} - \bar{f}_k(x)| \tag{26}
\]

\[
\geq |U_{\bar{y}y} - \bar{f}_{\bar{y}}(x)| + |U_{y\bar{y}} - \bar{f}_{\bar{y}}(x)| + \sum_{k \neq \bar{y}, \bar{y}}^K |U_{yk} - \bar{f}_k(x)| = \sum_{k=1}^K |U_{yk} - \bar{f}_k(x)| \tag{27}
\]

As a result, \( \bar{y} \) minimizes \( k \mapsto \|U_k - f(x)\|_1 \). Hence, we conclude that \( \arg\min_k \bar{f}_k(x) = \bar{y} = \arg\min_k \|U_k - f(x)\|_1 \). Then the proof is complete.

As the two decoders are equivalent, we have that Forward Correction is equivalent to CPE with

- Hypothesis Set: \( \{x \mapsto U^\top f(x; \theta) : \theta \in \Theta\} \)
- Decoder: \( \arg\min_k \|f(x; \theta) - U_k\|_1 \).

when the transition layer is fixed to the uniform transition matrix.
**Surrogate Complementary Loss** In the training phase, Surrogate Complementary Loss with Log Loss optimizes the following loss functions:

\[
L_{\text{SCL}}(\theta) = \frac{1}{N} \sum_{i=1}^{N} - \log(1 - f(x_i; \theta)) \hat{y}_i, \tag{28}
\]

In the inference phase, this method predicts the ordinary labels by \( \hat{y} = \arg\max_k f_k(x) \) for an unseen instance \( x \). We claim that this method is equivalent CPE with:

- Hypothesis Set: \( \{ x \mapsto U^\top f(x; \theta) : \theta \in \Theta \} \)
- Decoder: \( \arg\min_k \| \tilde{f}(x; \theta) - U_k \|_1 \).

**Proof.** Observe that the training objective function for CPE with the hypothesis set has the following property:

\[
L_{\text{CPE}}(\theta) = \frac{1}{N} \sum_{i=1}^{N} - \log (U^\top f(x_i; \theta) \hat{y}_i) = \frac{1}{N} \sum_{i=1}^{N} - \log \left( \frac{1}{K-1} \sum_{k \neq \hat{y}_i} f_k(x_i; \theta) \right) \tag{29}
\]

\[
= \frac{1}{N} \sum_{i=1}^{N} - \log \left( 1 - f_{\hat{y}_i}(x_i; \theta) \right) + \log(K - 1) = L_{\text{SCL}}(\theta) + \log(K - 1) \tag{30}
\]

That is, the objective function only differs by a constant. As a result, the two methods match during the training phase.

In inference phase, SCL predicts \( \hat{y} = \arg\max_k f_k(x) \) for unseen instance \( x \) as in Forward Correction. In addition, they have the same hypothesis set \( \{ x \mapsto U^\top f(x; \theta) : \theta \in \Theta \} \) if the transition layer of Forward Correction is fixed to uniform. Hence, SCL is equivalent to Forward Correction with uniform transition layer. It implies that they have the same decoder: \( \hat{y} = \arg\min_k \| \tilde{f}(x) - U_k \|_1 \).

**Discriminative Model** In the training phase, Discriminative Model with unweighted loss optimizes the following loss functions:

\[
L_{\text{DM}}(\theta) = \frac{1}{N} \sum_{i=1}^{N} - \log \left( \text{sm}(1 - f(x_i; \theta)) \right) \hat{y}_i, \tag{31}
\]

In the inference phase, this method predicts the ordinary labels by \( \hat{y} = \arg\max_k f_k(x) \) for an unseen instance \( x \). We claim that this method is equivalent CPE with:

- Hypothesis Set: \( \{ x \mapsto \text{sm}(1 - f(x; \theta)) : \theta \in \Theta \} \)
- Decoder: \( \arg\min_k \| \tilde{f}(x; \theta) - U_k \|_1 \).

**Proof.** The equivalence in the training phase is clear by plugging in the hypothesis to the surrogate complementary estimation loss. During inference phase, first observe that

\[
f_k(x) = \frac{1}{Z} \exp \left( 1 - f_k(x; \theta) \right) = \frac{e}{Z} \exp \left( - f_k(x_i; \theta) \right), \tag{32}
\]
where $Z = \sum_{k=1}^{K} \exp \left( 1 - f_k(x; \theta) \right)$ is the normalization term. As $x \mapsto \exp(-x)$ is monotonic decreasing, we have that $\arg\min_k \tilde{f}_k(x; \theta) = \arg\max_k f_k(x; \theta)$. Next, as we have shown $\arg\min_k \tilde{f}_k(x) = \arg\min_k \|U_k - \tilde{f}_k(x)\|_1$, so $\arg\max_k f_k(x; \theta) = \arg\min_k \|U_k - \tilde{f}_k(x)\|_1$, implying that both methods predict the same label for all instances.

Observations by viewing earlier approaches with the proposed framework: We also draw the following observations by viewing earlier approaches with the proposed CPE framework:

1. By viewing Fwd with the proposed framework, the equivalent decoder essentially converts the complementary probability estimates back to the ordinary probability estimates and predicts the largest one. We name it MAX decoding for future reference.

2. If the transition matrix is uniform, then Fwd and SCL with log loss match, suggesting that they are the same in this situation. It explains why those two methods have similar performances in [1], which is also reproduced in our experiment, reported in Table 3.

3. DM was proposed to lift the generation assumption of complementary labels [2], but from the view of the CPE framework, DM implicitly assumes the complementary labels are generated uniformly, as we can see from the decoder. This provides an alternative explanation why its performance deteriorates as the transition matrix deviates from the uniform matrix, as shown in [2].
C Experiment Details

In this section, we provide missing details of the experiments in Section 4.

C.1 Setup

Datasets Across the experiments, we use the following datasets:
- MNIST
- Fashion-MNIST
- Kuzushiji-MNIST

For the above dataset, the size of the training set is 60000, and the size of the testing set is 10000. To perform the hyperparameter selection, in each trial, we split 10 percent of the training dataset randomly as the validation dataset. We performed five trials with different random seeds for all the experiments in this paper. To ensure a fair comparison, the dataset split and the generated complementary labels are the same for the benchmark algorithms. Also, we did not include data augmentation or consistency regularization [8] in the experiment to prevent introducing extra factors and simplify the comparison.

Models We implemented the deep models in PyTorch. The base models considered in the experiment are linear and one-layer mlp model (d-500-c) with 500 hidden units. In CPE-T, the parameter of the transition layer is initialized such that it matches the provided transition matrix, i.e. it is initialized to $W_0$ such that $T(W_0) = T$. All models are optimized using Adam with learning rate selected from $\{1e^{-3}, 5e^{-4}, 1e^{-4}, 5e^{-5}, 1e^{-5}\}$ and a fixed weight decay $1e^{-4}$ for 300 epochs. We used the default parameters in PyTorch for other parameters in Adam. The experiments are run with Nvidia Tesla V100 GPUs.

For the two traditional models, we used the K nearest neighbor (KNN) classifier from scikit-learn with the number of neighbors selected from $\{10, 20, \ldots, 250\}$ based on the complementary estimation loss on the validation dataset. We performed PCA on the dataset to map the feature to a 32-dimension space for KNN to reduce the training/inference time. We used Gradient Boosting Decision Tree from LightGBM, and set the objective to “multiclass” to optimize the log loss. The hyperparameters include the number of trees $\{5, 10, \ldots, 500\}$ and learning rate $\{0.01, 0.025, 0.05, 0.1\}$. Those parameters are also selected based on the complementary estimation loss on the validation dataset.

C.2 Additional Results

This section provides figures and tables that are helpful in analyzing the experiment results.
Table 6. Comparison of the testing classification accuracies with different transition matrices.

|                | MNIST | Fashion-MNIST | Kuzushiji-MNIST |
|----------------|-------|---------------|-----------------|
|                | Unif. | Weak | Strong | Unif. | Weak | Strong | Unif. | Weak | Strong |
| URE-GA         | 81.7±0.5 | 73.4±1.4 | 23.2±2.9 | 76.2±0.3 | 70.8±1.5 | 21.3±5.5 | 51.0±1.0 | 43.7±1.0 | 16.7±2.5 |
| SCL            | 90.5±0.2 | 90.2±0.1 | 90.2±0.2 | 82.0±0.4 | 79.6±2.2 | 26.2±8.7 | 59.9±1.9 | 58.9±2.7 | 16.4±2.2 |
| DM             | 89.7±0.5 | 89.1±0.3 | 22.7±8.5 | 81.8±0.3 | 78.2±3.1 | 23.6±6.5 | 61.3±1.5 | 59.4±1.4 | 17.7±3.0 |
| Fwd            | 90.5±0.2 | 90.6±0.4 | 91.6±0.7 | 82.0±0.4 | 81.6±1.2 | 83.4±0.7 | 59.9±1.9 | 60.4±2.0 | 62.6±0.7 |
| CPE-I          | 80.4±0.3 | 73.5±1.3 | 76.1±1.0 | 74.6±0.5 | 71.0±2.0 | 74.7±1.5 | 61.0±0.6 | 42.8±0.8 | 48.8±1.4 |
| CPE-F          | 90.5±0.2 | 90.7±0.1 | 91.8±0.4 | 82.2±0.3 | 82.4±0.4 | 83.1±0.4 | 60.4±0.6 | 60.8±0.4 | 62.8±0.2 |
| CPE-T          | 90.5±0.2 | 90.6±0.3 | 91.8±0.4 | 82.0±0.3 | 82.1±0.5 | 83.2±0.2 | 60.3±0.5 | 60.6±0.5 | 63.0±0.3 |

Table 7. Comparison of the testing classification accuracies with different levels of noise.

|                | MNIST | Fashion-MNIST | Kuzushiji-MNIST |
|----------------|-------|---------------|-----------------|
|                | Unif. | Weak | Strong | Unif. | Weak | Strong | Unif. | Weak | Strong |
| URE-GA         | 22.8±2.0 | 21.1±4.4 | 21.4±1.6 | 20.2±6.7 | 23.5±3.9 | 22.6±3.1 | 16.8±2.1 | 16.4±2.8 | 15.2±2.2 |
| SCL            | 25.6±13.8 | 23.9±10.3 | 23.7±4.3 | 23.9±7.8 | 24.5±5.2 | 26.0±3.2 | 17.8±2.3 | 17.8±3.0 | 17.4±1.8 |
| DM             | 23.3±7.4 | 22.4±8.7 | 23.4±2.9 | 24.1±7.1 | 24.3±5.0 | 25.6±3.0 | 18.1±2.6 | 17.6±2.4 | 16.5±1.4 |
| Fwd            | 91.4±0.7 | 89.6±1.0 | 82.5±3.6 | 82.4±0.9 | 81.4±0.9 | 72.0±1.0 | 62.7±1.0 | 60.9±0.9 | 52.1±6.2 |
| CPE-I          | 75.7±2.0 | 75.4±2.0 | 73.8±2.2 | 74.6±2.3 | 73.9±2.2 | 71.1±2.0 | 47.0±1.4 | 46.5±1.3 | 43.4±1.1 |
| CPE-F          | 91.2±0.7 | 90.2±1.0 | 85.2±1.7 | 82.2±1.2 | 81.0±1.5 | 75.4±3.3 | 61.9±0.9 | 61.1±2.2 | 62.5±4.1 |
| CPE-F          | 91.3±0.7 | 90.5±0.8 | 85.7±1.6 | 82.6±1.3 | 81.6±1.3 | 78.0±1.0 | 62.2±0.8 | 61.7±1.7 | 59.0±2.1 |

Benchmark results of linear models Table 6 and 7 provide the noiseless and noisy benchmark results using linear models as base models, using the same setting in Section 4.1. We can see that the proposed CPE performs slightly better or is competitive with the baseline methods in most scenarios. When the transition matrix is highly inaccurate ($\lambda = 0.5$), CPE outperforms the baselines and is more stable in terms of testing accuracies. These are consistent with our observation when using mlp as base models.
Table 8. Comparison of CPE-T’s testing accuracies using different validation procedures.

|                | MNIST       | Fashion-MNIST | Kuzushiji-MNIST |
|----------------|-------------|---------------|-----------------|
|                | Use         | Weak          | Strong          | Use         | Weak          | Strong          | Use         | Weak          | Strong          |
| **LINEAR**     |             |               |                 |             |               |                 |             |               |                 |
| URE            | 90.3 ± 0.6  | 90.4 ± 0.3    | 91.8 ± 0.5     | 82.1 ± 0.3  | 81.5 ± 1.2    | 82.6 ± 1.3     | 59.9 ± 0.4  | 60.0 ± 0.9    | 62.5 ± 0.5     |
| SCEL           | 90.5 ± 0.2  | 90.6 ± 0.1    | 91.8 ± 0.4     | 82.0 ± 0.3  | 82.1 ± 0.5    | 83.2 ± 1.2     | 60.3 ± 0.5  | 60.6 ± 0.5    | 63.0 ± 0.3     |
| **MLP**        |             |               |                 |             |               |                 |             |               |                 |
| URE            | 92.7 ± 0.5  | 91.8 ± 0.7    | 90.4 ± 0.5     | 82.9 ± 0.1  | 83.0 ± 0.3    | 84.3 ± 1.5     | 63.8 ± 0.7  | 63.8 ± 1.9    | 74.5 ± 2.7     |
| SCEL           | 92.8 ± 0.6  | 92.1 ± 0.2    | 95.2 ± 0.5     | 83.6 ± 0.1  | 83.0 ± 0.3    | 85.8 ± 0.3     | 63.6 ± 0.4  | 64.6 ± 0.4    | 74.2 ± 2.8     |

A ± 0.1 A ± 0.2 A ± 0.5 A ± 0.1 A ± 0.2 A ± 0.5 A ± 0.1 A ± 0.2 A ± 0.5

Table 9. Comparison of Fwd’s testing accuracies using different validation procedures.

|                | MNIST       | Fashion-MNIST | Kuzushiji-MNIST |
|----------------|-------------|---------------|-----------------|
|                | Use         | Weak          | Strong          | Use         | Weak          | Strong          | Use         | Weak          | Strong          |
| **LINEAR**     |             |               |                 |             |               |                 |             |               |                 |
| URE            | 90.9 ± 1.0  | 90.2 ± 0.8    | 86.1 ± 1.3     | 82.2 ± 1.3  | 81.2 ± 1.4    | 77.1 ± 1.8     | 62.3 ± 0.8  | 60.6 ± 0.9    | 55.3 ± 0.3     |
| SCEL           | 91.3 ± 0.7  | 90.5 ± 0.8    | 85.7 ± 1.6     | 82.6 ± 1.3  | 81.6 ± 1.3    | 78.0 ± 1.6     | 62.2 ± 0.8  | 61.7 ± 1.7    | 55.0 ± 1.1     |
| **MLP**        |             |               |                 |             |               |                 |             |               |                 |
| URE            | 83.7 ± 9.7  | 90.8 ± 4.7    | 82.9 ± 9.4     | 83.0 ± 3.2  | 74.8 ± 10.1   | 74.3 ± 10.1    | 68.5 ± 11.4 | 67.1 ± 7.7    | 57.2 ± 16.3    |
| SCEL           | 94.4 ± 0.5  | 93.7 ± 0.5    | 89.6 ± 0.9     | 84.4 ± 0.8  | 83.2 ± 1.1    | 78.9 ± 3.0     | 76.1 ± 1.3  | 73.9 ± 1.6    | 64.2 ± 1.2     |

A ± 0.1 A ± 0.2 A ± 0.5 A ± 0.1 A ± 0.2 A ± 0.5 A ± 0.1 A ± 0.2 A ± 0.5

Comparison of validation processes: Table 8 and 9 provide comparison of validation process using URE and the proposed SCEL. In Table 8, we observe that SCEL selects better parameters in most cases. We also observe that when the transition matrix is inaccurate, the parameters selected by SCEL tends to be more stable, especially when the base models are mlp. This demonstrates the superiority of SCEL despite not being an unbiased estimator of the classification accuracies. In Table 9, we further apply SCEL to Fwd. Similarly, we observe that SCEL selects better parameters in most cases. This suggests that the proposed validation procedure can not only be applied to CPE but also earlier approaches. It enables a more robust approach to validate earlier methods.

Training and validation loss curves: Figure 1 and 2 demonstrate the loss curve of the proposed CPE framework.
Fig. 1. Comparison of the training and validation loss of CPE with different transition layers in MNIST under different transition matrices. CPE-F and CPE-T perform almost identically, so the red lines and blue lines overlap in the figures. The shaded area denotes the standard deviation of five random trials.

Fig. 2. Comparison of the training and validation loss of CPE with different transition layers in MNIST under different noise level. CPE-F and CPE-T perform almost identically when $\lambda$ is small, so the red lines and blue lines overlap in those figures. The shaded area denotes the standard deviation of five random trials.