

國立臺灣大學電機資訊學院資訊工程學研究所

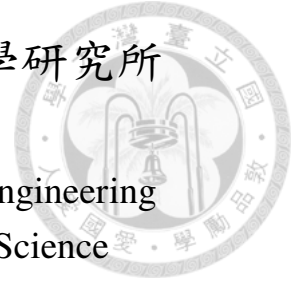
碩士論文

Department of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis



從互補標籤學習化約至機率估計

Reduction from Complementary-Label Learning to  
Probability Estimates

林瑋毅

Wei-I Lin

指導教授: 林軒田 博士

Advisor: Hsuan-Tien Lin, Ph.D.

中華民國 112 年 6 月

June, 2023



# Acknowledgements

First of all, I would like to express my gratitude to my supervisor, Dr. Hsuan-Tien Lin, for his invaluable guidance and support throughout this research journey. His mentorship extended beyond academia, as he generously shared his wisdom and vision, not only enriching my understanding in machine learning but also offering invaluable lessons on careers, life, and values. Each meeting with him was a source of inspiration, and I am truly grateful for the privilege of doing machine learning research in the free and comfortable way he provided.

I also thank my thesis committee members, Dr. Shou-De Lin, Dr. Hung-Yi Lee, and Dr. Shao-Hua Sun, for reviewing my thesis and providing a lot of insightful ideas for further exploration and improvement in my work.

Being a member of Computational Learning Lab has been a delightful and fortunate experience. I extend my appreciation to Poy Lu for engaging in vivid discussions, consistently listening to my argument, and providing authentic feedback to my research. I am grateful to Oscar Chew and Si-An Chen for their sharp yet constructive comments, which greatly enhance the quality of my work. Special thanks go to Andrew Bai, whose unwavering support always strengthened me during moments of doubt and confusion about my life and research. Additionally, I thoroughly enjoyed collaborating with Hsiu-Hsuan Wang on CLCIFAR, and I extend my thanks to all the CLLab members for their support and discussions.

I also thank National Center for High-performance Computing (NCHC) for providing computational and storage resources. The thesis cannot be done without their GPUs.

I am deeply grateful to my friends, especially Robin, Judy, and Ryan, from NTU Gymnastics, for creating countless unforgettable memories during my time at National Taiwan University.

Finally, my gratitude goes to my parents for letting me do the things I chose and enjoyed. I am forever grateful for everything they have done for me.



## 摘要

互補標籤學習 (Complementary-Label Learning, CLL) 是一個弱監督學習問題，其目標在於僅從互補標籤 (Complementary Labels) 訓練出一個分類器，其中互補標籤指是某個資料「不」屬於的類別。已知方法的主要思想是將此問題化約成一般的分類問題，並設計特殊的轉換以及代理損失函數使互補標籤可以與一般的分類問題連結，但這類的方法卻有一些缺點，例如容易過度擬合。在此論文中，我們設計一個新的框架「化約成互補標籤的分布估計」以避開先前方法可能有的缺點。我們證明了準缺地估計互補標籤的分布再加上一個簡單的解碼即可準確地分類未見過的資料。這個框架更可以解釋一些先前互補標籤學習的重要方法，並使他們在有雜訊的資料集中變得更穩健。此外，這個框架揭示了機率估計的準確度能夠用來驗證模型的準確度。由於此框架以機率估計為基礎，因此不論是深度模型或是傳統方法都能在此框架下進行互補標籤學習。我們同時以實驗驗證此框架在不同情境下皆有一定的準確度以及穩健性。最後，我們也收集、分析並公開了一個由真實人類標記，而非人工生成的互補標籤資料集：CLCIFAR。

**關鍵字：**互補標籤學習、弱監督學習、化約、監督式學習、機器學習



# 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. To further analyze the properties of complementary labels in real world, a CIFAR-based complementary dataset, CLCIFAR, was also collected, analyzed, and released publicly.

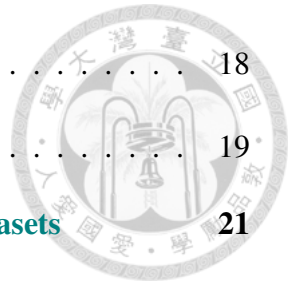
**Keywords:** Complementary-Label Learning, Weakly Supervised Learning, Reduction, Supervised Learning, Machine Learning



# Contents

	<b>Page</b>
<b>Acknowledgements</b>	<b>i</b>
<b>摘要</b>	<b>iii</b>
<b>Abstract</b>	<b>iv</b>
<b>Contents</b>	<b>vi</b>
<b>List of Figures</b>	<b>viii</b>
<b>List of Tables</b>	<b>ix</b>
<b>Chapter 1 Introduction</b>	<b>1</b>
<b>Chapter 2 Problem Setup</b>	<b>4</b>
2.1 Ordinary-label learning . . . . .	4
2.2 Complementary-label learning . . . . .	5
<b>Chapter 3 Proposed Framework</b>	<b>7</b>
3.1 Motivation . . . . .	7
3.2 Methodology . . . . .	8
3.3 Connection to Previous Methods . . . . .	13
<b>Chapter 4 Experiments</b>	<b>15</b>
4.1 Experiment Setup . . . . .	15
4.2 Discussion . . . . .	17

4.3	Learn from CL with Traditional Methods . . . . .	18
4.4	Comparison of validation processes . . . . .	19
<b>Chapter 5</b>	<b>CLCIFAR: Human-Annotated Complementary Datasets</b>	<b>21</b>
5.1	Motivation . . . . .	21
5.2	Data Collection Protocol . . . . .	22
5.3	Preliminary Dataset Analysis . . . . .	23
<b>Chapter 6</b>	<b>Conclusion</b>	<b>27</b>
	<b>References</b>	<b>28</b>
	<b>Appendix A — Proofs</b>	<b>31</b>
A.1	Proof of Proposition 3.2.1 . . . . .	31
A.2	Proof of Proposition 3.2.2 . . . . .	31
A.3	Proof of Corollary 3.2.3 . . . . .	32
	<b>Appendix B — Details of the Connections between Proposed Framework and Previous Methods</b>	<b>34</b>
	<b>Appendix C — Experiment Details</b>	<b>40</b>
C.1	Setup . . . . .	40
C.2	Additional Results . . . . .	41
	<b>Appendix D — Details of CLCIFAR20</b>	<b>44</b>
D.1	Label names of CLCIFAR20 . . . . .	44







# List of Figures

5.1	The label distribution of CLCIFAR10 and CLCIFAR20. . . . .	24
5.2	The empirical transition matrices of CLCIFAR10 and CLCIFAR20. . . . .	25
C.1	Comparison of the training and validation loss of CPE with different transition layers in MNIST under different transition matrices. . . . .	43
C.2	Comparison of the training and validation loss of CPE with different transition layers in MNIST under different noise level. . . . .	43



# List of Tables

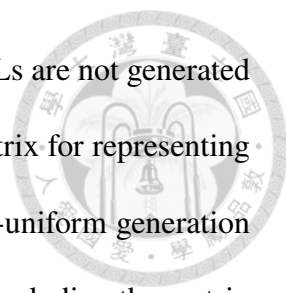
3.1	Comparison of recent approaches to CLL. . . . .	8
3.2	A unifying view of earlier approaches and proposed algorithms through the lens of reduction to probability estimates. . . . .	14
4.1	Comparison of the testing classification accuracies with different transition matrices and different noise levels. . . . .	16
4.2	Comparison of testing accuracies of decoders when the baseline models use fixed transition layers. . . . .	16
4.3	Comparison of testing accuracies of CPE with traditional models. . . . .	18
4.4	Comparison of <b>CPE-T</b> 's testing accuracies with different validation procedures. . . . .	19
4.5	Comparison of <b>Fwd</b> 's testing accuracies with different validation procedures. . . . .	19
C.1	Comparison of the testing classification accuracies with different transition matrices. . . . .	42
C.2	Comparison of the testing classification accuracies with different levels of noise. . . . .	42



# Chapter 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 [15]. 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 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, 12] and domain-adaptation [14].

Ishida et al. [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. Chou et al. [1] unveiled the overfitting tendency of URE and proposed the surrogate complementary loss (SCL) as



an alternative design. Yu et al. [13] 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. Gao and Zhang [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, 4]. 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 [13] 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. To understand the properties of complementary labels in the real world, we also collected and analyzed a human-annotated complementary dataset, CLCIFAR. 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.

It is worth noting that some of the results in Chapter 5 are jointly developed by Hsiu-Hsuan Wang and the author [9]. The results that should be credited to Hsiu-Hsuan Wang will be properly acknowledged in the coming chapters. The results without such acknowledgment are the original contributions of the author.



## Chapter 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  $\mathcal{Y} = [K] = \{1, 2, \dots, K\}$  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.  $\hat{y} = \operatorname{argmax}_i f_i(x)$ , 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:  $\mathbb{E}_{(x,y) \sim D} [\ell(f(x), e_y)]$ , where  $\ell: \mathbb{R}^K \times \mathbb{R}^K \rightarrow \mathbb{R}^+$  denotes the loss function, and  $e_y$  denotes the one-hot vector of label  $y$ .



## 2.2 Complementary-label learning

In complementary-label learning (CLL), the goal for the learning algorithm remains to find an  $f$  that minimizes the ordinary classification risk. The difference lies in the dataset to learn from. The CLL algorithm does not have access to the ground-truth labels  $y_i$ . Instead, for each instance  $x_i$ , the algorithm is given a complementary label  $\bar{y}_i$ . A complementary label is a class that  $x_i$  does not belong to; that is,  $\bar{y}_i \in [K] \setminus \{y_i\}$ . 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  $\bar{p}_{\bar{D}}(x, y)$ . Given access to i.i.d. samples  $\{x_i, \bar{y}_i\}_{i=1}^N$  from  $\bar{D}$ , the CLL algorithm 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, 13]. It assumes that the generation of complementary labels only depends on the ordinary labels; that is,  $P(\bar{y} | y, x) = P(\bar{y} | y)$ . The transition probability  $P(\bar{y} | y)$  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  $T = \frac{1}{K-1}(\mathbf{1}_k - \mathbf{I}_k)$ . 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 physical meaning in the context of CLL; however, it plays an important role in some theoretical analysis in

previous work [1, 13].

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 scenario, 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.







## Chapter 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, 13] mainly focus on applying different transformation and surrogate loss functions to the ordinary classifier, as summarized in Table 3.1. This paradigm of reduction to *ordinary*-label learning, however, faces some limitations. For instance, as Chou et al. [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*.

Table 3.1: 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 [13]	$\phi(f)(x) = T^\top f(x)$	$\ell(\phi(f)(x), \bar{y})$
DM [2]	$\phi(f)(x) = \text{sm}(1 - f(x))$	$\ell(\phi(f)(x), \bar{y})$

## 3.2 Methodology

**Overview** The proposed method consists of two steps: In training phase, we aim to find a hypothesis  $\bar{f}$  that predicts the distribution of the complementary labels well, i.e., an  $\bar{f}$  that approximates  $P(\bar{y} | x)$ . This step is motivated by Yu et al. [13] and Gao and Zhang [2], who proposed to model the conditional distribution of the complementary labels  $P(\bar{y} | x)$ , and Zhang et al. [14], who applied 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  $\hat{y} = \operatorname{argmin}_{k \in [K]} d(\bar{f}(x), T_k)$  for an unseen instance  $x$ , where  $d$  denotes a loss function. It is a natural choice to decode with respect to  $T$  because the transition vector  $T_k = (P(\bar{y} = 1 | y = k), \dots, P(\bar{y} = K | y = k))^\top$  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  $\bar{f}$  that predicts  $P(\bar{y} | x)$  well. To do so, given a hypothesis  $\bar{f}$  from hypothesis set  $\bar{\mathcal{H}}$ , we set the following *complementary estimation loss* to optimize:

$$R(\bar{f}; \ell) = \mathbb{E}_{(x,y) \sim \mathcal{D}} (\ell(\bar{f}(x), P(\bar{y} | x, y))) \quad (3.1)$$

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  $P(\bar{y} | x, y)$  is  $T_y$ , so we can rewrite Equation (3.1) as follows:

$$R(\bar{f}; \ell) = \mathbb{E}_{(x,y) \sim \mathcal{D}} (\ell(\bar{f}(x), T_y)) \quad (3.2)$$

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 3.2.1.** *There is a constant  $C$  such that*

$$\mathbb{E}_{(x,\bar{y}) \sim \bar{\mathcal{D}}} \ell(\bar{f}(x), e_{\bar{y}}) + C = \mathbb{E}_{(x,y) \sim \mathcal{D}} \ell(\bar{f}(x), T_y) \quad (3.3)$$

holds for all hypothesis  $\bar{f} \in \bar{\mathcal{H}}$  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 [5, 11]. It allows us to replace  $T_y$  by  $e_{\bar{y}}$  in Equation (3.2) 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 (3.2) becomes equivalent to minimizing the

following *surrogate complementary estimation loss (SCEL)*:

$$\bar{R}(\bar{f}; \ell) = \mathbb{E}_{(x, \bar{y}) \sim \bar{\mathcal{D}}} (\ell(\bar{f}(x), e_{\bar{y}})) \quad (3.4)$$



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

$$\bar{R}(\bar{f}; \ell) = \mathbb{E}_{(x, \bar{y}) \sim \bar{\mathcal{D}}} (-\log \bar{f}_{\bar{y}}(x)) \quad (3.5)$$

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  $\bar{f}^* = \operatorname{argmin}_{\bar{f} \in \bar{\mathcal{H}}} \frac{1}{N} \sum_{i=1}^N -\log \bar{f}_{\bar{y}_i}(x_i)$ , 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  $T_{\hat{y}}$  is closest to the

predicted complementary probability estimates. That is, the label is predicted by

$$\hat{y} = \operatorname{argmin}_{k \in [K]} d(\bar{f}^*(x), T_k) \quad (3.6)$$

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  $\operatorname{dec}(\bar{f}; d)$  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  $R_{01}(f) = \mathbb{E}_{(x,y) \sim \mathcal{D}} I_{f(x) \neq y}$ .

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

$$\operatorname{dec}(\bar{f}; L_1)(x) = \operatorname{argmin}_{y \in [K]} \|T_y - \bar{f}(x)\|_1 \quad (3.7)$$

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 3.2.2.** *For any  $\bar{f} \in \bar{\mathcal{H}}$ , and distance function  $d$  defined on the probability simplex  $\Delta^K$ , it holds that*

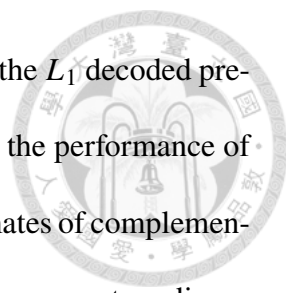
$$R_{01}(\operatorname{dec}(\bar{f}; d)) \leq \frac{2}{\gamma_d} R(\bar{f}; d) \quad (3.8)$$

where  $\gamma_d = \min_{i \neq j} d(T_i, T_j)$  is the minimal distance between any pair of transition vector.

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

$$R_{01}(\operatorname{dec}(\bar{f}; L_1)) \leq \frac{4\sqrt{2}}{\gamma} \sqrt{R(\bar{f}; \ell)} \quad (3.9)$$

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  $R(\bar{f}; \ell_{\text{KL}})$  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 ( $\bar{f} \rightarrow \bar{f}^*$ ) then the  $L_1$  decoded prediction is consistent ( $R_{01}(\text{dec}(\bar{f}; L_1)) \rightarrow 0$ ). 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,  $R(\bar{f}; \ell_{\text{KL}})$  contains irreducible error. We leave the analysis of the error bound in this case for the future research.

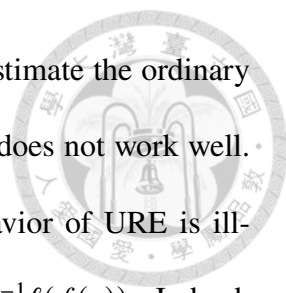
Another implication of the Proposition 3.2.2 is related to the inaccurate transition matrix. Suppose the complementary labels are generated with respect to the transition matrix  $T'$ , which may be different from  $T$ , the one provided to the learning algorithm. In the proposed framework, the only affected component is the decoding step. This allows us to quantify the effect of inaccuracy as follows:

**Corollary 3.2.3.** *For any  $\bar{f} \in \bar{\mathcal{H}}$ , if  $d$  is the  $L_1$  distance and  $\ell$  is the KL divergence, then*

$$R_{01}(\text{dec}(f; L_1)) \leq \frac{4\sqrt{2}}{\gamma} \sqrt{R(\bar{f}; \ell)} + \frac{2\epsilon}{\gamma}. \quad (3.10)$$

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 3.2.2 is an alternative validation procedure to the unbiased risk estimation (URE) [3]. According to Proposition 3.2.2, selecting the best-performing parameter minimizes the right hand side of Eq. (3.9) 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 not work well. For instance, when the transition matrix is non-invertible, the behavior of URE is ill-defined due to the presence of  $T^{-1}$  in the formula of URE:  $\mathbb{E}_{x,\bar{y}} e_{\bar{y}} T^{-1} \ell(f(x))$ . Indeed, replacing  $T^{-1}$  with  $T$ 's pseudo-inverse can avoid the issue; however, it remains unclear whether the unbiasedness of URE still holds after using pseudo-inverse. In contrast, the quality of complementary probability estimates sidesteps the issue because it does not need to invert the transition matrix. This prevents the proposed procedure from the issue of an ill-conditioned 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) [13], (2) Surrogate Complementary Loss (SCL) with log loss [1], and (3) Discriminative Model (DM) [2], which are explained in Table 3.2 and Appendix A.3. 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

Table 3.2: A unifying view of earlier approaches and proposed algorithms through the lens of reduction to probability estimates, where  $U$  denote the uniform transition matrix. Two versions of Forward Correction are considered: General  $T$  denotes the original version in [13], and the Uniform denotes the case when the transition layer is fixed to be uniform. Proof of the equivalence is in Appendix A.3.

Method	Hypothesis set	Decoder
Fwd (general $T$ ) [13]	$\{x \mapsto T^\top f(x; \theta) : \theta \in \Theta\}$	$\operatorname{argmax}_k ((T^\top)^{-1} \tilde{f}(x))_k$
Fwd (uniform) [13]	$\{x \mapsto U^\top f(x; \theta) : \theta \in \Theta\}$	$\operatorname{argmin}_k \ \tilde{f}(x) - U_k\ _1$
SCL [1]	$\{x \mapsto U^\top f(x; \theta) : \theta \in \Theta\}$	$\operatorname{argmin}_k \ \tilde{f}(x) - U_k\ _1$
DM [2]	$\{x \mapsto \operatorname{sm}(1 - f(x; \theta)) : \theta \in \Theta\}$	$\operatorname{argmin}_k \ \tilde{f}(x) - U_k\ _1$
CPE-I (no transition)	$\{x \mapsto f(x; \theta) : \theta \in \Theta\}$	$\operatorname{argmin}_k \ \tilde{f}(x) - T_k\ _1$
CPE-F (fixed transition)	$\{x \mapsto T^\top f(x; \theta) : \theta \in \Theta\}$	$\operatorname{argmin}_k \ \tilde{f}(x) - T_k\ _1$
CPE-T (trainable transition)	$\{x \mapsto T(W)^\top f(x; \theta) : \theta \in \Theta, W \in \mathbb{R}^{K \times K}\}$	$\operatorname{argmin}_k \ \tilde{f}(x) - T_k\ _1$

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*. 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  $T(W) = (\operatorname{sm}(W_1), \operatorname{sm}(W_2), \dots, \operatorname{sm}(W_K))^\top$ . 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**.





## Chapter 4 Experiments

In this section, we benchmarked 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 Chapter 4.3 and verify the effectiveness of the proposed validation procedure in the Appendix.

### 4.1 Experiment Setup

**Baseline and setup** We first evaluated 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 [13], (c) **SCL**: Surrogate Complementary Loss with negative log loss [1], and (d) **DM**: Discriminative Models with Weighted Loss [2]. Following the previous work, we tested those methods on MNIST, Fashion-MNIST, and Kuzushiji-MNIST, and use one-layer mlp model (d-500-c) as base models. All models were 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 was selected with the Surrogate Complementary Estimation Loss (SCEL) on the validation dataset. For the baseline method, it was selected with unbiased risk estimator (URE) of the zero-one loss. It is worth noting that

Table 4.1: Comparison of the testing classification accuracies with different transition matrices (upper part) and different noise levels (lower part).

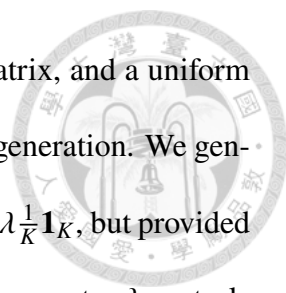
	MNIST			Fashion-MNIST			Kuzushiji-MNIST		
	Unif.	Weak	Strong	Unif.	Weak	Strong	Unif.	Weak	Strong
URE-GA	90.3± 0.2	87.8± 0.9	33.8± 8.1	79.4± 0.7	75.7± 2.0	32.3± 4.5	65.6± 0.8	62.5± 1.1	23.3± 5.4
SCL	94.3± 0.4	<b>93.8± 0.4</b>	27.5± 19.8	82.6± 0.4	81.2± 0.1	28.5± 10.8	<b>73.7± 1.4</b>	<b>71.2± 2.9</b>	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	24.8± 5.0	65.6± 2.9	64.5± 2.7	20.1± 3.2
Fwd	<b>94.4± 0.2</b>	91.9± 0.3	95.3± 0.4	82.6± 0.6	<b>83.0± 1.0</b>	85.5± 0.3	73.5± 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	<b>94.4± 0.2</b>	92.0± 0.2	<b>95.5± 0.3</b>	<b>83.0± 0.1</b>	<b>83.0± 0.3</b>	<b>85.8± 0.3</b>	73.5± 1.6	64.6± 0.5	<b>75.3± 2.6</b>
CPE-T	92.8± 0.6	92.1± 0.2	95.2± 0.5	<b>83.0± 0.1</b>	<b>83.0± 0.3</b>	<b>85.8± 0.3</b>	63.6± 0.4	64.6± 0.4	74.2± 2.8
	$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.5$	$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.5$	$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.5$
URE-GA	31.8± 6.4	27.8± 8.2	28.1± 4.1	27.3± 5.5	28.6± 4.1	26.3± 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± 9.2	20.6± 6.7	23.2± 5.7	20.4± 4.6	17.3± 2.9	16.8± 1.6
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± 3.0	19.2± 3.1	18.2± 1.6
Fwd	88.3± 8.7	83.9± 10.7	71.6± 18.4	<b>84.8± 0.6</b>	80.2± 6.2	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	78.2± 1.5	73.0± 0.7	71.6± 0.9	62.7± 1.6
CPE-F	94.3± 0.5	93.6± 0.5	89.0± 1.4	84.1± 0.8	83.0± 1.1	78.4± 2.5	<b>76.1± 1.3</b>	73.7± 1.5	63.7± 1.5
CPE-T	<b>94.4± 0.5</b>	<b>93.7± 0.5</b>	<b>89.6± 0.9</b>	84.1± 0.8	<b>83.2± 1.1</b>	<b>78.9± 2.0</b>	<b>76.1± 1.3</b>	<b>73.9± 1.6</b>	<b>64.2± 1.2</b>

Table 4.2: 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.

	MNIST			Fashion-MNIST			Kuzushiji-MNIST		
	Unif.	Weak	Strong	Unif.	Weak	Strong	Unif.	Weak	Strong
$M_{\text{MAX}}$	94.4± 0.2	92.0± 0.2	95.5± 0.2	83.0± 0.1	<b>83.3± 0.2</b>	<b>86.1± 0.5</b>	73.5± 1.6	<b>64.8± 0.5</b>	75.3± 2.6
$L_1$	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
	$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.5$	$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.5$	$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.5$
$M_{\text{MAX}}$	<b>94.4± 0.3</b>	93.5± 0.3	84.5± 4.1	<b>85.0± 0.3</b>	<b>84.0± 0.5</b>	76.5± 2.5	<b>76.4± 1.1</b>	<b>73.8± 1.2</b>	59.9± 3.4
$L_1$	94.3± 0.5	<b>93.6± 0.5</b>	<b>89.0± 1.4</b>	84.1± 0.8	83.0± 1.1	<b>78.4± 2.5</b>	76.1± 1.3	73.7± 1.5	<b>63.7± 1.5</b>

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 were benchmarked in the experiment. Besides the uniform transition matrix, following [2, 13], 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 were set to  $p_1$ ,  $p_2$  and  $p_3$ , respectively. We considered two cases for  $(p_1, p_2, p_3)$ : (a) *Strong*:  $(\frac{0.75}{3}, \frac{0.24}{3}, \frac{0.01}{3})$  to model stronger deviation from uniform transition matrices. (b) *Weak*:  $(\frac{0.45}{3}, \frac{0.30}{3}, \frac{0.25}{3})$  to model milder deviation from uniform transition matrices. In the experiment of *noisy* transition matrices, we considered 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 4.1.

## 4.2 Discussion

**Can Transition Layers Improve Performance?** The answer is positive in both clean and noisy experiments. We observed that **CPE-F** and **CPE-T** outperformed **CPE-I** in both settings, demonstrating that the transition layer helps achieve higher performances, no matter the provided transition matrix is clean or not. Also, we observed that **CPE-T** outperformed **CPE-F** in the noisy setting, especially when the noise factor  $\lambda$  was large. It demonstrated that by making transition layers trainable, the model could potentially fit the distribution of complementary labels better by altering the transition layer. In contrast, **CPE-F** was 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 Chapter 3.3, **Fwd** and **CPE-F** only differed 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 were **CPE-F** in Table 4.2. It is displayed that the **MAX** decoder outperformed  $L_1$  in most noiseless settings; however, when the transition matrix was highly inaccurate ( $\lambda = 0.5$ ), we observed that the  $L_1$  decoder outperformed the **MAX** decoder. This suggests that  $L_1$  could be more tolerant to an inaccurate transition matrix. These results reveal that a deeper sensitivity analysis of

Table 4.3: Comparison of testing accuracies of CPE with traditional models. **Boldfaced** ones outperform the baseline methods based on single-layer deep models.

Model	MNIST			Fashion-MNIST			Kuzushiji-MNIST		
	Unif.	Weak	Strong	Unif.	Weak	Strong	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	<b>74.9± 0.8</b>	<b>73.7± 0.8</b>	<b>80.4± 1.3</b>
CPE-GBDT	86.9± 0.4	86.0± 0.3	90.3± 0.9	79.8± 0.4	78.0± 0.4	81.4± 1.1	60.6± 0.4	56.6± 1.8	68.4± 2.1
	$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.5$	$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.5$	$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.5$
CPE-KNN	93.7± 0.4	93.4± 0.4	<b>91.9± 1.1</b>	78.7± 1.9	78.5± 1.9	76.6± 1.9	<b>77.2± 1.1</b>	<b>75.9± 1.6</b>	<b>73.2± 1.7</b>
CPE-GBDT	89.7± 1.0	88.6± 1.2	84.0± 1.7	80.6± 1.7	80.0± 1.6	76.0± 2.2	66.7± 2.4	64.7± 2.4	55.8± 3.1

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 performed well when the transition matrix was just slightly deviated from the uniform one, but their performances all dropped when the deviation from uniform becomes larger. As we discussed in Chapter 3.3, the result could 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** had similar testing accuracies across different transition matrices, demonstrating that **CPE** did exploit the information from the transition matrix that helped the models deliver better performance.

### 4.3 Learn from CL with Traditional Methods

As discussed in Chapter 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 4.3. It

Table 4.4: Comparison of **CPE-T**'s testing accuracies with different validation procedures.

	MNIST			Fashion-MNIST			Kuzushiji-MNIST		
	Unif.	Weak	Strong	Unif.	Weak	Strong	Unif.	Weak	Strong
linear									
URE	90.3± 0.6	90.4± 0.3	91.8± 0.5	<b>82.1± 0.3</b>	81.5± 1.2	82.6± 1.3	59.9± 0.4	60.0± 0.9	62.5± 0.5
SCEL	<b>90.5± 0.2</b>	<b>90.6± 0.1</b>	91.8± 0.4	82.0± 0.3	<b>82.1± 0.5</b>	<b>83.2± 1.2</b>	<b>60.3± 0.5</b>	<b>60.6± 0.5</b>	<b>63.0± 0.3</b>
mlp									
URE	92.7± 0.5	91.8± 0.7	90.4± 6.5	82.9± 0.1	83.0± 0.3	84.3± 1.5	<b>63.8± 0.7</b>	63.8± 1.9	<b>74.5± 2.7</b>
SCEL	<b>92.8± 0.6</b>	<b>92.1± 0.2</b>	<b>95.2± 0.5</b>	<b>83.0± 0.1</b>	83.0± 0.3	<b>85.8± 0.3</b>	63.6± 0.4	<b>64.6± 0.4</b>	74.2± 2.8
	$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.5$	$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.5$	$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.5$
linear									
URE	90.9± 1.0	90.2± 0.8	<b>86.1± 1.3</b>	82.2± 1.3	81.2± 1.4	77.1± 1.8	<b>62.3± 0.8</b>	60.6± 0.9	<b>55.3± 2.3</b>
SCEL	<b>91.3± 0.7</b>	<b>90.5± 0.8</b>	85.7± 1.6	<b>82.6± 1.3</b>	<b>81.6± 1.3</b>	<b>78.0± 1.6</b>	62.2± 0.8	<b>61.7± 1.7</b>	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	<b>94.4± 0.5</b>	<b>93.7± 0.5</b>	<b>89.6± 0.9</b>	<b>84.1± 0.8</b>	<b>83.2± 1.1</b>	<b>78.9± 2.0</b>	<b>76.1± 1.3</b>	<b>73.9± 1.6</b>	<b>64.2± 1.2</b>

Table 4.5: Comparison of **Fwd**'s testing accuracies with different validation procedures.

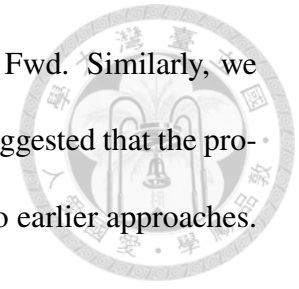
	MNIST			Fashion-MNIST			Kuzushiji-MNIST		
	Unif.	Weak	Strong	Unif.	Weak	Strong	Unif.	Weak	Strong
linear									
URE	90.5± 0.2	90.6± 0.4	91.6± 0.7	82.0± 0.4	81.6± 1.2	83.4± 0.7	59.9± 0.9	60.4± 0.9	62.6± 0.7
SCEL	90.5± 0.2	<b>90.7± 0.2</b>	<b>91.9± 0.4</b>	<b>82.2± 0.3</b>	<b>82.6± 0.3</b>	<b>83.8± 0.2</b>	<b>60.4± 0.6</b>	<b>61.2± 0.3</b>	<b>63.2± 0.2</b>
mlp									
URE	94.4± 0.2	91.9± 0.3	95.3± 0.4	82.6± 0.6	83.0± 1.0	85.5± 0.3	73.5± 1.6	63.1± 2.6	74.1± 4.8
SCEL	94.4± 0.2	<b>92.0± 0.2</b>	<b>95.5± 0.2</b>	<b>83.0± 0.1</b>	<b>83.3± 0.2</b>	<b>86.1± 0.5</b>	73.5± 1.6	<b>64.8± 0.5</b>	<b>75.3± 2.6</b>
	$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.5$	$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.5$	$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.5$
linear									
URE	91.1± 0.7	89.6± 1.0	82.5± 3.6	82.4± 0.9	81.4± 0.9	72.0± 7.5	<b>62.7± 1.0</b>	60.9± 0.9	52.1± 6.2
SCEL	<b>91.4± 0.5</b>	<b>90.5± 0.5</b>	<b>83.9± 2.6</b>	<b>83.2± 0.3</b>	<b>82.4± 0.4</b>	<b>76.3± 2.8</b>	62.5± 0.9	<b>62.5± 1.6</b>	<b>55.6± 2.0</b>
mlp									
URE	88.3± 8.7	83.9± 10.7	71.6± 18.4	84.8± 0.6	80.2± 6.2	62.9± 20.1	72.8± 5.6	67.6± 7.5	54.7± 12.4
SCEL	<b>94.4± 0.3</b>	<b>93.5± 0.3</b>	<b>84.5± 4.1</b>	<b>85.0± 0.3</b>	<b>84.0± 0.5</b>	<b>76.5± 2.5</b>	<b>76.4± 1.1</b>	<b>73.8± 1.2</b>	<b>59.9± 3.4</b>

displayed that traditional models, specifically,  $k$ -NN, outperformed all the methods using deep models in Kuzushiji-MNIST, indicating the benefit of the proposed CPE's flexibility in using non-deep models.

## 4.4 Comparison of validation processes

Table 4.4 and 4.5 provide comparison of validation process using URE and the proposed SCEL. In Table 4.4, we observed that SCEL selected better parameters in most cases. We also observed that when the transition matrix was inaccurate, the parameters selected by SCEL tended to be more stable, especially when the base models were mlp. This demonstrated the superiority of SCEL despite not being an unbiased estimator of the

classification accuracies. In Table 4.5, we further applied SCEL to Fwd. Similarly, we observed that SCEL selected better parameters in most cases. This suggested that the proposed validation procedure could not only be applied to CPE but also earlier approaches. It enables a more robust approach to validate earlier methods.





# Chapter 5 CLCIFAR: Human-Annotated Complementary Datasets

In this chapter, we introduce CLCIFAR, a CIFAR-based complementary dataset annotated by humans. In Section 5.1, we discuss the motivation for a human-annotated complementary dataset. In Section 5.2, we propose a protocol to collect complementary labels from human annotators. Finally, in Section 5.3, a preliminary analysis on the collected dataset is provided. <sup>1</sup>

## 5.1 Motivation

As mentioned in Chapter 1, many proponents of studying CLL often highlight its potential on reducing annotation costs by collecting complementary labels instead of ordinary labels. The argument roots from the fact that any multi-class instance is associated with more complementary labels than the one ordinal label. Nevertheless, the complementary labels contain less information than ordinary labels, and hence more complementary labels may be needed to achieve the same level of testing performance. It remains unclear whether in practice the learning algorithms can produce a meaningful classifier when the

---

<sup>1</sup>All the results in this chapter are significantly original contributions of the author except that Hsiu-Hsuan Wang deployed the protocol on MTurk and conducted the preliminary analysis. The protocol and analysis design comes from joint discussion between Hsiu-Hsuan Wang, Hsuan-Tien Lin and the author [9].

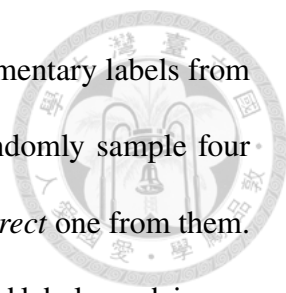
label information is not only very weak but potentially noisy. On the other hand, it remains unknown whether the class-conditional assumption that was utilized frequently in the literature hold true in practice and whether violation of these assumptions will affect the performance of the previous algorithms.

To answer the problems mentioned above and contribute to the community, we devised a label collection protocol that allows the annotators to choose a complementary label for the images in CIFAR10 and CIFAR100, then analyzed the collected complementary labels.

## 5.2 Data Collection Protocol

**Dataset Selection** We base our complementary datasets on CIFAR10 and CIFAR100. This selection is motivated by the real-world noisy label dataset by Wei et al. [10]. Building upon the CIFAR datasets allow us to evaluate the noise rate and the empirical transition matrix easily, as they already contain nearly noise-free ordinary labels. Besides, most of the SOTA CLL algorithms already perform benchmark on the CIFAR datasets, albeit using synthetic labels. This allows us to benchmark those methods without putting much efforts on selecting network architecture or tuning the training hyperparameters. Finally, CIFAR datasets are sufficiently hard in two aspects. For CLL algorithms, they are demonstrated to be learnable at least in a noise-free and uniform scenario, while they are still struggling to perform well on larger datasets such as ImageNet. For humans, the image labeling tasks are also hard enough to argue that annotating complementary labels are easier than the ordinary labels. In contrast, it is hard to believe that correctly annotating the digits in MNIST is challenging for humans. These observation makes us to base our complementary datasets on the CIFAR dataset.





**Complementary label collection protocol** To collect only complementary labels from the CIFAR dataset, for each image in the training split, we first randomly sample four distinct labels and ask the human annotators to select any of the *incorrect* one from them. To analyze the annotators' behavior and reduce the noise in the collected labels, each image is labeled by three different annotators. The four labels are re-sampled for each annotator on each image. That is, each annotator possibly receives a different set of four labels to choose from. Note that if the annotators always select one of the correct complementary labels randomly, the empirical transition matrix will be uniform in expectation. We will inspect the empirical transition matrix in the next section.

The labeling tasks are deployed on MTurk. We first divide the 50,000 images into five batches of 10,000 images. Then, each batch is further divided into 1,000 human intelligence tasks (HITs) with each HIT containing 10 images. Each HIT is deployed to three annotators, who receive 0.03 dollar as the reward by annotating 10 images. To make the labeling task easier and increase clarity, the size of the images are enlarged to  $200 \times 200$  pixels. For each super-class in CIFAR20, four to six example images from the classes within the super-class are provided to the annotators for reference.

### 5.3 Preliminary Dataset Analysis

Next, we take a closer look at the collected complementary labels. We first analyze the error rates of the collected labels, and then verify whether the transition matrix is uniform or not. Finally, we end with an analysis on the behavior of the human annotators observed in the label collection protocol.

**Observation 1: noise rate compared to ordinary label collection** We first look at the noise rate of the collected complementary labels. A complementary label is considered to

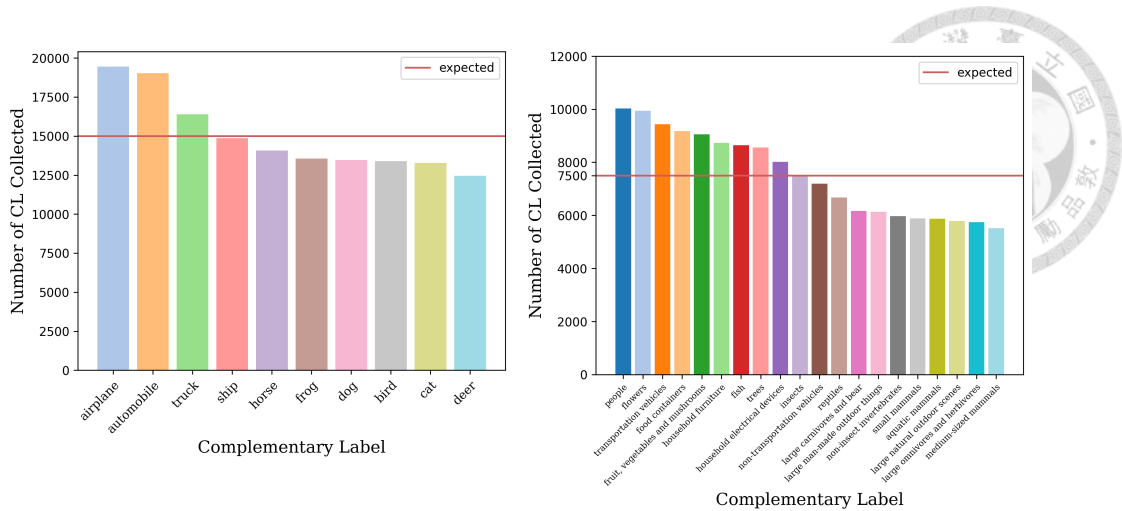
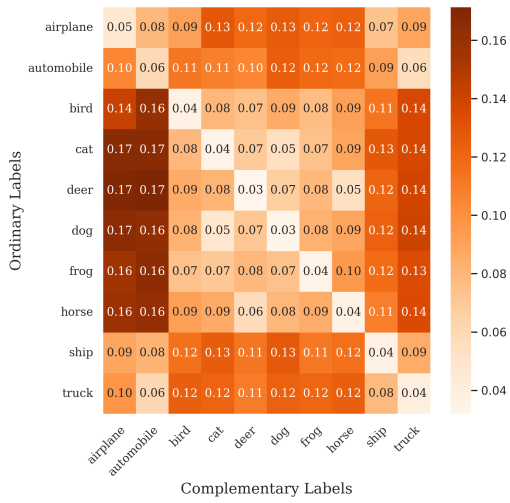
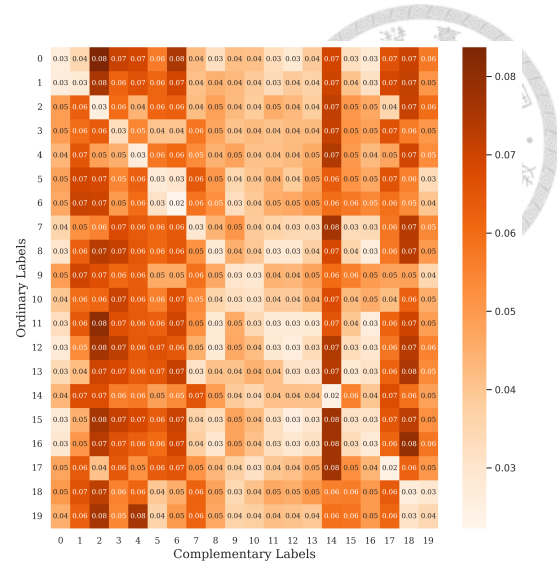


Figure 5.1: The label distribution of CLCIFAR10 (left) and CLCIFAR20 (right).

be incorrect if it is actually the ordinary label. The mean error rate made by the human annotators is 3.93% for CLCIFAR10 and 2.80% for CLCIFAR20. Although it is not a fair comparison due to the different protocols, we compare to the noise rate of the CIFAR-N dataset [10] for reference. The noise rate on CIFAR10-N and CIFAR100N-coarse are around 18% and 25.60%, respectively. This difference suggests that the collected complementary labels could be less noisy than the ordinary ones. On the other hand, if we compare the human annotators to a random annotator who always annotates the label randomly, the results become different. A random annotator achieves a noise rate of  $\frac{1}{K}$  for complementary label annotation and a noise rate of  $\frac{K-1}{K}$  for ordinary label annotation. If we compare the human annotators to a random annotator, then for CLCIFAR10, human annotators have 60.7% less noisy labels than the random annotator whereas for CIFAR10-N, human annotators have 80% less noisy labels. This demonstrates that human annotators are more competent compared to a random annotator in the ordinary-label annotation. Similarly, human annotators have 44% less noise than a random annotator for CLCIFAR20 and 73.05% less noise for CIFAR100N-coarse. This observation reveals that while the absolute noise rate is lower in annotating complementary labels, it may be more difficult to be competent against random labels than the ordinary label annotation.



(a) CLCIFAR10



(b) CLCIFAR20

Figure 5.2: The empirical transition matrices of CLCIFAR10 and CLCIFAR20. The label names of CLCIFAR20 are abbreviated as indexes to save space. The full label names are provided in Appendix D.1.

**Observation 2: imbalanced complementary label annotation** Next, we analyze the distribution of the collected complementary labels. The frequency of the complementary labels for the CLCIFAR datasets are reported in Figure 5.1. As we can see in the figure, the annotators have specific bias on certain labels. For instance, the annotators have a preference for “airplane” and “automobile” in CLCIFAR10 and a preference for “people” and “flower” in CLCIFAR20. In CLCIFAR10, the annotations are biased towards the labels with longer names whereas in CLCIFAR20, they are biased towards the labels with shorter, more concrete and understandable names.

**Observation 3: biased transition matrix** Finally, we visualize the empirical transition matrix using the collected complementary labels in Figure 5.2. Based on the first two observations, we could imagine that the transition matrix is biased. By inspecting Figure 5.2, we further discover that the bias in the complementary labels are dependent on the true labels. For instance, in CLCIFAR10, despite we see more annotations on airplane and automobile in aggregate, conditioning on the transportation-related labels (“airplane”,

“automobile”, “ship” and “truck”), the distribution of the complementary labels becomes more biased towards other animal-related labels (“bird”, “cat”, etc.)





## Chapter 6 Conclusion

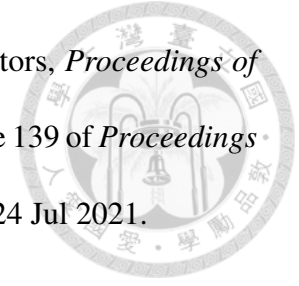
In this paper, we view the CLL problem from a novel perspective, reduction to complementary probability estimates. Through this perspective, we propose a framework that only requires complementary probability estimates and prove that a simple decoding step can map the estimates to ordinary labels. The 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. A real-world complementary dataset, CLCIFAR, is also collected and analyzed. We expect the realistic elements of the paper to keep inspiring future research towards making CLL practical.



## References

- [1] Y.-T. Chou, G. Niu, H.-T. Lin, and M. Sugiyama. Unbiased risk estimators can mislead: A case study of learning with complementary labels. In *International Conference on Machine Learning*, pages 1929–1938. PMLR, 2020.
- [2] Y. Gao and M.-L. Zhang. Discriminative complementary-label learning with weighted loss. In *International Conference on Machine Learning*, pages 3587–3597. PMLR, 2021.
- [3] T. Ishida, G. Niu, W. Hu, and M. Sugiyama. Learning from complementary labels. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 5644–5654, 2017.
- [4] T. Ishida, G. Niu, A. Menon, and M. Sugiyama. Complementary-label learning for arbitrary losses and models. In *International Conference on Machine Learning*, pages 2971–2980. PMLR, 2019.
- [5] M. Kull and P. Flach. 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*, pages 68–85. Springer, 2015.
- [6] X. Li, T. Liu, B. Han, G. Niu, and M. Sugiyama. Provably end-to-end label-noise

learning without anchor points. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 6403–6413. PMLR, 18–24 Jul 2021.



- [7] J. Liu, H. Hang, B. Wang, B. Li, H. Wang, Y. Tian, and Y. Shi. GAN-CL: Generative adversarial networks for learning from complementary labels. *IEEE Transactions on Cybernetics*, 2021.
- [8] D.-B. Wang, L. Feng, and M.-L. Zhang. Learning from complementary labels via partial-output consistency regularization. In *IJCAI*, pages 3075–3081, 2021.
- [9] H.-H. Wang, W.-I. Lin, and H.-T. Lin. CLCIFAR: CIFAR-derived benchmark datasets with human annotated complementary labels. *arXiv preprint arXiv:2305.08295*, 2023.
- [10] J. Wei, Z. Zhu, H. Cheng, T. Liu, G. Niu, and Y. Liu. Learning with noisy labels revisited: A study using real-world human annotations. In *International Conference on Learning Representations*, 2021.
- [11] R. C. Williamson, E. Vernet, and M. D. Reid. Composite multiclass losses. *Journal of Machine Learning Research*, 17(222):1–52, 2016.
- [12] Y. Xu, M. Gong, J. Chen, T. Liu, K. Zhang, and K. Batmanghelich. Generative-discriminative complementary learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6526–6533, 2020.
- [13] X. Yu, T. Liu, M. Gong, and D. Tao. Learning with biased complementary labels. In *Proceedings of the European conference on computer vision (ECCV)*, pages 68–83, 2018.

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



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





# Appendix A — Proofs

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

## A.1 Proof of Proposition 3.2.1

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

$$\mathbb{E}_{(x,y) \sim \mathcal{D}} \ell(\bar{f}(x), T_y) = \mathbb{E}_{(x,y) \sim \mathcal{D}} \sum_{k=1}^K -T_{yk} \log\left(\frac{\bar{f}_k(x)}{T_{yk}}\right) = C + \mathbb{E}_{(x,y) \sim \mathcal{D}} \sum_{k=1}^K -T_{yk} \log(\bar{f}_k(x)) \quad (\text{A.1})$$

Next, as  $P(\bar{y} | y) = T_{y\bar{y}}$ , then

$$\mathbb{E}_{(x,y) \sim \mathcal{D}} \sum_{k=1}^K -T_{yk} \log(\bar{f}_k(x)) = \mathbb{E}_{(x,y) \sim \mathcal{D}} \left( \mathbb{E}_{\bar{y} | y} -\log(\bar{f}_{\bar{y}}(x)) \right) = \mathbb{E}_{(x,\bar{y}) \sim \bar{\mathcal{D}}} \ell(\bar{f}(x), e_{\bar{y}}) \quad (\text{A.2})$$

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

## A.2 Proof of Proposition 3.2.2

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

$$R_{01}(\text{dec}(\bar{f}; d)) \leq P\left(d(\bar{f}(x), T_y) \geq \frac{\gamma d}{2}\right) \leq \frac{2}{\gamma d} \mathbb{E}\left[d(\bar{f}(x), T_y)\right] = \frac{2}{\gamma d} R(\bar{f}; d) \quad (\text{A.3})$$

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

$$d(\bar{f}(x), T_{y'}) \geq d(T_y, T_{y'}) - d(T_y, \bar{f}(x)) \geq \frac{\gamma_d}{2} \quad (\text{A.4})$$

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

$$R(\bar{f}; L_1) = \mathbb{E}_{(x,y) \sim \mathcal{D}} \|\bar{f}(x) - T_y\|_1 \quad (\text{A.5})$$

$$\leq 2 \mathbb{E}_{(x,y) \sim \mathcal{D}} \sqrt{2\ell_{\text{KL}}(\bar{f}(x), T_y)} \quad (\text{A.6})$$

$$\leq 2 \sqrt{2 \mathbb{E}_{(x,y) \sim \mathcal{D}} \ell_{\text{KL}}(\bar{f}(x), T_y)} = 2\sqrt{2R(\bar{f}; \ell_{\text{KL}})} \quad (\text{A.7})$$

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 3.2.3

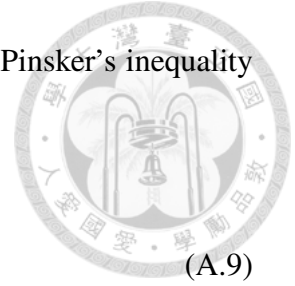
The decoding step remains the same when  $T' \neq T$  because the decoder uses the same transition matrix  $T$  to decode. The only difference is in the complementary probability estimates. Specifically, we have that the complementary estimation loss becomes  $R(\bar{f}; \ell) = \mathbb{E}_{(x,y) \sim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

Hence, the last equality in Equation (A.3) is no longer correct. Instead, we use the following:

$$\mathbb{E} \left[ d(\bar{f}(x), T_y) \right] \leq \mathbb{E} \left[ d(\bar{f}(x), T'_y) + d(T'_y, T_y) \right] \leq \mathbb{E} \left[ d(\bar{f}(x), T'_y) \right] + \epsilon \quad (\text{A.8})$$

to obtain that  $R_{01}(\text{dec}(\bar{f}; d)) \leq \frac{2}{\gamma d} R(\bar{f}; d) + \frac{2\epsilon}{\gamma d}$ . Then, we can use Pinsker's inequality and Jensen's inequality as in (A.5) to get

$$R_{01}(\text{dec}(f; L_1)) \leq \frac{4\sqrt{2}}{\gamma} \sqrt{R(\bar{f}; \ell)} + \frac{2\epsilon}{\gamma}. \quad (\text{A.9})$$





# Appendix 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  $f(\cdot; \theta)$  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:

$$L_{\text{Fwd}}(\theta) = \frac{1}{N} \sum_{i=1}^N -\log (T^\top f(x_i; \theta))_{\bar{y}_i} \quad (\text{B.10})$$

In the inference phase, Forward Correction predicts  $\hat{y} = \operatorname{argmax}_k f_k(x)$  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:  $\operatorname{argmax}_k ((T^\top)^{-1} \bar{f}(x; \theta))_k$ .

*Proof.* First, by setting the hypothesis set as above and plugging in the surrogate comple-

mentary estimation loss, we get the training objective function for CPE:

$$L_{\text{CPE}}(\theta) = \frac{1}{N} \sum_{i=1}^N -\log (T^\top f(x_i; \theta))_{\bar{y}_i} \quad (\text{B.11})$$



Equation (B.11) matches Equation (B.10), implying that in the training phase they select the same parameter  $\theta$ . Next, in the inference phase, it is clear that  $(T^\top)^{-1} \bar{f}(x; \theta) = (T^\top)^{-1} T^\top f(x; \theta) = f(x; \theta)$ , so both methods predict the same label for an instance  $x$ .  $\square$

Next, we further show that when  $T$  is the uniform transition matrix  $U$ , the decoder is equivalent to the  $L_1$  decoder, i.e.,  $\operatorname{argmax}_k ((U^\top)^{-1} \bar{f}(x))_k = \operatorname{argmin}_k \|U_k - \bar{f}(x)\|_1$ :

*Proof.* First, as

$$((U^\top)^{-1} \bar{f}(x))_k = -(K-1) \bar{f}_k(x) + \sum_{k=1}^K \bar{f}_k(x) = -(K-1) \bar{f}_k(x) + 1,$$

we have that  $\operatorname{argmax}_k ((U^\top)^{-1} \bar{f}(x))_k = \operatorname{argmin}_k \bar{f}_k(x)$ . Next, set  $\hat{y} = \operatorname{argmin}_k \bar{f}_k(x)$ . For any  $y \neq \hat{y}$ , we want to show

$$|U_{y\hat{y}} - \bar{f}_{\hat{y}}(x)| + |U_{yy} - \bar{f}_y(x)| \geq |U_{\hat{y}\hat{y}} - \bar{f}_{\hat{y}}(x)| + |U_{\hat{y}y} - \bar{f}_y(x)|. \quad (\text{B.12})$$

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

$$|U_{y\hat{y}} - \bar{f}_{\hat{y}}(x)| + |U_{yy} - \bar{f}_y(x)| = |U_{y\hat{y}} - \bar{f}_{\hat{y}}(x)| + \bar{f}_{\hat{y}}(x) + |U_{yy} - \bar{f}_y(x)| - \bar{f}_y(x) \quad (\text{B.13})$$

$$= |U_{\hat{y}\hat{y}} - \bar{f}_{\hat{y}}(x)| + |U_{y\hat{y}} - \bar{f}_{\hat{y}}(x)| + |U_{yy} - \bar{f}_y(x)| - \bar{f}_{\hat{y}}(x) \quad (\text{B.14})$$

$$= |U_{\hat{y}\hat{y}} - \bar{f}_{\hat{y}}(x)| + \frac{1}{K-1} - \bar{f}_{\hat{y}}(x) + \bar{f}_y(x) - \bar{f}_{\hat{y}}(x) \quad (\text{B.15})$$



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

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

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

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

Hence, Equation (B.12) holds. Now,

$$\sum_{k=1}^K |U_{yk} - \bar{f}_k(x)| = |U_{y\hat{y}} - \bar{f}_{\hat{y}}(x)| + |U_{yy} - \bar{f}_y(x)| + \sum_{k \neq y, \hat{y}} |U_{yk} - \bar{f}_k(x)| \quad (\text{B.16})$$

$$\geq |U_{\hat{y}y} - \bar{f}_y(x)| + |U_{\hat{y}\hat{y}} - \bar{f}_{\hat{y}}(x)| + \sum_{k \neq y, \hat{y}} |U_{\hat{y}k} - \bar{f}_k(x)| = \sum_{k=1}^K |U_{\hat{y}k} - \bar{f}_k(x)|$$

(B.17)

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

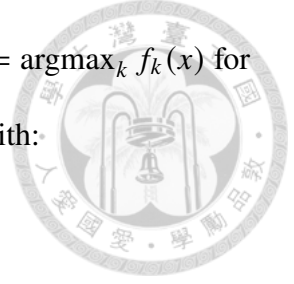
As the two decoders are equivalent, we have that Forward Correction is equivalent to CPE with

- Hypothesis Set:  $\{x \mapsto U^\top f(x; \theta) : \theta \in \Theta\}$
- Decoder:  $\operatorname{argmin}_k \|\bar{f}(x; \theta) - U_k\|_1$ .

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:

$$L_{\text{SCL}}(\theta) = \frac{1}{N} \sum_{i=1}^N -\log(1 - f(x_i; \theta))_{\bar{y}_i} \quad (\text{B.18})$$



In the inference phase, this method predicts the ordinary labels by  $\hat{y} = \operatorname{argmax}_k f_k(x)$  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:  $\operatorname{argmin}_k \|\bar{f}(x; \theta) - U_k\|_1$ .

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

$$L_{\text{CPE}}(\theta) = \frac{1}{N} \sum_{i=1}^N -\log(U^\top f(x_i; \theta)_{\bar{y}_i}) = \frac{1}{N} \sum_{i=1}^N -\log\left(\frac{1}{K-1} \sum_{k \neq \bar{y}_i} f_k(x_i; \theta)\right) \quad (\text{B.19})$$

$$= \frac{1}{N} \sum_{i=1}^N -\log(1 - f_{\bar{y}_i}(x_i; \theta)) + \log(K-1) = L_{\text{SCL}}(\theta) + \log(K-1) \quad (\text{B.20})$$

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  $\hat{y} = \operatorname{argmax}_k f(x; \theta)$  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:  $\hat{y} = \operatorname{argmin}_k \|\bar{f}(x) - U_k\|_1$ .  $\square$

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

$$L_{\text{DM}}(\theta) = \frac{1}{N} \sum_{i=1}^N -\log(\operatorname{sm}(1 - f(x_i; \theta)))_{\bar{y}_i} \quad (\text{B.21})$$

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

- Hypothesis Set:  $\{x \mapsto \operatorname{sm}(1 - f(x; \theta)) : \theta \in \Theta\}$

- Decoder:  $\operatorname{argmin}_k \|\bar{f}(x; \theta) - U_k\|_1$ .

*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

$$\bar{f}_k(x) = \frac{1}{Z} \exp(1 - f_k(x; \theta)) = \frac{e}{Z} \exp(-f_k(x; \theta)), \quad (\text{B.22})$$

where  $Z = \sum_{k=1}^K \exp(1 - f_k(x; \theta))$  is the normalization term. As  $x \mapsto \exp(-x)$  is monotonic decreasing, we have that  $\operatorname{argmin}_k \bar{f}_k(x; \theta) = \operatorname{argmax}_k f_k(x; \theta)$ . Next, as we have shown  $\operatorname{argmin}_k \bar{f}_k(x) = \operatorname{argmin}_k \|U_k - \bar{f}_k(x)\|_1$ , so  $\operatorname{argmax}_k f_k(x; \theta) = \operatorname{argmin}_k \|U_k - \bar{f}_k(x)\|_1$ , implying that both methods predict the same label for all instances.  $\square$

**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 4.1.
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].





# Appendix 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  $\{10, 20, \dots, 250\}$  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  $\{5, 10, \dots, 500\}$  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 C.1: 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	23.7± 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	<b>90.5± 0.2</b>	90.2± 0.2	25.0± 17.9	82.0± 0.4	79.6± 2.2	26.2± 8.7	59.9± 0.9	58.9± 0.7	16.4± 2.2
DM	89.7± 0.5	89.1± 0.2	22.7± 8.5	81.8± 0.3	78.2± 3.1	23.6± 5.5	<b>61.0± 1.5</b>	59.4± 1.4	17.7± 3.0
Fwd	<b>90.5± 0.2</b>	90.6± 0.4	91.6± 0.7	82.0± 0.4	81.6± 1.2	<b>83.4± 0.7</b>	59.9± 0.9	60.4± 0.9	62.6± 0.7
CPE-I	80.4± 0.3	73.5± 1.3	76.1± 1.6	74.6± 0.5	71.0± 1.5	74.7± 2.3	49.7± 0.6	42.8± 0.8	46.8± 1.4
CPE-F	<b>90.5± 0.2</b>	<b>90.7± 0.1</b>	<b>91.8± 0.4</b>	<b>82.2± 0.3</b>	<b>82.4± 0.4</b>	83.1± 1.0	60.4± 0.6	<b>60.8± 0.4</b>	62.8± 0.2
CPE-T	<b>90.5± 0.2</b>	90.6± 0.1	<b>91.8± 0.4</b>	82.0± 0.3	82.1± 0.5	83.2± 1.2	60.3± 0.5	60.6± 0.5	<b>63.0± 0.3</b>

Table C.2: Comparison of the testing classification accuracies with different levels of noise.

	MNIST			Fashion-MNIST			Kuzushiji-MNIST		
	$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.5$	$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.5$	$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.5$
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± 2.5	17.8± 3.2	17.4± 1.3
DM	23.3± 7.4	22.4± 8.7	23.4± 2.9	24.1± 7.1	24.3± 5.0	25.6± 3.9	18.1± 2.6	17.6± 2.4	16.5± 1.4
Fwd	91.1± 0.7	89.6± 1.0	82.5± 3.6	82.4± 0.9	81.4± 0.9	72.0± 7.5	<b>62.7± 1.0</b>	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	<b>61.1± 2.2</b>	<b>53.4± 1.5</b>
CPE-T	<b>91.3± 0.7</b>	<b>90.5± 0.8</b>	<b>85.7± 1.6</b>	<b>82.6± 1.3</b>	<b>81.6± 1.3</b>	<b>78.0± 1.6</b>	62.2± 0.8	<b>61.7± 1.7</b>	<b>55.0± 1.1</b>

**Benchmark results of linear models** Table C.1 and C.2 provide the 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.

**Training and validation loss curves** Figure C.1 and C.2 demonstrate the loss curve of the proposed CPE framework.

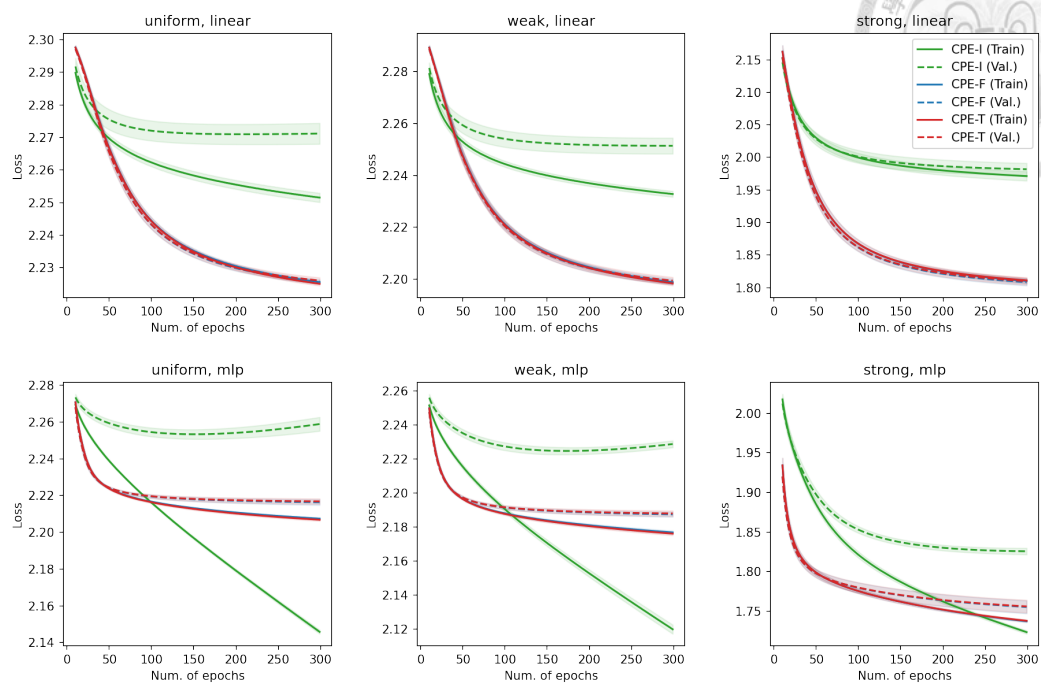
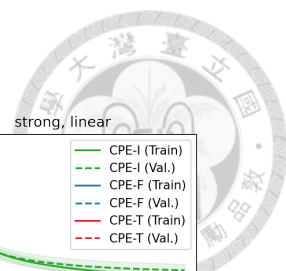


Figure C.1: Comparison of the training and validation loss of CPE with different transition layers in MNIST under different transition matrices. CPE-F and CPE-T perform almost identically, so the red lines and blue lines overlap in the figures. The shaded area denotes the standard deviation of five random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

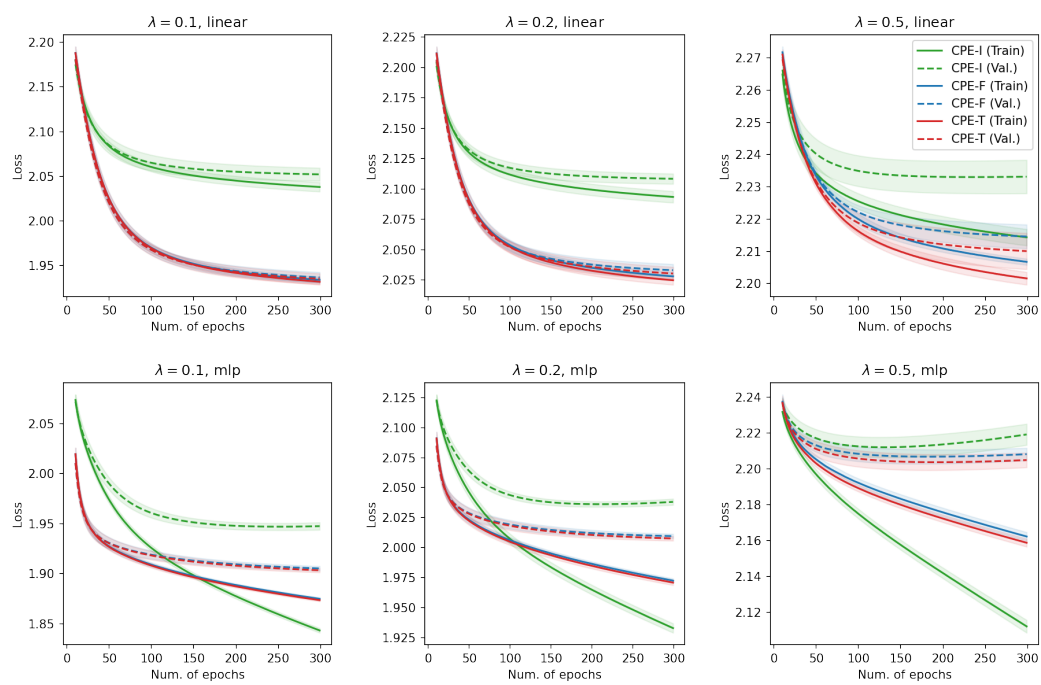


Figure C.2: Comparison of the training and validation loss of CPE with different transition layers in MNIST under different noise level. CPE-F and CPE-T perform almost identically when  $\lambda$  is small, so the red lines and blue lines overlap in those figures. The shaded area denotes the standard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



# Appendix D — Details of CLCIFAR20

## D.1 Label names of CLCIFAR20

Index	Full Label Name
0	aquatic mammals
1	fish
2	flowers
3	food containers
4	fruit, vegetables and mushrooms
5	household electrical devices
6	household furniture
7	insects
8	large carnivores and bear
9	large man-made outdoor things
10	large natural outdoor scenes
11	large omnivores and herbivores
12	medium-sized mammals
13	non-insect invertebrates
14	people
15	reptiles
16	small mammals
17	trees
18	transportation vehicles
19	non-transportation vehicles