A Theoretical Perspective on Streaming Noisy Data With Distribution Shift

Wenshui Luo, Shuo Chen, Tao Zhou, *Senior Member, IEEE*, and Chen Gong, *Senior Member, IEEE*

***Abstract*—Intelligent systems typically need to continually learn from streaming data subject to distribution shift, where a key requirement is that they cannot catastrophically forget the historical knowledge learned from previous data. More seriously, streaming data often contain substantial label noise, which can exacerbate catastrophic forgetting and lead to performance degradation on forthcoming data. To address these problems, Continual Noisy Label Learning (CNLL) has been proposed. However, existing CNLL methods still fall short of the ability in addressing catastrophic forgetting because they adopted heuristic strategies in handling label noise and did not explicitly characterize the distributional shift across time, which hinders effective knowledge transfer from historical data to new data. To tackle these challenges, we theoretically analyze the problem of learning from streaming noisy data with distribution shift and propose a unified framework called Continual Noisy Label Learning on Drifting Data Streams (CNLDD). Specifically, we theoretically explore, for the first time, the upper bound of cumulative generalization error for CNLL problem, which reveals three factors leading to forgetting, namely selection bias of buffered data, distribution shift, and label noise. To alleviate the selection bias of buffered data, we design a two-step buffer update strategy to narrow the distribution gap between the original historical data and the selected representative data in buffer. To address distribution shift, our CNLDD explicitly characterizes the distribution discrepancies between buffered data and incoming data, prioritizing historical data with minimal discrepancies to enhance knowledge transfer. To tackle noisy labels, CNLDD estimates the importance weight of each example with the instance-dependent noise transition matrix, thereby avoiding the data bias and knowledge forgetting arising from noisy labels. Empirically, due to the unified modeling of the aforementioned issues, our CNLDD achieves superior classification performance when compared with state-of-the-art CNLL methods on both synthetic and real-world datasets.**

Received 5 May 2025; revised 28 October 2025; accepted 5 December 2025. Date of publication 11 December 2025; date of current version 6 March 2026. This work was supported in part by the National Natural Science Fund of China under Grant 62336003, Grant 12371510, Grant 62576153, and Grant 62506155, in part by Shanghai Municipal Science and Technology Major Project under Grant 2025SHZDZX025G12, in part by the Provincial Natural Science Fund of Jiangsu under Grant BK20251985, and in part by Suzhou Municipal Leading Talents Fund under Grant 2025. Recommended for acceptance by P. Zhao. *(Corresponding author: Chen Gong.)*

Wenshui Luo and Chen Gong are with the School of Automation and Intelligent Sensing, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: randylo@sjtu.edu.cn; chen.gong@sjtu.edu.cn).

Shuo Chen is with the School of Intelligence Science and Technology, Nanjing University, Nanjing 210093, China, and also with the Center for Advanced Intelligence Project, RIKEN, Saitama 351-0198, Japan (e-mail: shuo.chen@nju.edu.cn).

Tao Zhou is with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210093, China (e-mail: taozhou.dreams@gmail.com).

This article has supplementary downloadable material available at https://doi.org/10.1109/TPAMI.2025.3643174, provided by the authors.

Digital Object Identifier 10.1109/TPAMI.2025.3643174

***Index Terms*—Continual learning, streaming data, distribution shift, label noise, generalization error.**

I. Introduction

LEARNING is foundational for intelligent systems to accommodate dynamic environments. To handle external changes, continual learners should have strong adaptability to continually acquire, update, accumulate, and utilize knowledge from streaming data with distribution shift [40], [41]. In this scenario, catastrophic forgetting of prior knowledge contained in historical data is a significant challenge [43]. More seriously, in many practical situations, the accurate labels of streaming data may be difficult to obtain due to various subjective or objective factors such as unavoidable human fatigue, limitation of human knowledge, unreliable automatic labeling process, etc [13]. The existence of noisy labels may exacerbate knowledge forgetting and degrade model performance on upcoming data [1], [20]. Therefore, **C**ontinual **N**oisy **L**abel **L**earning (CNLL) [3], [20], [21] has been proposed to address noisy labels in continual learning scenarios.

To the best of our knowledge, there are only a handful of existing works focusing on the CNLL problem, and they usually combine **C**ontinual **L**earning (CL) techniques with **L**abel **N**oise **L**earning (LNL) approaches [3], [20], [21] in a direct way, where clean sample selection is commonly employed to establish a memory buffer to store useful historical patterns, so that catastrophic forgetting can be avoided. For example, Self-Purified Replay (SPR) [21] demonstrates that noisy labels can accelerate forgetting and severely degrade performance on learned tasks. Therefore, to effectively maintain the purity of memory buffer, SPR introduces a self-supervised replay technique combined with a self-centered filter. In addition to prioritizing the purity of buffered data, Purity and Diversity aware Episode Replay (PuriDivER) [3] emphasizes the significance of data diversity. To this end, PuriDivER defines a scoring function to promote diversity by aligning the distribution of buffered data with that of the original noisy data. Moreover, inspired by DivideMix [27], Karim et al. [20] additionally adopt a noisy memory buffer and employ semi-supervised learning techniques to enhance the training of robust classifiers. In a word, existing CNLL methods usually focus on selecting clean representative data to reduce forgetting.

However, these CNLL methods often adopt heuristic data selection strategies to mitigate catastrophic forgetting, and they also did not consider the CNLL problem in a holistic view. In this paper, we theoretically identify that forgetting can be attributed

to three critical factors, namely selection bias of buffered data, distribution shift, and label noise. Among them, the distribution shift over time has also not been explicitly modeled by existing CNLL methods. Therefore, we simultaneously consider the three factors in a single framework, and formally refer to the problem studied in this paper as "learning from streaming noisy data with distribution shift".

Actually, distribution shift is ubiquitous in various applications of CNLL. For example, in the task of financial fraud detection [15], the distribution of transaction data may change over time due to the factors such as evolving fraud patterns, shifts in user behavior, different economic conditions, etc. Given the need for quick detection responses, the labels of most incoming transaction data are typically generated by historical models, inevitably introducing noisy labels. Therefore, the continual detector should adapt to the dynamic changes, retain previous knowledge on fraud, and also maintain robustness against noisy labels. Moreover, CNLL is also crucial in machine-aided medical diagnosis. A typical application is the tumor detection task [24], where tumor images may exhibit distribution shift due to the variations in imaging techniques, patient demographics, etc. Additionally, there is a continuous need for incremental learning as new tumor types, imaging techniques, or clinical knowledge emerge over time. Meanwhile, inexperienced experts may annotate noisy labels for medical images due to a lack of sufficient expertise. This highlights the importance of developing robust and adaptive tumor detectors capable of handling catastrophic forgetting caused by distribution shift and label noise.

To address the problem of learning from streaming noisy data with distribution shift, we theoretically explore, for the first time, the upper bound of the cumulative generalization error, which can be regarded as a sum of learning plasticity and memory stability [43]. Here, learning plasticity refers to the model ability to learn from new data, while memory stability pertains to its capacity to retain previously acquired knowledge. To achieve an optimal balance between plasticity and stability, we focus on exploring a minimizable upper bound for the cumulative generalization error and propose a new method termed "**C**ontinual **N**oisy **L**abel learning on **D**rifting **D**ata streams (CNLDD)".

Guided by the upper bound, we propose a new update strategy for the memory buffer aimed at minimizing the distribution gap between buffered data in memory and the data seen so far. To update the memory buffer, we select representative examples in a two-step manner, where at each step CNLDD selects a subset with a minimum covering radius, allowing a minimum selection bias of the buffered data. Additionally, we explicitly characterize the distribution discrepancy between buffered data and new incoming data, and prioritize historical data with small discrepancy to ensure effective knowledge transfer, which alleviates catastrophic forgetting arising from distribution shift. Finally, to address label noise, we employ the instance-dependent noise transition matrix to determine the importance weight of each example, thereby mitigating the negative impact of directly replaying noisy examples.

The main contributions of our paper can be highlighted in three folds:

- Theoretically, we are the first to study the upper bound of the cumulative generalization error for the problem of learning from streaming noisy data with distribution shift.
- Algorithmically, we propose CNLDD, a unified framework that simultaneously addresses the three critical factors leading to catastrophic forgetting, namely selection bias of buffered data, distribution shift, and label noise.
- Empirically, we conducted intensive experiments on synthetic and real-world datasets with distribution shift and label noise, which demonstrate the superiority of CNLDD over existing continual noisy label learning methods.

The rest of this paper is organized as follows. In Section II, we review related works on continual learning and label noise learning. In Section III, we introduce useful preliminary knowledge, which is followed by Section IV that presents a theoretical study on the generalization error. Subsequently, the CNLDD method is introduced thoroughly in Section V. After that, theoretical justifications of CNLDD are detailed in Section VI. Experimental results are provided in Section VII. Finally, we conclude our paper in Section VIII.

II. Related Work

Since the main focus of this paper is continual learning on streaming noisy data with distribution shift, here we briefly review some related works on continual learning and label noise learning.

A. Continual Learning

Existing continual learning methods can be roughly classified into three categories, namely regularization-based methods, optimization-based methods, and replay-based methods [12], [43].

Regularization-based methods are characterized by the addition of explicit regularization terms to balance old and new data, with weight regularization and function regularization as the two main branches. Here, weight regularization methods usually add a quadratic term to the loss function, penalizing the variation of each parameter based on its contribution or "importance" to the old task. Representative approaches include Elastic Weights Consolidation (EWC) [23], which leverages the Fisher Information Matrix (FIM) to characterize parametric importance, and Synaptic Intelligence (SI) [49], which approximates the parameter importance by its contribution to the loss variation over the entire training trajectory. In contrast, function regularization methods often treat the previously learned model as a teacher and the currently trained model as a student. These methods then use knowledge distillation [39] to maintain historical knowledge. For example, Learning without Forgetting (LwF) [43] encourages output consistency on new data between the fine-tuned model and the old model. Furthermore, to directly distill from historical data, Functional-Regularization of Memorable Past (FROMP) [34] employs buffered data to regularize the functional behavior of the model within a Bayesian framework.

Since the estimation of the importance matrix in regularization-based methods often incurs additional computational overhead, and they may face challenges

in modeling parametric importance for complex network architectures, their generalizability to complicated scenarios becomes limited. Consequently, the second line of research, namely optimization-based method, aims to address the forgetting problem via optimization techniques such as gradient projection and meta-learning [41]. Representative methods include Adam-NSCL [44], which projects gradients onto the null space of the feature covariance, and Layerwise Proximal Replay (LPR) [48], which constraints the gradient variation of hidden layers to ensure that the direction of the gradient update lies in the orthogonal complement of the subspace spanned by the gradients of historical data [11].

Due to the lack of access to historical data, the above two types of methods may struggle to effectively alleviate catastrophic forgetting, which frequently results in suboptimal performance. Therefore, the third line of research focuses on the explicit storage and replay of historical examples by using a small memory buffer. Typical types of replayed data in this direction include historical examples [5], [40], generated examples [37], and transformed examples [29]. Moreover, to further account for noisy labels, many methods intend to replay potentially clean examples. For example, Self-Purified Replay (SPR) [21] identifies that noisy labels can exacerbate catastrophic forgetting and significantly degrade performance on previously learned tasks. To ensure the purity of memory buffer, SPR combines a self-supervised replay mechanism with a self-centered filter, and selects clean examples via Beta Mixture Models [19]. Beyond emphasizing the purity of buffered data, Purity and Diversity aware Episode Replay (PuriDivER) [3] highlights the critical role of diversity. That is to say, PuriDivER proposes a scoring function to enhance the diversity of data in buffer by aligning the distribution of buffered data with the overall distribution of the original noisy data. Furthermore, inspired by DivideMix [27], Karim et al. [20] proposed Continual Noisy Label Learning (CNLL), which introduces a separate noisy memory buffer and leverages semi-supervised learning techniques to facilitate the training of robust classifiers.

However, these replay-based methods often rely on heuristic strategies to identify potential clean examples, which may result in a biased memory buffer and significantly mislead the continual learner. Furthermore, they are unable to explicitly characterize the distribution discrepancy between buffered data and new data, making it more challenging to transfer knowledge from historical data to new data. In light of these issues, we propose a unified framework that simultaneously addresses the critical factors contributing to forgetting, i.e., selection bias of buffered data, distribution shift, and label noise.

B. Label Noise Learning

Existing methods for handling label noise can be classified into three categories, namely sample selection-based methods, robust loss function design-based methods, and statistic estimation-based methods.

Sample selection-based methods focus on identifying clean examples or removing noisy examples from the original training dataset. Leveraging the memorization effect [1] of deep neural networks, these methods typically consider the examples with small loss values as clean ones during training. Representative methods in this category include Co-teaching [16] and CoDis [45]. However, a major limitation of most sample selection-based methods is their inability to theoretically guarantee the correctness of the labels for the selected examples, which undermines their stability and effectiveness in practical applications. As an alternative, a second strand of research emphasizes the development of robust loss functions to address noisy labels. Representative approaches include ϵ-Softmax [42] and Regularly Truncated M-Estimators (RTME) [46]. Nevertheless, the above two types of methods do not explicitly characterize the generation process of the label noise, so they inevitably become weak in some complicated noisy scenarios [8].

The third strand of research is to estimate some critical statistics of clean or noisy data. These methods can be further categorized based on the estimated statistics, such as the label noise transition matrix [28], the dataset centroid [13], and the mean/covariance of data [30]. However, the aforementioned label noise learning methods cannot be directly applied to tackle noisy labels in continual learning scenarios, as they often assume a fixed data distribution, which makes them less effective in scenarios with shifting distribution. Moreover, the absence of mechanisms to retain and integrate previously learned knowledge poses additional challenges for continual learning.

III. Preliminaries

In this section, we present the detailed definition and the related mathematical notations for our setting, i.e., learning from streaming noisy data with distribution shift. Additionally, we provide a brief introduction to the noise transition matrix estimation utilized by our CNLDD method.

A. Problem Definition

We first introduce some mathematical notations which will be used in this paper. Specifically, the superscript "$\tilde{}$" indicates that the variable is calculated based on noisy observations, and the variable with a superscript "$\hat{}$" is the corresponding empirical estimation. Note that a statistic value accompanied by the term "clean" means that this statistic is calculated by using underlying clean labels, whereas a statistic accompanied by the term "noisy" implies that it is calculated by using observed noisy labels. We use the notation $[\![K]\!]$ to represent the set $\{1, 2, \ldots, K\}$ for any $K \in \mathbb{Z}$. Besides, the one-hot vector with a value of 1in its j-th element is denoted by $\mathbf{e}_j$. The mathematical expectation is denoted by $\mathbb{E}[\cdot]$. Moreover, we use $\mathbf{A} = (A_{ij})_{1\le i\le m}^{1\le j\le n} \in \mathbb{R}^{m\times n}$ to denote an $m \times n$ matrix $\mathbf{A}$ of which the (i, j)-th element is A_{ij}. We also use "$\bigotimes$" to represent the Cartesian product of distributions. For clarity, the main mathematical notations that will be later used for algorithm description are listed in Table I.

We now formally define the setting considered in this paper. Specifically, we focus on a C-way classification task on streaming noisy data with time-varying distribution shift. Let $X_t \in \mathcal{X} \subseteq \mathbb{R}^d$ and $Y_t \in \mathcal{Y} = \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_C\}$ denote the random variables of the feature and label at the t-th timestep, respectively. The joint probability distribution for (X_t, Y_t) and

TABLE I
SUMMARY OF MAIN MATHEMATICAL NOTATIONS

Notation	Interpretation
$(X_t, Y_t),\ (X_t, \widetilde{Y}_t)$	A pair of input random variables and the observed contaminated counterpart at the t-th timestep. Here $X_t \in \mathcal{X} \subseteq \mathbb{R}^d$ represents the feature and $Y_t \in \mathcal{Y} = \{0,1\}^C$ represents the one-hot label, where d is the dimension of the feature space and C denotes the number of classes.
$\mathbb{D}_t,\ \widetilde{\mathbb{D}}_t$	The joint probability distribution of X_t and Y_t, and its noisy counterpart, respectively. Their density functions are $P_t(X_t, Y_t)$ and $\widetilde{P}_t(X_t, \widetilde{Y}_t)$, respectively.
$\mathcal{S}_t = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{n_t}$	The unobservable clean sample of $\mathbb{D}_t$ with n_t data points.
$\widetilde{\mathcal{S}}_t = \{(\mathbf{x}_i, \widetilde{\mathbf{y}}_i)\}_{i=1}^{n_t}$	The observed noisy sample of $\widetilde{\mathbb{D}}_t$ with n_t possibly mislabeled training data.
$[\![C]\!]$	The shorthand for the set $\{1, 2, \cdots, C\}$.
$\widetilde{\mathcal{M}}_{1:t}$	The memory buffer up to the t-th timestep.
$\beta_{\mathbb{P}}(\cdot,\cdot)$	The density ratio function with respect to the distribution $\mathbb{P}$, defined as $\beta_{\mathbb{P}}(\mathbf{x}, \widetilde{\mathbf{y}}) := P(\mathbf{x}, \widetilde{\mathbf{y}}) / \widetilde{P}(\mathbf{x}, \widetilde{\mathbf{y}}) = P(\widetilde{\mathbf{y}} \mid \mathbf{x}) / \widetilde{P}(\widetilde{\mathbf{y}} \mid \mathbf{x})$.
$\overline{L},\ \overline{\beta}$	The upper bounds of the loss function and the density ratio function, respectively.
$\lambda^P,\ \lambda^\ell$	The Lipschitz constants of $P_t(Y_t \mid X_t)$ and the loss function $\ell(f_{\boldsymbol{\theta}}(\cdot), \cdot)$, respectively.
$\|\mathbf{x}\|_p$	The ℓ_p-norm of a given vector $\mathbf{x}$, with $p \in \{1, 2, \infty\}$.
$\|\mathbf{A}\|_p,\ \|\mathbf{A}\|_{\mathrm{F}}$	The matrix norms induced by the corresponding vector norms $\|\cdot\|_p$, and by the Frobenius norm.

its noisy counterpart are denoted by $\mathbb{D}_t$ and $\widetilde{\mathbb{D}}_t$, respectively, with the corresponding density functions being $P_t(X_t, Y_t)$ and $\widetilde{P}_t(X_t, \widetilde{Y}_t)$. In the following, we use $\mathbb{D}_{1:t}$ to denote the cumulative distribution up to timestep t, and any quantity with a subscript $1:t$ indicates a cumulative quantity. At timestep t, the clean sample set $\mathcal{S}_t = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{n_t}$ contains n_t independent and identically distributed (i.i.d.) examples from $\mathbb{D}_t$. However, due to the existence of noisy labels, we are only accessible to a sample $\widetilde{\mathcal{S}}_t = \{(\mathbf{x}_i, \widetilde{\mathbf{y}}_i)\}_{i=1}^{n_t}$ from a noisy distribution $\widetilde{\mathbb{D}}_t$. In line with existing CNLL methods [3], [20], [21], we also assume the existence of a small memory buffer to store historical data. At timestep t, the updated memory buffer with capacity M is denoted by $\widetilde{\mathcal{M}}_{1:t}$, while $\mathcal{M}_{1:t}$ represents the buffer with all labels replaced by the corresponding clean ones.

In this paper, we consider the instance-dependent label noise [6], which closely aligns with the real-world label noise. In this setting, the observed noisy label for each $\mathbf{x} \in \mathcal{X}$ depends not only on its underlying clean label, but also on the feature itself. To be more specific, for any $\mathbf{x} \in \mathcal{X}$, the transition probability of class i to class j is given by $P(\widetilde{Y} = \mathbf{e}_j | Y = \mathbf{e}_i, X = \mathbf{x}) = T_{ij}(\mathbf{x}), \forall i, j \in [\![C]\!]$, where $\mathbf{T}(\mathbf{x}) := (T_{ij}(\mathbf{x}))_{1 \le i,j \le C} \in \mathbb{R}^{C \times C}$ is the noise transition matrix for $\mathbf{x}$, and C is the number of classes.

B. Estimation of Noise Transition Matrix for Instance-Dependent Label Noise

In this section, we briefly introduce the High-Order Consensuses (HOC) [52] algorithm, which can be adopted to estimate the instance-dependent transition matrix $\mathbf{T}(\mathbf{x})$ for any example $\mathbf{x}$ in the training set.

The key observation of HOC is that noisy data can still induce good representations, even though label noise makes the model generalize poorly [26]. Based on this, HOC assumes the "2-NN label clusterability", i.e., for any example $\mathbf{x}$ in a dataset $\mathcal{D}$, its two nearest neighbors belong to the same class as $\mathbf{x}$. Under this condition, the joint probability distributions of noisy labels for one, two, and three adjacent examples can be modeled accordingly. The transition matrix and class prior can then be jointly estimated by using up to the third-order consensus of the label distribution. In the following, we first present the estimation procedure for class-dependent label noise, which means that the generation of label noise is independent to feature $\mathbf{x}$, so $\mathbf{T}(\mathbf{x})$ degenerates to a fixed matrix $\mathbf{T}$ for any input $\mathbf{x}$. Subsequently, we introduce the extension to the instance-dependent case with noise transition matrix strictly being $\mathbf{T}(\mathbf{x})$.

For the class-dependent case, let X and Y denote the random variables representing the feature and the label, respectively. We define the class prior distribution as $\mathbf{p} := [P(Y = \mathbf{e}_1), P(Y = \mathbf{e}_2), \ldots, P(Y = \mathbf{e}_C)]^\top$. To facilitate the derivation of three orders of consensus, we define the transition matrices with column permutation as:

$$\mathbf{T}_r := \mathbf{T}\mathbf{S}_r, \quad \forall r \in [\![C]\!], \tag{1}$$

where $\mathbf{S}_r := [\mathbf{e}_{r+1}, \mathbf{e}_{r+2}, \ldots, \mathbf{e}_C, \mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_r]$ is a permutation matrix. Let $(i + r)_C := [(i + r - 1) \bmod C] + 1$, and the first-, second-, and third-order consensuses of noisy labels can then be denoted in vector forms, namely,

$$\begin{aligned}
\mathbf{c}^{[1]} &= [P(\widetilde{Y} = \mathbf{e}_i), i \in [\![C]\!]]^\top, \\
\mathbf{c}_r^{[2]} &= [P(\widetilde{Y} = \mathbf{e}_i, \widetilde{Y}_1 = \mathbf{e}_{(i+r)_C}), i \in [\![C]\!]]^\top, \\
\mathbf{c}_{r,s}^{[3]} &= [P(\widetilde{Y} = \mathbf{e}_i, \widetilde{Y}_1 = \mathbf{e}_{(i+r)_C}, \widetilde{Y}_2 = \mathbf{e}_{(i+s)_C}), i \in [\![C]\!]]^\top,
\end{aligned} \tag{2}$$

where $\widetilde{Y}_1$ and $\widetilde{Y}_2$ are the random variables corresponding to the noisy labels of the two nearest neighbors, respectively. To empirically estimate quantities in (2), a subset of the noisy dataset is sampled, and the corresponding estimated values are denoted by $\widehat{\mathbf{c}}^{[1]}$, $\widehat{\mathbf{c}}_r^{[2]}$, and $\widehat{\mathbf{c}}_{r,s}^{[3]}$, respectively.

Furthermore, the three orders of consensuses can also be formally defined as functions of $(\mathbf{T}, \mathbf{p})$, respectively, namely,

$$\begin{aligned}
\mathbf{c}^{[1]}(\mathbf{T}, \mathbf{p}) &:= \mathbf{T}^\top \mathbf{p}, \\
\mathbf{c}_r^{[2]}(\mathbf{T}, \mathbf{p}) &:= (\mathbf{T} \circ \mathbf{T}_r)^\top \mathbf{p}, \ \forall r \in [\![C]\!], \\
\mathbf{c}_{r,s}^{[3]}(\mathbf{T}, \mathbf{p}) &:= (\mathbf{T} \circ \mathbf{T}_r \circ \mathbf{T}_s)^\top \mathbf{p}, \ \forall r, s \in [\![C]\!],
\end{aligned} \tag{3}$$

where $\circ$ denotes the element-wise matrix product. For brevity, we compactly define $\mathbf{c}^{[2]}(\mathbf{T}, \mathbf{p}) := [\mathbf{c}_r^{[2]}(\mathbf{T}, \mathbf{p}), \forall r \in [\![C]\!]]$ and $\mathbf{c}^{[3]}(\mathbf{T}, \mathbf{p}) := [\mathbf{c}_{r,s}^{[3]}(\mathbf{T}, \mathbf{p}), \forall r, s \in [\![C]\!]]$. The estimated quantities $\widehat{\mathbf{c}}^{[2]}$ and $\widehat{\mathbf{c}}^{[3]}$ are defined analogously to $\mathbf{c}^{[2]}(\mathbf{T}, \mathbf{p})$ and $\mathbf{c}^{[3]}(\mathbf{T}, \mathbf{p})$. Based on these quantities, the noise transition matrix $\mathbf{T}$ and the class prior $\mathbf{p}$ in (3) can then be estimated by leveraging the consensus between the expected values $\mathbf{c}^{[z]}(\mathbf{T}, \mathbf{p})$ and their

empirical counterparts $\widehat{\mathbf{c}}^{[z]}$ for all $z \in \{1,2,3\}$. More specifically, estimating $\mathbf{T} = (T_{ij})_{1\le i,j\le C}$ and $\mathbf{p} = (p_i)_{1\le i\le C}$ can be formulated as the following constrained optimization problem:

$$\min_{\mathbf{T},\mathbf{p}} \sum_{z=1}^{3} \left\| \widehat{\mathbf{c}}^{[z]} - \mathbf{c}^{[z]}(\mathbf{T},\mathbf{p}) \right\|_2^2$$
$$\text{s.t.} \quad p_i \ge 0,\ T_{ij} \ge 0,\ \forall\, i,j \in [\![C]\!]$$
$$\sum_{i=1}^{C} p_i = 1,\ \sum_{j=1}^{C} T_{ij} = 1,\ \forall\, i \in [\![C]\!]. \tag{4}$$

Building upon the estimation method for the class-dependent noise transition matrix $\mathbf{T}$ in (4), HOC extends this framework to address the instance-dependent case. To this end, HOC assumes that the entire dataset can be partitioned into U disjoint subsets, with each subset characterized by a shared noise transition matrix. Then, to estimate the shared matrix for a single subset, HOC algorithm developed for the class-dependent scenario is employed. Formally, let $q(\mathbf{x})$ denote the index of the subset to which feature $\mathbf{x}$ belongs, and the estimated transition matrices for the U subsets are $\{\widehat{\mathbf{T}}^u\}_{u=1}^U$. Based on these quantities, the transition matrix for $\mathbf{x}$ is expressed as $\mathbf{T}(\mathbf{x}) = \widehat{\mathbf{T}}^{q(\mathbf{x})}$.

IV. Theoretical Study

In this section, we study the problem of learning from streaming noisy data with distribution shift from a theoretical perspective and present a comprehensive evaluation on the cumulative generalization error in our setting. Due to space limitations, all proofs of theorems are deferred to the supplementary materials.

A. A General Upper Bound for Cumulative Generalization Error

According to the aforementioned problem definition, a small memory buffer $\widetilde{\mathcal{M}}_{1:t}$ is accessible at timestep t. Representative CNLL methods often update this buffer with the consideration of purity [21] or diversity [3]. However, these heuristic strategies cannot be directly applied to balance memory stability and learning plasticity [43]. Moreover, the updated buffer may still be biased, primarily due to the inclusion of noisy labels, which can result in toxic replay [20]. In view of this, we directly study the cumulative generalization error defined on the data distributions seen so far, namely $\mathbb{E}_{\mathbb{D}_{1:t}}[\ell(f_{\boldsymbol{\theta}}(\mathbf{x}),\mathbf{y})]$, where $\ell(\cdot,\cdot)$ is a loss function, and $f_{\boldsymbol{\theta}} \in \mathcal{F}_\Theta$ represents a certain hypothesis in the hypothesis space $\mathcal{F}_\Theta = \{f_{\boldsymbol{\theta}} : \mathcal{X} \to \Delta^K, \boldsymbol{\theta} \in \Theta\}$. Here, Δ^C is a C-dimensional probability simplex, and Θ is the parameter space. Since the generalization error is difficult to estimate in a direct way with data in memory buffer, an optimizable upper bound for this error naturally serves as a practical alternative. As shown later, the cumulative generalization error can be expressed as a convex combination of two components, representing memory stability and learning plasticity, respectively. Therefore, minimizing the upper bound for this error is directly related to the performance of a continual learner.

Before formally deriving the upper bound of the cumulative generalization error, we first need to define two key concepts, namely distribution discrepancy (Definition 1) and density ratio (Definition 2):

Definition 1. (Distribution Discrepancy): For a loss function $\ell(\cdot,\cdot)$ and two distributions $\mathbb{P}$ and $\mathbb{Q}$, the distribution discrepancy between $\mathbb{P}$ and $\mathbb{Q}$ is defined as: $disc_{\mathcal{F}_\Theta}(\mathbb{P},\mathbb{Q}) := \sup_{f_{\boldsymbol{\theta}}\in\mathcal{F}_\Theta} |\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathbb{P}}[\ell(f_{\boldsymbol{\theta}}(\mathbf{x}),\mathbf{y})] - \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathbb{Q}}[\ell(f_{\boldsymbol{\theta}}(\mathbf{x}),\mathbf{y})]|$.

Definition 2. (Density Ratio): For a distribution $\mathbb{P}$ and its corresponding noisy distribution $\widetilde{\mathbb{P}}$, assuming their probability density functions are $P(\cdot,\cdot)$ and $\widetilde{P}(\cdot,\cdot)$, respectively, the density ratio between the posterior probability distributions $\mathbb{P}$ and $\widetilde{\mathbb{P}}$ is defined as: $\beta_{\mathbb{P}}(\mathbf{x},\widetilde{\mathbf{y}}) := \frac{P(\widetilde{\mathbf{y}}|\mathbf{x})}{\widetilde{P}(\widetilde{\mathbf{y}}|\mathbf{x})},\ \forall\,(\mathbf{x},\widetilde{\mathbf{y}}) \in \text{supp}(\widetilde{\mathbb{P}})$.

The two definitions are useful for strictly characterizing the discrepancies between data distributions. In the following, unless otherwise stated, we use $\mathbb{E}_{\mathbb{Q}}[\cdot]$ to denote the expectation over $(\mathbf{x},\mathbf{y})\sim\mathbb{Q}$, i.e., $\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathbb{Q}}[\cdot]$. Moreover, to decompose the generalization error over the first t timesteps, we make the Assumption 1, wherein it is assumed that the cumulative distribution $\mathbb{D}_{1:t}$ can be expressed as a weighted sum of $\mathbb{D}_{1:t-1}$ and $\mathbb{D}_t$ with weight α_t.

Assumption 1. (Mixture of Distributions): Assume that for any t, there exists $0 < \alpha_t < 1$, such that $\mathbb{D}_{1:t} = (1-\alpha_t)\mathbb{D}_{1:t-1} + \alpha_t\mathbb{D}_t$. For example, if $\mathbb{D}_{1:t} = \frac{1}{t}\sum_{i=1}^{t}\mathbb{D}_i$, then $\alpha_t = \frac{1}{t}$.

Assumption 1 is widely adopted in machine learning research [31]. By the factorization in this assumption, it can be observed that the cumulative error consists of two factors associated with memory stability and learning plasticity [43]. The former pertains to the model capacity to retain acquired knowledge (namely $\mathbb{D}_{1:t-1}$), while the latter refers to its ability to learn from new data (namely $\mathbb{D}_t$). Additionally, we follow [28], [35] and assume the existence of upper bounds for the adopted loss function as well as the density ratio functions, as described in Assumption 2:

Assumption 2: (**Boundedness of** $\ell(\cdot,\cdot)$ **and** $\beta_{\mathbb{P}}(\cdot,\cdot)$) Assume that the loss function and the density ratio function are respectively bounded above by $\overline{L} > 0$ and $\overline{\beta} > 0$, namely $\overline{L} = \sup_{f_{\boldsymbol{\theta}}\in\mathcal{F}_\Theta,\mathbf{x}\in\mathcal{X},1\le i\le C}\ell(f_{\boldsymbol{\theta}}(\mathbf{x}),\mathbf{e}_i)$ and $\overline{\beta} = \sup_{(\mathbf{x},\widetilde{\mathbf{y}})\in\text{supp}(\widetilde{\mathbb{P}})}\beta_{\mathbb{P}}(\mathbf{x},\widetilde{\mathbf{y}})$ for all measurable distributions $\mathbb{P}$.

This assumption will be leveraged to establish the upper bound of the cumulative generalization error. To justify these assumptions, we present detailed analyses in Section A of the supplementary material.

Based on the definitions and assumptions above, we proceed to analyze the cumulative generalization error over the first t timesteps. The empirical distribution for examples in $\mathcal{M}_{1:t-1}$, i.e., the buffer containing examples with clean but unobservable labels, is given by

$$\mathbb{P}^{\mathcal{M}}_{1:t-1} = \frac{1}{|\mathcal{M}_{1:t-1}|}\sum_{\mathbf{z}=(\mathbf{x},\mathbf{y})\in\mathcal{M}_{1:t-1}}\delta(\mathbf{z}), \tag{5}$$

where $\delta(\cdot)$ denotes the Dirac delta function. Similarly, the data distribution corresponding to the noisy buffer is $\widetilde{\mathbb{P}}^{\mathcal{M}}_{1:t-1}$. By considering the approximation error between buffered data distribution and cumulative distribution $\mathbb{D}_{1:t-1}$, we provide in Theorem 1 a preliminary upper bound for the cumulative generalization error.

Theorem 1: (**Preliminary Sketch of Cumulative Generalization Error Bound**) Based on Assumption 1, we define the density ratio functions corresponding to buffered data distribution and clean data distribution at timestep t as $\beta_1 = \beta_{\mathbb{P}^{\mathcal{M}}_{1:t-1}}$ and $\beta_2 = \beta_{\mathbb{D}_t}$, respectively. Then, the generalization error is upper-bounded as:

$$\mathbb{E}_{\mathbb{D}_{1:t}}[\ell(f_{\boldsymbol{\theta}}(\mathbf{x}), \mathbf{y})] \leq \underbrace{(1-\alpha_t)\mathbb{E}_{\widetilde{\mathbb{P}}^{\mathcal{M}}_{1:t-1}}[\beta_1(\mathbf{x}, \widetilde{\mathbf{y}})\ell(f_{\boldsymbol{\theta}}(\mathbf{x}), \widetilde{\mathbf{y}})]}_{\text{Term 1}}$$
$$+\underbrace{(1-\alpha_t)Gap(\mathbb{P}^{\mathcal{M}}_{1:t-1}, \mathbb{D}_{1:t-1})}_{\text{Term 2}} + \underbrace{\alpha_t \mathbb{E}_{\widetilde{\mathbb{D}}_t}[\beta_2(\mathbf{x}, \widetilde{\mathbf{y}})\ell(f_{\boldsymbol{\theta}}(\mathbf{x}), \widetilde{\mathbf{y}})]}_{\text{Term 3}}, \tag{6}$$

where Term 2 contains the distribution gap between $\mathbb{P}^{\mathcal{M}}_{1:t-1}$ and $\mathbb{D}_{1:t-1}$, which is defined as

$$Gap(\mathbb{P}^{\mathcal{M}}_{1:t-1}, \mathbb{D}_{1:t-1}) := \left| \mathbb{E}_{\mathbb{P}^{\mathcal{M}}_{1:t-1}}[\ell(f_{\boldsymbol{\theta}}(\mathbf{x}), \mathbf{y})] - \mathbb{E}_{\mathbb{D}_{1:t-1}}[\ell(f_{\boldsymbol{\theta}}(\mathbf{x}), \mathbf{y})] \right|. \tag{7}$$

From Theorem 1, we observe that for any timestep t, the upper bound of generalization error can be decomposed into three parts. Specifically, Term 1 at the right-hand side of (6) represents the empirical risk defined on the buffered data, Term 2 characterizes the gap between the distribution of buffered data and the distribution of clean cumulative data, and Term 3 accounts for the generalization error defined on the data distribution $\mathbb{D}_t$. Here, Term 1in the upper bound can be estimated by using a finite number of noisy examples in the buffer. However, Term 2, namely the selection bias of buffered data, cannot be minimized directly, because it lacks a concrete measure that quantifies the closeness between $\mathbb{P}^{\mathcal{M}}_{1:t-1}$ and $\mathbb{D}_{1:t-1}$. We will show later in Section V-A a theoretically solid method to minimize this term. Moreover, the bound in Theorem 1 is coarse-grained, because it cannot incorporate the distribution shift across time. Therefore, in the following section, we particularly consider such shift and propose to transfer knowledge based on distribution discrepancy.

B. Discrepancy-Based Knowledge Transfer

In this section, we thoroughly analyze Term 3 in Theorem 1, where distribution discrepancy-based knowledge transfer between buffered data and new data is considered.

Since the examples in $\widetilde{\mathcal{M}}_{1:t}$ are drawn from various distributions that may differ significantly from $\mathbb{D}_t$ at timestep t, directly optimizing both the first and third terms in the error upper bound of (6) poses substantial optimization challenges. Moreover, the examples in the memory buffer may be beneficial for the improvement of learning plasticity [43], i.e., the ability to learn from $\mathbb{D}_t$. However, the aforementioned generalization error bound does not explicitly capture the process of knowledge transfer from buffered data to new data at timestep t.

Inspired by the theory of batch distribution drift [2], we derive an upper bound for $\mathbb{E}_{\widetilde{\mathbb{D}}_t}[\beta_2(\mathbf{x}, \widetilde{\mathbf{y}})\ell(f_{\boldsymbol{\theta}}(\mathbf{x}), \widetilde{\mathbf{y}})]$, which incorporates knowledge transfer across distinct distributions. To facilitate this discussion, we first define the weighted Rademacher complexity, which quantifies the expressiveness of a given hypothesis space [32], [36]. Let $\widetilde{\mathcal{S}}$ denote a dataset of m examples drawn from the distribution $\widetilde{\mathbb{P}}$. The weighted Rademacher complexity of the hypothesis space $\mathcal{F}_\Theta$ is then defined as:

$$\mathfrak{R}_{\mathbf{q}}(\beta_{\mathbb{P}} \circ \ell \circ \mathcal{F}_\Theta) := \mathbb{E}_{\widetilde{\mathcal{S}} \sim \widetilde{\mathbb{P}}} \mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_{f_{\boldsymbol{\theta}} \in \mathcal{F}_\Theta} \sum_{i=1}^{m} \sigma_i q_i \beta_{\mathbb{P}}(\mathbf{x}_i, \widetilde{\mathbf{y}}_i) \ell(f_{\boldsymbol{\theta}}(\mathbf{x}_i), \widetilde{\mathbf{y}}_i) \right]. \tag{8}$$

Here, we use $\boldsymbol{\sigma} = (\sigma_i)_{1 \leq i \leq m} \in \{-1, +1\}^m$ to represent a Rademacher random vector, where $P(\sigma_i = +1) = P(\sigma_i = -1) = \frac{1}{2}$, $\forall\, i \in [\![m]\!]$. The density ratio function $\beta_{\mathbb{P}}(\cdot, \cdot)$ is defined in Definition 2. Additionally, the vector $\mathbf{q} = (q_i)_{1 \leq i \leq m}$ satisfies $0 < q_i < 1$, $\forall\, i \in [\![m]\!]$ and $\|\mathbf{q}\|_1 = 1$.

Since buffered data are sampled from distinct distributions (i.e., $\widetilde{\mathbb{D}}_1$ to $\widetilde{\mathbb{D}}_{t-1}$), they may not be equally useful for the learning of $\widetilde{\mathbb{D}}_t$. However, historical examples similar to new data should hold greater importance in facilitating the learning of the latest model. Therefore, we can cluster buffered data into K clusters, namely $\{\mathcal{M}^{(k)}_{1:t-1}\}_{k=1}^K$ and dynamically determine importance values according to distribution discrepancies. To this end, we incorporate the discrepancies into the upper bound of $\mathbb{E}_{\widetilde{\mathbb{D}}_t}[\beta_2(\mathbf{x}, \widetilde{\mathbf{y}})\ell(f_{\boldsymbol{\theta}}(\mathbf{x}), \widetilde{\mathbf{y}})]$. Let $\mathbb{P}_k$ denote the empirical distribution on $\mathcal{M}^{(k)}_{1:t-1}$ for all $k \in [\![K]\!]$, with m_k denoting the number of examples in the k-th cluster. The noisy counterparts of the K clusters and the empirical distributions are given by $\{\widetilde{\mathcal{M}}^{(k)}_{1:t-1}\}_{k=1}^K$ and $\{\widetilde{\mathbb{P}}_k\}_{k=1}^K$, respectively. The empirical distribution of $\widetilde{\mathcal{S}}_t$ is denoted by $\widetilde{\mathbb{P}}_{\mathcal{S}_t}$. Based on the aforementioned definitions, Theorem 2 presents an upper bound for Term 3in Theorem 1, which is:

Theorem 2: Under Assumption 2, when the dataset $\{\widetilde{\mathcal{M}}^{(k)}_{1:t-1}\}_{k=1}^K \cup \widetilde{\mathcal{S}}_t$ is sampled from $\widetilde{\mathbb{P}} = \widetilde{\mathbb{P}}_1^{m_1} \otimes \widetilde{\mathbb{P}}_2^{m_2} \otimes \cdots \otimes \widetilde{\mathbb{P}}_K^{m_K} \otimes \widetilde{\mathbb{D}}_t^{n_t}$ with $\mathbb{P}$ denoting the corresponding clean distribution, for any $f_{\boldsymbol{\theta}} \in \mathcal{F}_\Theta$, with probability at least $1-\delta$, it holds that:

$$\begin{aligned}
&\mathbb{E}_{\widetilde{\mathbb{D}}_t}[\beta_{\mathbb{D}_t}(\mathbf{x}, \widetilde{\mathbf{y}})\ell(f_{\boldsymbol{\theta}}(\mathbf{x}), \widetilde{\mathbf{y}})] \\
&\leq \sum_{(\mathbf{x}_i, \widetilde{\mathbf{y}}_i) \in \widetilde{\mathcal{M}}_{1:t-1} \cup \widetilde{\mathcal{S}}_t} q_i \cdot \beta_{\mathbb{P}}(\mathbf{x}_i, \widetilde{\mathbf{y}}_i)\ell(f_{\boldsymbol{\theta}}(\mathbf{x}_i), \widetilde{\mathbf{y}}_i) \\
&\quad + \sum_{k=1}^{K} \overline{q}_k \cdot disc_{\mathcal{F}_\Theta}(\mathbb{P}_k, \mathbb{P}_{\mathcal{S}_t}) + \mathcal{O}\left(\frac{1}{\sqrt{|\widetilde{\mathcal{S}}_t|}}\right) \\
&\quad + 2\mathfrak{R}_{\mathbf{q}}(\beta_{\mathbb{P}} \circ \ell \circ \mathcal{F}_\Theta) + \|\mathbf{q}\|_2 \overline{L} \sqrt{\frac{\log(2/\delta)}{2}},
\end{aligned} \tag{9}$$

where $\mathbf{q} = (q_i)_{i=1}^{|\widetilde{\mathcal{M}}_{1:t-1}| + |\widetilde{\mathcal{S}}_t|}$ satisfies $q_i > 0$ and $\|\mathbf{q}\|_1 = 1$. Moreover, $\overline{q}_k = \sum_{(\mathbf{x}_i, \widetilde{\mathbf{y}}_i) \in \widetilde{\mathcal{M}}^{(k)}_{1:t-1}} q_i$ represents the sum of the weights assigned to all the examples in the k-th cluster $\widetilde{\mathcal{M}}^{(k)}_{1:t-1}$.

Theorem 2 indicates that the generalization error on the data distribution $\mathbb{D}_t$ is bounded above by the weighted loss computed over both buffered data and newly observed data. For a specific subset $\widetilde{\mathcal{M}}^{(k)}_{1:t-1}$ of the memory buffer, if its distribution substantially deviates from that of the current data $\widetilde{\mathcal{S}}_t$, the corresponding distribution discrepancy term $disc_{\mathcal{F}_\Theta}(\mathbb{P}_k, \mathbb{P}_{\mathcal{S}_t})$ becomes large.

Consequently, the weight sum $\overline{q}_k$ of this subset should be reduced to maintain a tight upper bound on the generalization error. Since uniform weights can be assigned to new data and zero weights to buffered data, it is evident that this bound is no worse than the empirical error bound computed solely on $\widetilde{\mathcal{S}}_t$.

It is observed that the distribution discrepancy term $disc_{\mathcal{F}_\Theta}(\mathbb{P}_k, \mathbb{P}_{\mathcal{S}_t})$ in Theorem 2 depends on the clean data $\mathcal{M}_{1:t-1}^{(k)}$ and $\mathcal{S}_t$, which are inaccessible in our setting. Therefore, we can further estimate this term by using the noisy counterparts, namely $\widetilde{\mathcal{M}}_{1:t-1}^{(k)}$ and $\widetilde{\mathcal{S}}_t$. For clarity, the estimation algorithm is derived for two distributions $\mathbb{P}$ and $\mathbb{Q}$. The corresponding empirical distributions, composed of n and m examples, are denoted as $\mathbb{P}_n$ and $\mathbb{Q}_m$, respectively. Similarly, the noisy distributions are defined as $\widetilde{\mathbb{P}}$, $\widetilde{\mathbb{Q}}$, $\widetilde{\mathbb{P}}_n$, and $\widetilde{\mathbb{Q}}_m$, respectively. Given above notational definitions, we can instead estimate the empirical value of $disc_{\mathcal{F}_\Theta}(\mathbb{P}, \mathbb{Q})$ via importance reweighting on noisy data [28] by introducing density ratio functions $\beta_{\mathbb{P}}(\cdot,\cdot)$ and $\beta_{\mathbb{Q}}(\cdot,\cdot)$, namely

$$\widehat{disc}_{\mathcal{F}_\Theta}(\widetilde{\mathbb{P}}_n, \widetilde{\mathbb{Q}}_m) = \sup_{f_{\boldsymbol{\theta}} \in \mathcal{F}_\Theta} \left| \mathbb{E}_{\widetilde{\mathbb{P}}_n} \left[\beta_{\mathbb{P}}(\mathbf{x}, \widetilde{\mathbf{y}}) \ell(f_{\boldsymbol{\theta}}(\mathbf{x}), \widetilde{\mathbf{y}})\right] - \mathbb{E}_{\widetilde{\mathbb{Q}}_m} \left[\beta_{\mathbb{Q}}(\mathbf{x}, \widetilde{\mathbf{y}}) \ell(f_{\boldsymbol{\theta}}(\mathbf{x}), \widetilde{\mathbf{y}})\right] \right|. \tag{10}$$

C. Critical Factors Leading to Catastrophic Forgetting

Based on the analysis of three terms in Theorem 1 from Section IV-A∼IV-B, we provide the main result in Theorem 3 below, namely

Theorem 3. (Main Theorem of Cumulative Generalization Error Bound): Under Assumptions 1∼2, when the dataset $\{\widetilde{\mathcal{M}}_{1:t-1}^{(k)}\}_{k=1}^K \cup \widetilde{\mathcal{S}}_t$ is sampled from the distribution $\widetilde{\mathbb{P}}_1^{m_1} \otimes \widetilde{\mathbb{P}}_2^{m_2} \otimes \cdots \otimes \widetilde{\mathbb{P}}_K^{m_K} \otimes \widetilde{\mathbb{D}}_t^{|\widetilde{\mathcal{S}}_t|}$, with probability at least $1-\delta$, the final cumulative generalization error bound for any $f_{\boldsymbol{\theta}} \in \mathcal{F}_\Theta$ is:

$$\begin{aligned}
&\mathbb{E}_{\mathbb{D}_{1:t}}[\ell(f_{\boldsymbol{\theta}}(\mathbf{x}), \mathbf{y})] \\
&\leq \underbrace{\sum_{(\mathbf{x}_i, \widetilde{\mathbf{y}}_i) \in \widetilde{\mathcal{M}}_{1:t-1}} \left(\frac{1-\alpha_t}{|\widetilde{\mathcal{M}}_{1:t-1}|} + \alpha_t q_i \right) \beta_{\mathbb{P}_{1:t-1}^{\mathcal{M}}}(\mathbf{x}_i, \widetilde{\mathbf{y}}_i) \ell(f_{\boldsymbol{\theta}}(\mathbf{x}_i), \widetilde{\mathbf{y}}_i)}_{\text{Weighted loss on buffered data}} \\
&+ \underbrace{\alpha_t \cdot \sum_{(\mathbf{x}_j, \widetilde{\mathbf{y}}_j) \in \widetilde{\mathcal{S}}_t} q_j \beta_{\mathbb{D}_t}(\mathbf{x}_j, \widetilde{\mathbf{y}}_j) \ell(f_{\boldsymbol{\theta}}(\mathbf{x}_j), \widetilde{\mathbf{y}}_j)}_{\text{Weighted loss on new data}} + \underbrace{O(1)}_{\text{Approximation error}} \\
&+ \underbrace{\alpha_t \sum_{k=1}^K \overline{q}_k \widehat{disc}_{\mathcal{F}_\Theta}(\widetilde{\mathbb{P}}_k, \widetilde{\mathbb{P}}_{\mathcal{S}_t})}_{\text{Distribution shift}} + \underbrace{(1-\alpha_t) Gap(\mathbb{P}_{1:t-1}^{\mathcal{M}}, \mathbb{D}_{1:t-1})}_{\text{Buffered data selection bias}},
\end{aligned} \tag{11}$$

where q_i is the i-th element of $\mathbf{q}$, which satisfies $q_i > 0$ and $\|\mathbf{q}\|_1 = 1$, and $\overline{q}_k$ represents the sum of the weights assigned to the examples in the k-th cluster.

Theorem 3 reveals that catastrophic forgetting is fundamentally influenced by the three critical factors below, namely

Algorithm 1: Greedy Algorithm for the k-Center Problem: Cover($\mathcal{S}$, k).

Input: Dataset $\mathcal{S} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$; and the number of selected examples k.
$\mathcal{M} \leftarrow$ {a random integer j sampled from $\{1, 2, \ldots, n\}$};
Initialize a distance matrix D^{min} with size n;
$D_i^{min} \leftarrow \|\mathbf{x}_i - \mathbf{x}_j\|_2, \forall\, i \in [\![n]\!]$;
while $|\mathcal{M}| < k$ **do**
  $u \in \arg\max_{i \in [\![n]\!] \setminus \mathcal{M}} D_i^{min}$;
  $\mathcal{M} \leftarrow \mathcal{M} \cup \{u\}$;
  $D_i^{min} \leftarrow \min\{D_i^{min}, \|\mathbf{x}_i - \mathbf{x}_u\|_2\}, \forall\, i \in [\![n]\!]$.
end while
Output: Selected subset $\mathcal{M}$.

- **Buffered data selection bias:** The selection bias term contains $Gap(\mathbb{P}_{1:t-1}^{\mathcal{M}}, \mathbb{D}_{1:t-1})$, which quantifies the distribution gap between buffered data and the data seen so far, and a smaller bias can contribute to a tighter bound.
- **Distribution shift:** The K clusters $\{\mathcal{M}_{1:t-1}^{(k)}\}_{k=1}^K$ may exhibit distribution shift relative to $\widetilde{\mathcal{S}}_t$, and the discrepancies among their distributions (e.g., $\widetilde{\mathbb{P}}_k$ and $\widetilde{\mathbb{P}}_{\mathcal{S}_t}$) are characterized by the terms $\widehat{disc}_{\mathcal{F}_\Theta}(\widetilde{\mathbb{P}}_k, \widetilde{\mathbb{P}}_{\mathcal{S}_t}), \forall\, k \in [\![K]\!]$. Therefore, for a larger discrepancy value, a smaller weight $\overline{q}_k$ can lead to a tighter bound.
- **Label noise:** The density ratio functions (i.e., $\beta_{\mathbb{P}_{1:t-1}^{\mathcal{M}}}(\cdot,\cdot)$ and $\beta_{\mathbb{D}_t}(\cdot,\cdot)$) appearing in the right-hand side of (11) serve as measures of importance for examples. Consequently, incorrectly labeled examples, which typically incur large loss values, should be assigned low importance values to ensure a tight bound.

V. The Proposed CNLDD Method

In this section, we design practical algorithms to tackle the three challenging factors revealed by Theorem 3. After that, the overall optimization method is introduced to minimize the upper bound of cumulative generalization error.

A. A Two-Step Buffer Update Strategy

To ensure a minimum selection bias of buffered data, we propose a two-step buffer update strategy. To introduce this strategy, we need some additional definitions. We begin by presenting the concept of "covering radius" [35]. Intuitively, we draw spheres of uniform size centered at arbitrary points within a subset of data. The smallest radius needed to encompass all the data is then referred to as the covering radius. Formally, for a set $\mathcal{U}$ and its subset $\mathcal{U}^{\text{sub}}$, the covering radius γ of $\mathcal{U}^{\text{sub}}$ w.r.t. $\mathcal{U}$ is defined as:

$$\gamma := \max_{\mathbf{u} \in \mathcal{U}} \min_{\mathbf{u}' \in \mathcal{U}^{\text{sub}}} \|\mathbf{u} - \mathbf{u}'\|_2. \tag{12}$$

Furthermore, we refer to $\mathcal{U}^{\text{sub}}$ as a γ-cover of set $\mathcal{U}$.

Identifying a subset that has a minimum covering radius γ with k points is known as the k-center problem [14]. The k-center problem assumes that the original set is $\mathcal{V}$, and the subset is

$\mathcal{C}$. The goal is to identify an optimal subset $\mathcal{C}^*$ of size k that minimizes the covering radius, i.e.,

$$\mathcal{C}^* \in \arg\min_{|\mathcal{C}|=k, \mathcal{C}\subset\mathcal{V}} \max_{\mathbf{v}\in\mathcal{V}} \min_{\mathbf{c}\in\mathcal{C}} \|\mathbf{v}-\mathbf{c}\|_2. \tag{13}$$

However, the k-center problem is NP-hard [7], making it computationally expensive to find an exact solution for large-scale datasets. In light of this, several "2-opt" approximation algorithms have been proposed [38]. In particular, these algorithms guarantee that if the optimal covering radius is r^*, the approximation algorithm yields a solution with a covering radius no greater than $2r^*$. Among them, the simple greedy algorithm (summarized in Algorithm 1) is widely recognized for its effectiveness as a 2-opt method.

By the definition of covering radius in (12), the subset with a minimum covering radius typically contains the most representative examples in $\mathcal{U}$. Motivated by this observation, we design a two-step buffer update strategy to ensure that the buffered examples are the most representative data seen so far. Specifically, at timestep t, if the memory buffer is not full, all newly received examples are directly added to it. If the memory buffer is full, we first select a subset from new data $\widetilde{\mathcal{S}}_t$ with a minimal covering radius, which is denoted by $\widetilde{\mathcal{M}}_t$. In practice, we use the ratio ρ to represent the proportion of examples selected from the new data $\widetilde{\mathcal{S}}_t$, such that $\widetilde{\mathcal{M}}_t$ has a size of $\left|\widetilde{\mathcal{S}}_t\right| \cdot \rho$. Subsequently, the ultimate memory buffer for timestep-t (i.e., $\widetilde{\mathcal{M}}_{1:t}$) is constructed by selecting M examples from the union $\widetilde{\mathcal{M}}_{1:t-1} \cup \widetilde{\mathcal{M}}_t$ with a minimal covering radius, where M is the size of memory buffer. Since Algorithm 1 is easy to implement and can already produce satisfactory performance, it is employed at each step of the proposed two-step buffer update strategy.

As a 2-opt approximation algorithm, the greedy method adopted in our buffer update strategy keeps the covering radius close to the optimal value, which means that the buffered data are sufficiently representative of the data seen so far. Furthermore, as demonstrated in the post-hoc theoretical justification (Section VI), our strategy effectively leads to a small selection bias $Gap(\mathbb{P}^{\mathcal{M}}_{1:t-1}, \mathbb{D}_{1:t-1})$. Therefore, by employing our strategy, a tight upper bound on the generalization error in Theorem 3 can be achieved, which alleviates catastrophic forgetting in a principled manner.

B. The Estimation of Density Ratio Functions

In Theorem 3, several density ratio functions must be estimated, namely $\beta_{\mathbb{P}^{\mathcal{M}}_{1:t-1}}(\cdot,\cdot)$, $\beta_{\mathbb{D}_t}(\cdot,\cdot)$, $\beta_{\mathbb{P}_k}(\cdot,\cdot)$, and $\beta_{\mathbb{P}_{\mathcal{S}_t}}(\cdot,\cdot)$, which play a crucial role in mitigating label noise in both buffered data and new data. Since estimating each of these density ratios individually may result in significant estimation error due to the limited number of buffered data and incoming data, we instead propose a unified procedure to estimate all density ratios simultaneously.

To this end, we employ the HOC algorithm introduced in Section III. To apply HOC algorithm, we merge the data from the memory buffer $\widetilde{\mathcal{M}}_{1:t-1}$ with the data for timestep-t (i.e., $\widetilde{\mathcal{S}}_t$). Given each example $(\mathbf{x}, \widetilde{\mathbf{y}})$ in the combined dataset, the transition matrix $\mathbf{T}(\mathbf{x})$ and the corresponding noisy posterior $\widetilde{P}(\mathbf{e}_k|\mathbf{x}) = \sum_{j=1}^{C} T_{jk}(\mathbf{x}) \cdot P(\mathbf{e}_j|\mathbf{x})$ are estimated. Consequently, by Definition 2, for any example $(\mathbf{x}, \widetilde{\mathbf{y}}) \in \widetilde{\mathcal{M}}_{1:t-1}$, its density ratio can be expressed as:

$$\beta_{\mathbb{P}^{\mathcal{M}}_{1:t-1}}(\mathbf{x}, \widetilde{\mathbf{y}}) = \frac{\widehat{p}(Y=\widetilde{\mathbf{y}}|X=\mathbf{x})}{\widetilde{\mathbf{y}}^\top \mathbf{T}^\top(\mathbf{x}) \widehat{\mathbf{p}}(Y|X=\mathbf{x})}, \tag{14}$$

where $\widehat{\mathbf{p}}(Y|X=\mathbf{x}) = [\widehat{p}(Y=\mathbf{e}_1|X=\mathbf{x}), \widehat{p}(Y=\mathbf{e}_2|X=\mathbf{x}), \ldots, \widehat{p}(Y=\mathbf{e}_C|X=\mathbf{x})]^\top =: f_{\boldsymbol{\theta}}(\mathbf{x})$ represents the output probability vector of the hypothesis $f_{\boldsymbol{\theta}}$. Likewise, for any example $(\mathbf{x}, \widetilde{\mathbf{y}})$ sampled from $\mathbb{D}_t$, $\mathbb{P}_k$, and $\mathbb{P}_{\mathcal{S}_t}$, its density ratio can also be estimated via (14). For simplicity, we use the abbreviated symbol $\beta(\mathbf{x}, \widetilde{\mathbf{y}})$ to represent one of the four density ratio functions according to the related distribution that the specific example $(\mathbf{x}, \widetilde{\mathbf{y}})$ falls into.

C. The Estimation of Distribution Discrepancy

In addition to the factors of buffered data selection bias and label noise, it is also necessary to design an algorithm for computing the terms regarding distribution shift in Theorem 3. To achieve this, we propose a practical approach to estimate distribution discrepancy terms.

First of all, K clusters of buffered data must be identified. In this paper, we directly employ the K-Means method to obtain the clusters $\{\mathcal{M}^{(k)}_{1:t-1}\}_{k=1}^{K}$, due to its simplicity and its demonstrated effectiveness in our experiments. Moreover, since the hypothesis space $\mathcal{F}_\Theta$ can be highly complicated (such as the neural networks used in this paper), the term regarding distribution shift, i.e., $\widehat{disc}_{\mathcal{F}_\Theta}(\widetilde{\mathbb{P}}_k, \widetilde{\mathbb{P}}_{\mathcal{S}_t})$, is difficult to compute directly. To address this issue, we propose to employ a linear hypothesis space with a fixed latent representation as a simplified alternative to $\mathcal{F}_\Theta$, namely

$$\mathcal{F}^{\text{lin}}_\Theta = \left\{ f_{\boldsymbol{\theta}} : f_{\boldsymbol{\theta}}(\mathbf{x}) = \boldsymbol{\theta}^\top \phi(\mathbf{x}), \boldsymbol{\theta} \in \Theta \subset \mathbb{R}^{d_\phi \times C} \right\}, \tag{15}$$

where $\phi(\mathbf{x}) \in \mathbb{R}^{d_\phi}$ represents the latent representation of $\mathbf{x}$, and d_ϕ denotes its dimensionality.

Next, we briefly outline the estimation of distribution discrepancy based on $\mathcal{F}^{\text{lin}}_\Theta$. For clarity, we denote the two underlying distributions as $\mathbb{P}$ and $\mathbb{Q}$. The corresponding density ratio vectors are represented as $\boldsymbol{\beta}_1 = (\beta_{\mathbb{P}}(\mathbf{x}_i, \widetilde{\mathbf{y}}_i))_{1\le i\le n}$ and $\boldsymbol{\beta}_2 = (\beta_{\mathbb{Q}}(\mathbf{x}_j, \widetilde{\mathbf{y}}_j))_{1\le j\le m}$, respectively. Subsequently, the optimal hypothesis $f_{\boldsymbol{\theta}^*}$ can be obtained by solving the following problem:

$$\min_{f_{\boldsymbol{\theta}}\in\mathcal{F}^{\text{lin}}_\Theta} \left\{ \frac{1}{n}\sum_{i=1}^{n} \beta_{1i}\ell(f_{\boldsymbol{\theta}}(\mathbf{x}_i), \widetilde{\mathbf{y}}_i) - \frac{1}{m}\sum_{j=1}^{m} \beta_{2j}\ell(f_{\boldsymbol{\theta}}(\mathbf{x}_j), \widetilde{\mathbf{y}}_j) \right\}. \tag{16}$$

The procedure for solving this problem is detailed in Section C.1 of supplementary material, where the Cross-Entropy loss is adopted as the loss function. The optimal hypothesis $f_{\boldsymbol{\theta}^*}$, once obtained, is then utilized to compute the empirical discrepancy in (10). It is also proven in Section B.3 of supplementary material that the empirical estimation $\widehat{disc}_{\mathcal{F}^{\text{lin}}_\Theta}(\widetilde{\mathbb{P}}_n, \widetilde{\mathbb{Q}}_m)$ achieves an approximation error of $\mathcal{O}(\sqrt{\frac{1}{m}+\frac{1}{n}})$ to $disc_{\mathcal{F}^{\text{lin}}_\Theta}(\mathbb{P}, \mathbb{Q})$ under mild conditions.

Algorithm 2: **C**ontinual **N**oisy **L**abel Learning on **D**rifting **D**ata Streams (CNLDD).

Input: New data $\widetilde{\mathcal{S}}_t = \{(\mathbf{x}_i, \widetilde{\mathbf{y}}_i)\}_{i=1}^{n_t}$; the memory buffer $\widetilde{\mathcal{M}}_{1:t-1}$; selection ratio ρ; the capacity M of memory buffer; number of clusters K; number of iterations I; and the parameter $\boldsymbol{\theta}_{t-1,I+1}$ learned at the $t-1$-th timestep.
Estimate the noise transition matrix $\mathbf{T}(\mathbf{x})$ for $\forall\,(\mathbf{x}, \widetilde{\mathbf{y}}) \in \widetilde{\mathcal{M}}_{1:t-1} \cup \widetilde{\mathcal{S}}_t$ using the HOC algorithm [52];
Calculate the density ratio $\beta(\mathbf{x}, \widetilde{\mathbf{y}})$ for $\forall\,(\mathbf{x}, \widetilde{\mathbf{y}}) \in \widetilde{\mathcal{M}}_{1:t-1} \cup \widetilde{\mathcal{S}}_t$ using (14);
Partition $\widetilde{\mathcal{M}}_{1:t-1}$ into K clusters $\{\widetilde{\mathcal{M}}_{1:t-1}^{(k)}\}_{k=1}^{K}$ using K-Means clustering;
for $k = 1$ to K **do**
  Estimate $\widehat{disc}_{\mathcal{F}_\Theta^{\text{lin}}}(\widetilde{\mathbb{P}}_k, \widetilde{\mathbb{P}}_{\mathcal{S}_t})$ via (16);
end for
$\mathbf{q}_{t,1} \leftarrow$ Uniform Distribution;
$\boldsymbol{\theta}_{t,1} \leftarrow \boldsymbol{\theta}_{t-1,I+1}$;
for $j = 1$ to I **do**
  Compute the updated parameters $\boldsymbol{\theta}_{t,j+1}$ and $\mathbf{q}_{t,j+1}$ via (18);
end for
$\widetilde{\mathcal{M}}_t \leftarrow \text{cover}(\widetilde{\mathcal{S}}_t, \rho \cdot |\widetilde{\mathcal{S}}_t|)$;
$\widetilde{\mathcal{M}}_{1:t} \leftarrow \text{cover}(\widetilde{\mathcal{M}}_{1:t-1} \cup \widetilde{\mathcal{M}}_t, M)$.
Output: The predictive model $f_{\boldsymbol{\theta}_{t,I+1}}$ and the updated memory buffer $\widetilde{\mathcal{M}}_{1:t}$ for t timesteps.

D. The Overall Optimization Problem

So far, we have thoroughly investigated each term in the upper bound of cumulative generalization error in Theorem 3. In this section, we present the optimization procedure designed to minimize this bound.

To obtain a tight upper bound for the generalization error, we propose to alternatively optimize over the hypothesis $f_{\boldsymbol{\theta}}$ and the weight vector $\mathbf{q}$ in Theorem 3. The approximation error is negligible as it does not depend on $f_{\boldsymbol{\theta}}$ or $\mathbf{q}$. To further regularize the optimization of $\mathbf{q}$, we add two terms to the objective function, namely $\|\mathbf{q} - \mathbf{p}^0\|_1$ and $\|\mathbf{q}\|_2$, where $\mathbf{p}^0$ is a prior for $\mathbf{q}$. By appending the two terms, prior knowledge can be leveraged, namely, we can assign more weight to historical data if they are more important than newly arrived data, and vice versa. Here, we set $\mathbf{p}^0$ as a uniform vector, which ensures the full utilization of data while preventing the collapse of $\mathbf{q}$. Finally, by considering all relevant factors, the upper bound of generalization error forms the following objective function:

$$\mathcal{L}_t(\boldsymbol{\theta}, \mathbf{q}) := \underbrace{\sum_{i=1}^{|\widetilde{\mathcal{M}}_{1:t-1}|} \left(\alpha_t q_i + \frac{1-\alpha_t}{|\widetilde{\mathcal{M}}_{1:t-1}|}\right) \beta(\mathbf{x}_i, \widetilde{\mathbf{y}}_i) \ell(f_{\boldsymbol{\theta}}(\mathbf{x}_i), \widetilde{\mathbf{y}}_i)}_{\text{Term 1}}$$
$$+ \underbrace{\alpha_t \sum_{j=1}^{|\widetilde{\mathcal{S}}_t|} q_j \beta(\mathbf{x}_j, \widetilde{\mathbf{y}}_j) \ell(f_{\boldsymbol{\theta}}(\mathbf{x}_j), \widetilde{\mathbf{y}}_j)}_{\text{Term 2}} + \underbrace{\lambda_1 \|\mathbf{q} - \mathbf{p}^0\|_1}_{\text{Term 3}} + \underbrace{\lambda_2 \|\mathbf{q}\|_2}_{\text{Term 4}}$$
$$+ \underbrace{\sum_{k=1}^{K} \overline{q}_k \widehat{disc}_{\mathcal{F}_\Theta^{\text{lin}}}(\widetilde{\mathbb{P}}_k, \widetilde{\mathbb{P}}_{\mathcal{S}_t})}_{\text{Term 5}}, \tag{17}$$

where the density ratio $\beta(\cdot,\cdot)$ can be estimated via (14). Moreover, λ_1 and λ_2 are two non-negative hyperparameters. In (17), Term 1 and Term 2 correspond to the losses on buffered data and new data, respectively. Term 3 and Term 4 serve as regularizers for $\mathbf{q}$. Additionally, Term 5 captures the discrepancy between the distributions of the K clusters and the data distribution at timestep t.

Subsequently, we present the detailed procedure for the optimization over $f_{\boldsymbol{\theta}}$ and $\mathbf{q}$ in the objective function $\mathcal{L}_t(\boldsymbol{\theta}, \mathbf{q})$. Let I denote the number of iterations performed at each timestep. We denote by $\boldsymbol{\theta}_{t,j}$ and $\mathbf{q}_{t,j}$ the parameters after j iterations at timestep t. The parameter $\boldsymbol{\theta}$ and the weight vector $\mathbf{q}$ are updated iteratively according to (18):

$$\begin{cases} \boldsymbol{\theta}_{t,j+1} = \boldsymbol{\theta}_{t,j} - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}_t(\boldsymbol{\theta}, \mathbf{q}_{t,j})\big|_{\boldsymbol{\theta}_{t,j}} \\ \mathbf{q}_{t,j+1} = \text{Proj}\left(\mathbf{q}_{t,j} - \eta \nabla_{\mathbf{q}} \mathcal{L}_t\left(\boldsymbol{\theta}_{t,j+1}, \mathbf{q}\right)\big|_{\mathbf{q}_{t,j}}\right) \end{cases}, \tag{18}$$

where $\text{Proj}(\cdot)$ denotes a projection operator onto the probability simplex, and η is the learning rate. For a p-dimensional vector $\boldsymbol{\nu} = (\nu_i)_{1\le i\le p} \in \mathbb{R}^p$, this operator is defined as $\text{Proj}(\boldsymbol{\nu}) = ([\nu_i - \mu^*]_+)_{1\le i\le p}$, where μ^* is the unique solution to the equation $\mathbf{1}^\top [\mathbf{q} - \mu^* \mathbf{1}]_+ = 1$ [4]. Here, $\mathbf{1} \in \mathbb{R}^p$ is an all-one vector, and $[a]_+ = \max\{a, 0\}$ for any scalar a.

The main steps of the proposed CNLDD method are summarized in Algorithm 2, where $\text{cover}(\cdot,\cdot)$ denotes the greedy k-center approach described in Algorithm 1. For this algorithm, we provide a detailed computational complexity analysis in Section D of the supplementary material. In Section E, we further present a per-step runtime analysis and compare the computational cost of CNLDD with those of representative continual noisy label learning methods in memory update and model training. The results demonstrate that, although our method involves multiple optimization steps, its overall runtime remains acceptable.

VI. Theoretical Analysis of CNLDD Method

In this section, we provide supplementary justifications to our specifically designed CNLDD method.

First, we present rigorous theoretical support for our buffer update strategy, demonstrating its close relationship to the minimization of $Gap(\mathbb{P}^{\mathcal{M}}_{1:t-1}, \mathbb{D}_{1:t-1})$ in Theorem 3. Our analysis is mainly based on the following lemma:

Lemma 1: Under Assumption 2, we further assume that the class posterior probability $P_t(Y_t = \mathbf{y}|X_t = \mathbf{x})$ is λ^P-Lipschitz continuous at any timestep, and the loss function $\ell(f(\cdot), \mathbf{y})$ is λ^ℓ-Lipschitz continuous for all $\mathbf{y}$. At timestep $t-1$, we assume the following: the memory buffer $\widetilde{\mathcal{M}}_{1:t-2}$ with size k_1 is a γ-cover of the entire dataset $\widetilde{\mathcal{S}}_{1:t-2}$; the temporary buffer $\widetilde{\mathcal{M}}_{t-1}$ with size k_2 is a γ'-cover of the data $\widetilde{\mathcal{S}}_{t-1}$; and the updated memory buffer $\widetilde{\mathcal{M}}_{1:t-1}$ with size k_3 at timestep $t-1$ is a γ''-cover of $\widetilde{\mathcal{M}}_{1:t-2} \cup \widetilde{\mathcal{M}}_{t-1}$. If the loss on buffered data is sufficiently small, i.e., $\mathbb{E}_{\mathbb{P}^{\mathcal{M}}_{1:t-1}}[\ell(f(\mathbf{x}), \mathbf{y})] < \epsilon$ with ϵ being a very small positive value,

then with probability at least $1-\delta$, we have:

$$Gap(\mathbb{P}^{\mathcal{M}}_{1:t-1}, \mathbb{D}_{1:t-1}) < 2\epsilon + (\max\{\gamma, \gamma'\} + \gamma'')(\lambda^{\ell} + \lambda^{P}\overline{L}C) + 2\overline{L}\sqrt{\frac{\log(2/\delta)}{2|\widetilde{\mathcal{S}}_{1:t-1}|}}. \quad (19)$$

The above lemma demonstrates that an appropriate selection of data will lead to small values of γ, γ' and γ'', which further result in a lower upper bound for $Gap(\mathbb{P}^{\mathcal{M}}_{1:t-1}, \mathbb{D}_{1:t-1})$. In view of this, by applying Lemma 1, we show in Theorem 4 that the two-step buffer update strategy proposed in Section V-A can induce an upper bound characterized by a single covering radius γ.

Theorem 4: Based on Lemma 1, if the update strategy for the memory buffer at timestep $t-1$ is designed as follows: first, a $\frac{\gamma}{t}$-cover of $\widetilde{\mathcal{S}}_{t-1}$, denoted by $\widetilde{\mathcal{M}}_{t-1}$, is selected; then, a $\frac{\gamma}{t}$-cover of $\widetilde{\mathcal{M}}_{1:t-2} \cup \widetilde{\mathcal{M}}_{t-1}$, denoted by $\widetilde{\mathcal{M}}_{1:t-1}$, is further selected. The resulting $\widetilde{\mathcal{M}}_{1:t-1}$ serves as the final memory buffer at timestep $t-1$. Based on this strategy, if the loss on buffered data is sufficiently small, i.e., $\mathbb{E}_{\mathbb{P}^{\mathcal{M}}_{1:t-1}}[\ell(f(\mathbf{x}), \mathbf{y})] < \epsilon$ with ϵ being a very small positive value, then with probability at least $1-\delta$, it holds that

$$Gap(\mathbb{P}^{\mathcal{M}}_{1:t-1}, \mathbb{D}_{1:t-1}) < \gamma \log(t+1)(\lambda^{\ell} + \lambda^{P}\overline{L}C) + 2\epsilon + 2\overline{L}\sqrt{\frac{\log(2/\delta)}{2|\widetilde{\mathcal{S}}_{1:t-1}|}} = \widetilde{\mathcal{O}}(\gamma) + \mathcal{O}\left(\sqrt{\frac{1}{\sum_{i=1}^{t-1} n_i}} + \epsilon\right), \quad (20)$$

where $\widetilde{\mathcal{O}}(\cdot)$ denotes the big-$\mathcal{O}$ that hides all logarithmic factors.

Theorem 4 establishes an upper bound on the distribution gap term $Gap(\mathbb{P}^{\mathcal{M}}_{1:t-1}, \mathbb{D}_{1:t-1})$, which includes a term $\widetilde{\mathcal{O}}(\gamma)$ that explicitly depends on the covering radius γ in each step of our update strategy. By substituting this bound into Theorem 3, we can readily derive the upper bound of cumulative generalization error specific to our CNLDD method. Recalling our two-step buffer update strategy in Section V-A, we ensure a minimum covering radius in each step of our strategy, and thus γ in (20) is minimized. Consequently, our buffer update strategy reduces the selection bias term in Theorem 3. In contrast to existing CNLL methods [3], [20], [21], which typically rely on heuristic selection strategies for memory buffer maintenance, our theoretical framework demonstrates that the proposed update strategy is intrinsically linked to the minimization of cumulative generalization error, contributing to improved performance in a principled manner.

Moreover, we demonstrate in supplementary material (Section B.7) that a prior $\mathbf{p}_0$ can also be introduced for the weight vector $\mathbf{q}$ in Theorem 3, and the corresponding upper bound exhibits a structure similar to that of (17). Disregarding the approximation error, the objective $\mathcal{L}_t(\boldsymbol{\theta}, \mathbf{q})$ serves as an exact upper bound for cumulative generalization error. Consequently, minimizing $\mathcal{L}_t(\boldsymbol{\theta}, \mathbf{q})$ can lead to a reduction in generalization error, and thus the proposed CNLDD method can effectively mitigate catastrophic forgetting induced by distribution shift and label noise.

VII. Experimental Results

To validate the effectiveness of the proposed CNLDD algorithm, we conducted extensive experiments on representative synthetic and real-world datasets. The parametric sensitivity and contribution of various modules in our CNLDD method are also investigated.

In this paper, we investigate three common types of label noise [6], [13] on both synthetic and real-world datasets, namely 1) symmetric noise, which means that for each example, we uniformly select a label different from its ground-truth label with probability $\frac{\epsilon}{C-1}$, where ϵ is the noise rate; 2) pairflip noise, where we flip the label of each example cyclically to the next class with a probability of ϵ; and 3) instance-dependent noise, where the noise rate of a certain example depends on its feature and here we adopt the noise generation strategy proposed in prior work [6]. Additionally, two levels of label noise, namely 20% and 40% , are considered in our experiments. For simplicity, "sym. ϵ", "asym. ϵ" and "inst. ϵ" are used to denote the symmetric noise, pairflip noise, and instance-dependent noise with the noise rate of ϵ, respectively. Furthermore, we also consider the noise-free case, which is denoted as "clean. 0% ".

The baseline methods adopted in our experiments are Finetune, ER [40], PuriDivER [3], SPR [21], CNLL [20], Meta-DR [41], Online EWC [23], AdaStreams [50], Co-teaching [16], ϵ-Softmax [42], ContinualCRUST [33], STAR [10], and WSC [51]. In detail, Finetune refers to a naive method that does not employ any strategy to combat label noise or catastrophic forgetting. It simply finetunes the trained model with new data at each timestep without preserving historical models or examples. ER is a classical replay-based paradigm in continual learning. Moreover, online EWC and STAR are representative regularization-based CL approaches, whereas Meta-DR serves as a typical CL method built upon meta-learning principles. PuriDivER, SPR, CNLL, and ContinualCRUST are methods specifically designed for continual noisy label learning, with their key differences lying in the mechanisms adopted for buffer update. Additionally, AdaStreams, Co-teaching, and ϵ-Softmax are typical LNL approaches based on statistic estimation, sample selection, and robust loss function design, respectively. WSC is a recently proposed method that aims to learn a robust representation space even in the presence of imprecise supervision. In accordance with the common practice in continual noisy label learning [20], a noisy memory buffer is incorporated in AdaStreams, Co-teaching, ϵ-Softmax, and WSC. During training, the reservoir sampling technique [40] is leveraged to sample and replay historical data, enabling these LNL methods to mitigate the problem of catastrophic forgetting to some extent. Notably, PuriDivER, SPR, CNLL, and WSC incorporate auxiliary strategies, such as data augmentation, self-supervised learning, and semi-supervised learning, to improve their robustness and generalization capability. Therefore, to ensure fairness, all compared methods employ AugMix data augmentation [18]. Overall, the selected methods include existing continual noisy label learning methods as well as continual learning and label noise learning methods based on different strategies, ensuring a comprehensive evaluation of the proposed CNLDD method.

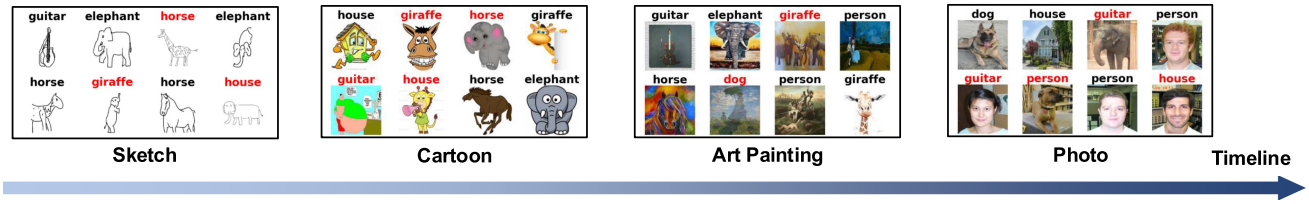

Fig. 1. The streaming training data constructed from *PACS* dataset. Each image is labeled with its observed class (one of seven categories), where black labels denote correct annotations while red labels indicate incorrect ones. Additionally, the image styles vary at certain timesteps, which is model-agnostic. The goal is to accurately predict the class of test images across all styles along the entire timeline after training.

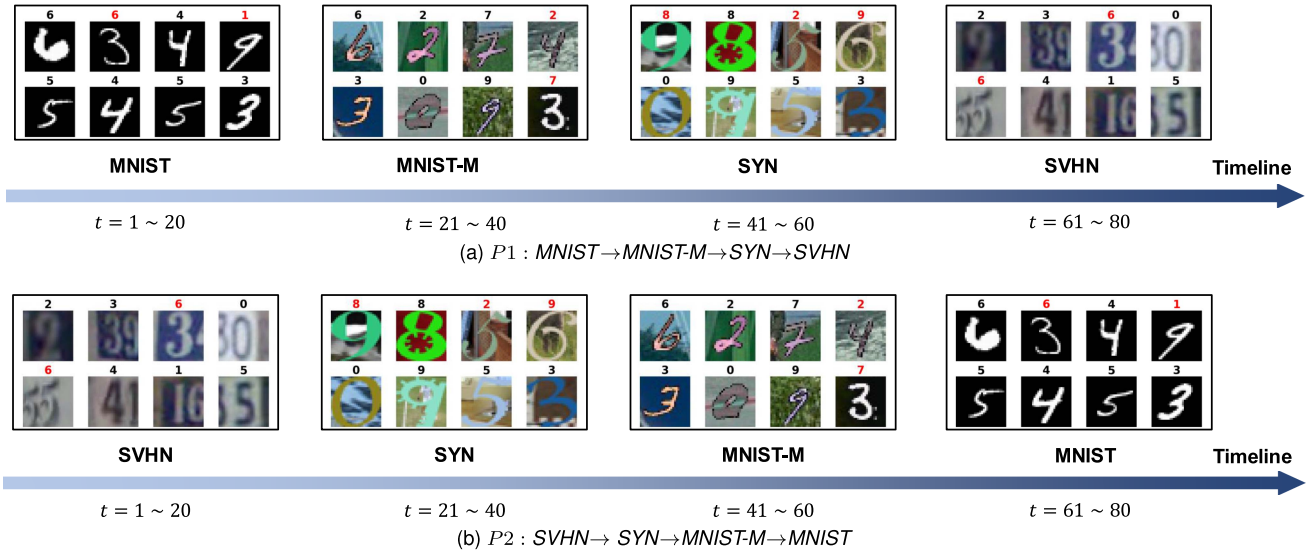

Fig. 2. Two arrangement protocols for *Digits*, where (a) and (b) illustrate protocol 1 (easy to difficult) and protocol 2 (difficult to easy), respectively. The number above each image indicates the observed class (one of ten categories), with black labels signifying correct annotations and red labels indicating incorrect ones. The objective is to accurately recognize the ten digits "0∼9" across various data subsets along the entire timeline.

A. *Experiments on Synthetic Datasets*

In this section, we use different domains or datasets to simulate the distribution shift in our setting. Inspired by the study of continual domain adaptation [41], we start by conducting experiments on two synthetic datasets, namely *PACS* [25] and *Digits* [41], which are broadly adopted by the computer vision community. Here, *Digits* comprises four digit datasets with different styles, namely, *MNIST*, *MNIST-M*, *SYN*, and *SVHN* [41].

The *PACS* dataset, widely utilized in image classification tasks, consists of 9,991 images across four distinct styles, namely Sketch, Cartoon, Art Painting, and Photo. In our setting, different styles in *PACS* refer to different data distributions, and the same type and level of label noise are applied to all styles. The training data derived from *PACS* is illustrated in Fig. 1. It can be seen that the style of images evolves over time from the most abstract to the most realistic. At each timestep, the model receives noisy examples from a certain style without being informed of the exact type of the style. Moreover, 80% of the data from each style in *PACS* is used for model training, while the rest is reserved for testing. The size of new data $\widetilde{\mathcal{S}}_t$ received by the model is set to 200. Additionally, in prior work [41], 100∼300 examples are buffered for each style. Since style information is unknown in our setting, here we set the size M of memory buffer as 200, resulting in a total of 40 timesteps.

For *Digits* dataset, by following previous study [41], we adopt two distinct protocols to organize the four data subsets, namely $P1$: *MNIST→MNIST-M→SYN→SVHN* and $P2$: *SVHN→SYN→MNIST-M→MNIST*. Here, the two protocols allow the evaluation of model performance from easy datasets to hard datasets and vice versa. The two protocols are shown in Fig. 2. For compatibility, the size of each image in *Digits* is adjusted to 28× 28pixels beforehand. Consistent with the setting of Meta-DR [41], 10,000 examples are randomly selected from each data subset for model training, and additional 2,000 noise-free examples from each dataset are selected for testing. In our experiments, the size of new data $\widetilde{\mathcal{S}}_t$ received by the model at each time step t is set to 500, and the size of the memory buffer M is set to 1,000, with 80 timesteps in total.

For *PACS* and *Digits* datasets, the adopted backbone network is ResNet-18 [17] and the network is pretrained on ImageNet [9] for *PACS*. On *PACS* dataset, we train the network for 20 iterations at each timestep with a batch size of 32. For our CNLDD, α_t is set to 0.2, because small α_t facilitates memory stability.

TABLE II
AVERAGE TEST ACCURACIES (%) OF VARIOUS APPROACHES ON *PACS* DATASET. THE TWO BEST RECORDS UNDER EACH NOISE SETTING ARE HIGHLIGHTED IN RED AND BLUE, RESPECTIVELY.

Method	clean. 0%	sym. 20%	sym. 40%	asym. 20%	asym. 40%	inst. 20%	inst. 40%
Finetune	39.69 ± 0.24	34.77 ± 0.34	27.89 ± 1.02	45.24 ± 0.91	37.04 ± 0.21	33.38 ± 0.41	26.92 ± 1.02
ER [5]	67.43 ± 0.53	53.39 ± 0.22	38.76 ± 1.09	50.43 ± 0.23	43.49 ± 0.51	50.96 ± 0.52	34.30 ± 0.14
Online EWC [23]	44.36 ± 0.11	36.98 ± 0.14	29.01 ± 0.72	38.03 ± 0.25	30.09 ± 2.52	33.28 ± 0.51	28.89 ± 0.62
AdaStreams [50]	86.85 ± 0.09	76.12 ± 0.24	61.70 ± 0.91	72.75 ± 0.71	52.66 ± 0.51	68.87 ± 0.63	27.94 ± 0.24
CNLL [20]	61.09 ± 0.27	54.34 ± 0.91	47.45 ± 0.55	57.98 ± 0.42	44.00 ± 0.61	53.50 ± 0.69	27.95 ± 0.90
Meta-DR [41]	78.02 ± 0.35	62.37 ± 0.13	46.40 ± 2.05	61.87 ± 0.45	50.78 ± 1.29	62.64 ± 0.10	51.36 ± 2.88
SPR [21]	58.99 ± 0.25	50.73 ± 0.22	42.29 ± 0.25	60.47 ± 0.52	47.10 ± 0.96	49.99 ± 0.81	39.90 ± 0.25
PuriDivER [3]	54.49 ± 0.18	45.72 ± 0.45	34.57 ± 0.99	53.34 ± 0.61	43.10 ± 0.14	45.86 ± 0.91	32.98 ± 0.61
Co-teaching [16]	82.77 ± 0.11	77.10 ± 0.29	60.07 ± 0.71	71.30 ± 0.81	57.16 ± 0.62	74.18 ± 0.90	43.52 ± 1.24
ϵ-Softmax [42]	71.99 ± 0.10	65.31 ± 0.25	53.13 ± 0.44	63.30 ± 0.88	45.87 ± 0.86	57.04 ± 0.05	25.47 ± 0.91
ContinualCRUST [33]	83.04 ± 0.23	64.00 ± 0.18	47.50 ± 1.81	64.61 ± 0.23	50.22 ± 2.41	67.48 ± 1.72	52.44 ± 1.67
STAR [10]	85.54 ± 0.90	67.28 ± 0.99	51.06 ± 1.29	68.66 ± 1.34	49.79 ± 3.92	64.04 ± 1.70	48.39 ± 3.09
WSC [51]	85.35 ± 0.25	76.21 ± 0.08	55.15 ± 1.73	73.68 ± 1.12	53.22 ± 0.70	72.24 ± 1.80	52.78 ± 0.25
CNLDD	85.75 ± 0.28	77.72 ± 0.24	62.22 ± 0.51	75.08 ± 0.82	58.80 ± 0.41	75.68 ± 0.34	60.85 ± 0.55

The number of clusters K is set to 6, and the selection ratio ρ is set to 0.3, which are tuned on a noisy validation set. The sensitivity analysis in Section VII-C also demonstrates that $5 \leq K \leq 8$ and $\rho \in \{0.2, 0.3\}$ typically yield satisfactory performance. Additionally, in all experiments, the coefficients λ_1 and λ_2 are set to 10.0 and 0.01, respectively, since this configuration often leads to satisfactory performance across all datasets. For *Digits* dataset, the batch size is 128 for all the compared methods. For our CNLDD, the coefficient α_t is set to 0.1, the ratio ρ is set to 0.3, the number of clusters is set to $K = 2$. For all compared methods, the Adam optimizer [22] is employed. We record the highest test accuracy from the last five timesteps of each experiment and report the mean and standard deviation of accuracies from three independent trials [3], [20], [21].

The experimental results on *PACS* dataset are shown in Table II. As shown in this table, our CNLDD consistently ranks among the top two places across all noise settings. Particularly, in the "inst. 40% " noise setting, CNLDD outperforms Meta-DR by nearly 9% in average accuracy. This highlights the advantage of explicitly modeling the instance-dependent transition matrix over clean sample selection. Moreover, although CNLL is specifically designed for continual noisy label learning, its performance suffers due to its failure to mitigate the impact of distribution shift. In contrast, our CNLDD evaluates the importance of buffered data based on distribution discrepancy, and thus it achieves superior performance when compared with CNLL. Additionally, when comparing label noise learning methods (e.g., Co-teaching and ϵ-Softmax) with continual learning methods (e.g., ER and Online EWC) which lack explicit noise-handling strategies, it reflects that label noise significantly exacerbates model forgetting and hinders the memorization of new data. Consequently, typical continual learning methods often perform poorly.

The experimental results on *Digits* dataset using two distinct protocols are presented in Table III. It can be seen that, under the "inst. 40% " noise condition, the accuracy of CNLDD surpasses ϵ-Softmax by approximately 12% , indicating that robust loss functions designed without explicitly modeling the noise generation process often perform poorly in complex noise scenarios. Furthermore, under the setting of "asym. 40% ", the test accuracy of CNLDD exceeds that of the second-best method, i.e., Co-teaching, by 3.58% . This suggests that the theoretically grounded CNLDD method is more effective than the experience-driven LNL method based on the technique of sample selection.

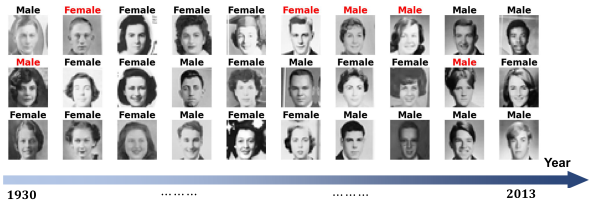

Fig. 3. The streaming noisy data constructed from the *Yearbook* dataset for model training. Each image is labeled with the observed class (one of two categories), where black labels represent correct annotations while red labels indicate incorrect ones. Due to the changes in societal norms, fashion styles, and demographics over time, the styles of images, hairstyles, and other features evolve over time. The goal is to accurately predict the gender of each face across all the time periods.

B. Experiments on Real-World Datasets With Natural Distribution Shift

In addition to evaluating the effectiveness of our CNLDD over baseline methods on synthetic datasets, we also conduct experiments on two real-world datasets with temporal distribution shift and label noise.

The real-world classification datasets utilized are *Yearbook* and *FMoW* (Functional Map of the World) from the Wild-Time benchmark [47]. The *Yearbook* dataset contains 33,431 annotated frontal-view yearbook photographs of American high school students spanning 1930∼2013. Each image is a single-channel grayscale image with the resolution of 32×32. The dataset provides a correct label for each face image to indicate the gender. Due to the changes in societal norms, fashion styles, and demographics, the styles of the images, clothing, and other features in *Yearbook* dataset also change over time, resulting in natural distribution shift [47]. Fig. 3 demonstrates some training examples sampled from different time periods in *Yearbook* dataset. This figure illustrates the subtle shift in data distribution over time, which differs from the abrupt distribution shift observed in *PACS* and *Digits* datasets introduced in the previous section. To adapt the *Yearbook* dataset to the setting studied in this paper, 20% of the images and their correct labels are randomly selected from each year as the test set, while the remaining examples are used as the training set.

TABLE III
AVERAGE TEST ACCURACIES (%) OF VARIOUS APPROACHES ON *DIGITS* DATASET. THE TWO BEST RECORDS UNDER EACH NOISE SETTING ARE HIGHLIGHTED IN RED AND BLUE, RESPECTIVELY.

Protocol	Method	clean. 0%	sym. 20%	sym. 40%	asym. 20%	asym. 40%	inst. 20%	inst. 40%
$P1$	Finetune	60.30 ± 0.11	51.09 ± 0.23	45.60 ± 4.26	59.77 ± 0.27	43.05 ± 0.33	48.38 ± 0.52	42.11 ± 0.54
	ER [5]	86.11 ± 0.42	67.44 ± 0.14	42.15 ± 0.55	74.06 ± 0.25	51.24 ± 0.66	71.48 ± 0.24	44.62 ± 0.91
	Online EWC [23]	58.59 ± 0.32	54.18 ± 0.14	45.77 ± 0.34	53.39 ± 0.55	45.43 ± 0.72	51.25 ± 0.13	43.21 ± 0.44
	AdaStreams [50]	74.81 ± 0.14	69.58 ± 0.20	52.22 ± 0.31	63.03 ± 0.23	44.05 ± 0.50	61.16 ± 0.15	30.90 ± 0.33
	CNLL [20]	78.42 ± 0.34	84.66 ± 0.32	65.95 ± 0.91	82.49 ± 0.55	60.66 ± 0.66	79.65 ± 0.11	58.22 ± 0.55
	Meta-DR [41]	91.41 ± 0.12	73.29 ± 0.33	48.20 ± 0.54	77.98 ± 0.20	54.71 ± 0.55	69.38 ± 1.00	51.89 ± 1.01
	SPR [21]	69.72 ± 0.23	60.14 ± 0.28	50.45 ± 1.05	62.11 ± 0.08	46.55 ± 0.05	60.12 ± 0.23	47.88 ± 0.99
	PuriDivER [3]	88.17 ± 0.06	73.12 ± 0.31	49.41 ± 1.22	76.20 ± 0.21	55.44 ± 1.02	72.50 ± 1.01	49.10 ± 1.55
	Co-teaching [16]	87.09 ± 0.42	82.45 ± 0.29	66.00 ± 0.82	83.53 ± 0.43	64.26 ± 0.22	80.94 ± 0.65	57.15 ± 0.39
	ϵ-Softmax [42]	80.28 ± 0.23	79.62 ± 0.25	66.06 ± 0.23	75.65 ± 0.35	53.92 ± 0.25	75.70 ± 0.24	49.24 ± 0.17
	ContinualCRUST [33]	91.88 ± 0.00	65.32 ± 1.32	45.68 ± 0.80	71.30 ± 1.62	51.04 ± 2.65	67.76 ± 0.56	53.80 ± 1.84
	STAR [10]	89.65 ± 0.16	72.46 ± 4.84	46.74 ± 0.64	74.56 ± 1.25	49.28 ± 4.67	72.98 ± 5.88	47.32 ± 3.21
	WSC [51]	88.56 ± 0.50	84.79 ± 0.08	64.71 ± 3.25	84.04 ± 0.59	51.57 ± 2.08	78.78 ± 3.10	50.69 ± 1.39
	CNLDD	92.05 ± 0.19	85.01 ± 0.18	67.05 ± 0.72	85.60 ± 0.65	67.84 ± 0.95	81.04 ± 0.13	61.69 ± 0.85
$P2$	Finetune	69.60 ± 0.24	56.02 ± 0.10	39.24 ± 0.99	59.62 ± 0.23	47.24 ± 0.77	63.24 ± 0.57	49.83 ± 0.30
	ER [5]	90.06 ± 0.14	76.06 ± 0.24	44.83 ± 0.45	76.11 ± 0.43	50.69 ± 0.23	72.74 ± 0.72	51.86 ± 1.02
	Online EWC [23]	67.00 ± 0.72	57.69 ± 0.19	38.59 ± 0.24	61.12 ± 0.91	49.29 ± 1.87	63.16 ± 0.39	47.26 ± 2.03
	AdaStreams [50]	78.95 ± 0.25	71.23 ± 0.07	55.67 ± 0.56	67.92 ± 0.23	49.25 ± 1.03	62.42 ± 0.53	40.24 ± 2.01
	CNLL [20]	81.24 ± 0.36	67.60 ± 0.90	53.56 ± 1.02	75.45 ± 0.34	64.39 ± 1.03	56.40 ± 0.32	38.89 ± 2.12
	Meta-DR [41]	73.19 ± 0.29	57.14 ± 0.02	44.67 ± 0.34	58.65 ± 1.01	45.27 ± 0.29	59.67 ± 0.25	48.15 ± 0.34
	SPR [21]	61.26 ± 0.10	48.44 ± 0.16	40.42 ± 0.26	50.40 ± 0.92	33.95 ± 2.94	47.06 ± 0.21	39.55 ± 2.24
	PuriDivER [3]	90.39 ± 0.23	77.91 ± 0.23	47.50 ± 1.02	78.69 ± 0.44	55.01 ± 0.32	77.19 ± 0.42	55.93 ± 0.35
	Co-teaching [16]	89.67 ± 0.08	81.05 ± 0.25	54.51 ± 0.23	77.27 ± 0.32	55.26 ± 1.02	76.94 ± 0.29	60.60 ± 1.29
	ϵ-Softmax [42]	85.38 ± 0.35	81.31 ± 0.56	53.71 ± 0.43	77.79 ± 0.42	53.01 ± 0.23	76.60 ± 0.19	50.52 ± 0.42
	ContinualCRUST [33]	92.24 ± 1.12	68.35 ± 1.62	43.97 ± 1.02	74.43 ± 2.95	53.16 ± 1.75	69.50 ± 1.85	56.46 ± 3.29
	STAR [10]	94.32 ± 0.65	73.97 ± 0.28	48.48 ± 5.51	78.04 ± 0.01	54.32 ± 1.38	76.50 ± 1.21	50.64 ± 0.64
	WSC [51]	92.05 ± 0.78	80.54 ± 0.23	61.51 ± 2.14	78.16 ± 0.71	51.05 ± 2.25	77.39 ± 1.84	57.30 ± 0.78
	CNLDD	93.49 ± 0.11	81.88 ± 0.23	63.65 ± 0.29	79.74 ± 0.23	62.08 ± 0.38	78.75 ± 0.48	61.34 ± 0.66

The *FMoW* dataset, comprising satellite imagery from 2002 to 2017, is initially designed to support humanitarian and policy efforts by monitoring croplands and predicting poverty levels. Due to human activity, temporal shift inevitably exists in the distribution of satellite imagery, making *FMoW* a suitable real-world dataset for studying distribution shift. In our experiments, we use the examples from the ten categories of this dataset with the most examples to mitigate the impact of minority categories on model training, resulting in 51,946 training examples and 6,432 test examples. Here, test examples are drawn uniformly from 2002 to 2017. During training, each input image is resized to 224×224×3, the size of new data at each timestep is set to 5,000, and the size of memory buffer is set to 2,000.

To align with the experiments in the previous section, we also investigate three noise settings, namely symmetric noise, pairflip noise and instance-dependent noise. It is important to note that the pairflip noise is not studied on *Yearbook* dataset due to its equivalence to symmetric label noise in binary classification. The backbone network employed in both datasets is ResNet-18 [17]. For our CNLDD method, α_t and ρ should be small to maintain historical knowledge and the number of clusters K should be set to a moderate value to ensure an effective knowledge transfer. The sensitivity analysis in Section VII-C also provides justifications on the choices of hyperparameters. Therefore, hyperparameters on *Yearbook* are set as $\rho = 0.25$, $K = 5$, and $\alpha_t = 0.1$, while for *FMoW* dataset they are $\rho = 0.20$, $K = 5$, and $\alpha_t = 0.2$.

The experimental results on *Yearbook* and *FMoW* datasets are presented in Table IV. As shown in this table, our CNLDD method achieves the highest classification accuracies across almost all the noise cases, except for the clean. 0% setting on *Yearbook* dataset. This is due to the sophisticated weight perturbation technique adopted in ContinualCRUST, which facilitates the learning of a stable parameter space in noise-free scenarios. On *Yearbook* dataset, CNLDD consistently outperforms AdaStreams under instance-dependent noise scenarios, which can be attributed to the fact that AdaStreams only models the class-conditional noise transition matrix, while the proposed CNLDD explicitly characterizes the instance-dependent noise transition matrix with the HOC [52] algorithm. On *FMoW* dataset, the methods relying on heuristic sample selection strategies, such as PuriDivER and Co-teaching, fail to achieve satisfactory performance. Additionally, the proposed CNLDD method outperforms the second-best method, i.e., WSC, by a margin of 2.01% under the most challenging setting (namely "inst. 40% "). This result implies that WSC exhibits limited capability in learning discriminative representations when exposed to severe label noise.

In a word, the classification results on real-world datasets clearly verify that CNLDD is also effective in handling classification tasks with temporal distribution shift.

C. Sensitivity Analysis

In this section, we perform detailed experiments to evaluate the sensitivity of the performance of our CNLDD to parameter variations. Specifically, the analysis focuses on the distribution combination coefficient α_t, the ratio ρ for the first-step subset selection in the buffer update strategy, the number of clusters K in the clustering algorithm, and the weight coefficients λ_1 and λ_2 in the objective function. To ensure generality, experiments are conducted on *PACS*, *Digits*, *Yearbook*, and *FMoW* datasets with the most challenging noise setting (namely "inst. 40% ").

TABLE IV
AVERAGE TEST ACCURACIES (%) OF VARIOUS APPROACHES ON *YEARBOOK* AND *FMOW* DATASETS. THE TWO BEST RECORDS UNDER EACH NOISE SETTING ARE HIGHLIGHTED IN RED AND BLUE, RESPECTIVELY.

Dataset	Method	clean. 0%	sym. 20%	sym. 40%	asym. 20%	asym. 40%	inst. 20%	inst. 40%
Yearbook	Finetune	83.26 ± 0.23	74.59 ± 0.35	57.61 ± 0.36	-	-	71.47 ± 0.12	55.54 ± 0.21
	ER [5]	94.01 ± 0.56	78.45 ± 1.23	59.50 ± 0.14	-	-	79.70 ± 0.31	61.68 ± 1.19
	Online EWC [23]	87.20 ± 0.41	72.33 ± 0.57	57.11 ± 1.32	-	-	71.87 ± 1.00	61.87 ± 0.43
	AdaStreams [50]	92.58 ± 0.90	84.73 ± 1.09	62.53 ± 1.34	-	-	86.14 ± 0.52	66.39 ± 0.44
	CNLL [20]	88.19 ± 0.42	87.52 ± 0.48	75.15 ± 0.23	-	-	87.33 ± 0.34	73.23 ± 0.08
	Meta-DR [41]	85.07 ± 0.36	81.16 ± 0.91	70.20 ± 0.32	-	-	81.16 ± 0.23	70.65 ± 0.55
	SPR [21]	84.57 ± 0.45	82.57 ± 0.28	70.49 ± 0.46	-	-	79.27 ± 1.21	67.19 ± 1.01
	PuriDivER [3]	92.95 ± 0.89	80.76 ± 1.43	69.88 ± 1.19	-	-	82.70 ± 1.03	68.87 ± 1.05
	Co-teaching [16]	91.27 ± 0.66	85.42 ± 0.45	70.62 ± 0.24	-	-	87.60 ± 0.29	70.62 ± 0.27
	ϵ-Softmax [42]	91.56 ± 0.43	86.35 ± 0.12	53.19 ± 0.22	-	-	87.04 ± 0.30	71.65 ± 0.45
	ContinualCRUST [33]	95.07 ± 0.99	75.96 ± 1.98	52.39 ± 0.66	-	-	75.18 ± 1.85	52.05 ± 3.95
	STAR [10]	96.22 ± 0.39	85.99 ± 0.21	66.32 ± 0.26	-	-	85.72 ± 2.76	60.28 ± 2.20
	WSC [51]	95.04 ± 1.15	83.18 ± 1.99	63.69 ± 0.35	-	-	80.24 ± 0.89	58.46 ± 2.00
	CNLDD	94.49 ± 0.23	89.17 ± 0.55	76.24 ± 0.78	-	-	88.11 ± 0.28	73.44 ± 1.21
FMoW	Finetune	55.81 ± 0.65	38.69 ± 1.03	33.35 ± 1.32	42.51 ± 1.55	33.80 ± 0.60	40.77 ± 1.18	31.56 ± 1.04
	ER [5]	59.53 ± 1.13	40.77 ± 1.39	34.91 ± 0.82	45.09 ± 0.71	35.01 ± 1.22	42.65 ± 1.31	32.56 ± 0.71
	Online EWC [23]	58.19 ± 1.43	39.15 ± 1.59	33.89 ± 1.31	42.79 ± 1.76	32.35 ± 0.72	39.39 ± 1.27	30.51 ± 1.11
	AdaStreams [50]	58.76 ± 0.93	55.16 ± 1.58	47.32 ± 1.08	53.19 ± 0.69	38.51 ± 0.24	48.53 ± 1.10	29.15 ± 8.23
	CNLL [20]	55.95 ± 2.09	48.26 ± 0.39	42.78 ± 1.36	53.15 ± 0.32	42.81 ± 0.25	49.12 ± 0.78	36.47 ± 1.11
	Meta-DR [41]	70.77 ± 0.91	51.63 ± 0.67	37.81 ± 0.78	53.69 ± 1.32	38.40 ± 1.21	52.36 ± 1.94	36.36 ± 1.44
	SPR [21]	46.50 ± 0.00	39.49 ± 0.00	31.89 ± 0.00	39.13 ± 1.43	29.48 ± 0.00	39.35 ± 0.00	25.75 ± 0.00
	PuriDivER [3]	62.52 ± 1.14	45.74 ± 0.65	36.05 ± 0.89	48.50 ± 1.13	36.99 ± 1.78	45.98 ± 3.56	33.13 ± 3.74
	Co-teaching [16]	51.95 ± 1.47	41.00 ± 1.15	31.72 ± 0.94	44.80 ± 2.23	32.92 ± 2.19	42.82 ± 0.68	30.88 ± 1.18
	ϵ-Softmax [42]	53.35 ± 2.17	49.94 ± 0.49	42.66 ± 0.68	48.41 ± 1.57	36.99 ± 0.89	46.98 ± 1.74	26.43 ± 1.05
	ContinualCRUST [33]	69.61 ± 2.16	54.00 ± 0.69	42.07 ± 0.59	57.47 ± 2.80	44.40 ± 1.29	55.84 ± 1.94	45.15 ± 0.64
	STAR [10]	74.56 ± 0.25	61.45 ± 1.37	45.88 ± 4.04	60.88 ± 0.35	44.38 ± 0.33	59.18 ± 4.77	41.48 ± 9.39
	WSC [51]	74.73 ± 0.33	63.94 ± 0.42	44.90 ± 1.34	62.17 ± 0.45	43.01 ± 0.69	64.46 ± 0.20	48.44 ± 0.49
	CNLDD	75.16 ± 0.28	67.42 ± 0.52	54.85 ± 0.81	64.84 ± 0.71	46.23 ± 0.75	65.22 ± 0.13	50.45 ± 1.50

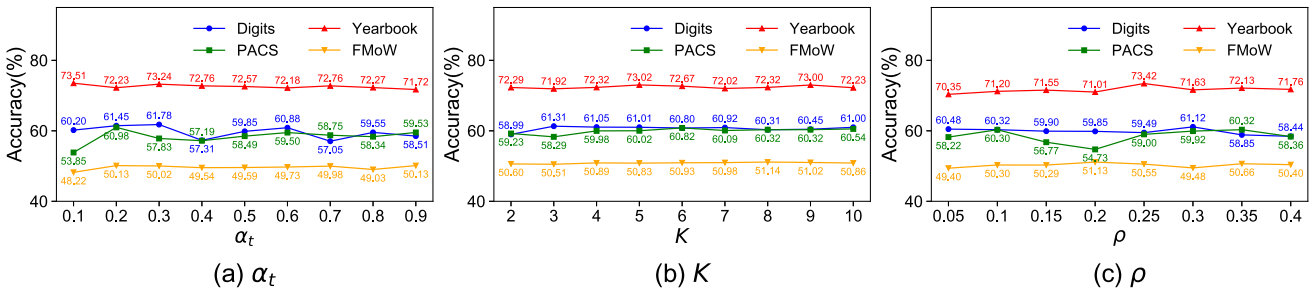

Fig. 4. Parametric sensitivity under different values of (a) α_t, (b) K, and (c) ρ. The experiments are conducted on *Digits*, *PACS*, *Yearbook*, and *FMoW* datasets.

The experimental results for the distribution combination coefficient α_t, the selection ratio ρ, and the number of clusters K are shown in Fig. 4. As shown in Fig. 4(a), small values of α_t generally result in satisfactory classification accuracies. For instance, on *Digits* dataset, when α_t is set to 0.2, the classification accuracy is 61.45% , and it improves to 61.78% when α_t is set to 0.3. On *PACS* dataset, a value of $\alpha_t = 0.2$ achieves an accuracy of 60.98% , and on *Yearbook* dataset, a smaller value of α_t also corresponds to relatively higher accuracy. Therefore, setting α_t within the range of (0.1,0.3) yields the encouraging results overall. Additionally, the evaluation of the number of clusters K in K-Means algorithm (Fig. 4(b)) reveals that moderate values of K, such as 5 or 6, generally yield better results. For example, on *PACS* dataset, $K = 6$ results in the highest accuracy of 60.82% . Lastly, Fig. 4(c) shows that the CNLDD algorithm shows sensitivity to the selection ratio ρ. The best performance is observed when $\rho \in \{0.20, 0.30\}$.

For the weight coefficients λ_1 and λ_2 in the objective function (i.e., (17)), the classification accuracies under various combinations are shown in Fig. 5. Here, the range of λ_1 is $\{0.1, 1.0, 10.0, 100.0\}$, and λ_2 is varied over $\{0.001, 0.01, 0.1, 1.0\}$. The experimental results reveal that the classification performance of CNLDD is not sensitive to the selection of (λ_1, λ_2). In general, large values of λ_1 and small values of λ_2 tend to yield consistently better results.

D. Ablation Study

In this section, we conduct ablative experiments to evaluate the contribution of each component in our CNLDD.

Our theoretical analysis in Section IV reveals three critical factors that lead to catastrophic forgetting, namely selection bias of buffered data, distribution shift, and label noise. Therefore, we examine the effectiveness of the three techniques introduced in Section V, each of which targets one of these challenging factors. Specifically, to evaluate the effectiveness of CNLDD in mitigating selection bias of buffered data, we replace the proposed two-step buffer update strategy (Section V-A) with

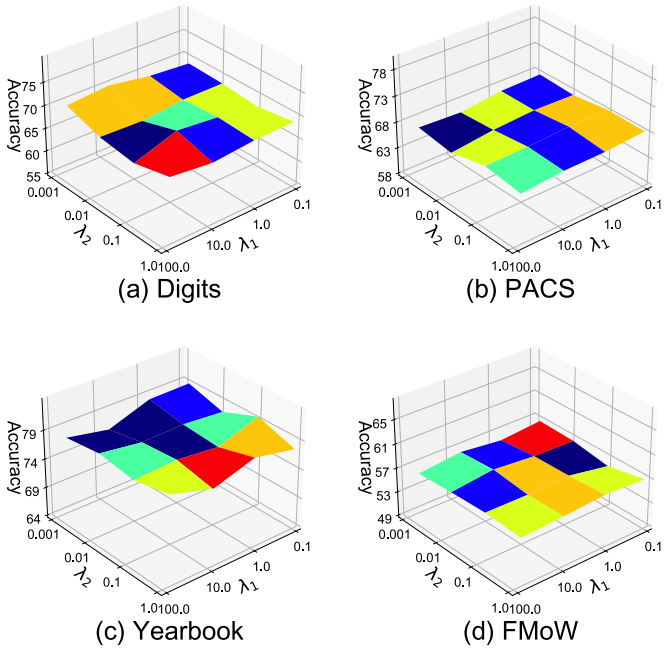

Fig. 5. Parametric sensitivity of the CNLDD method with different values of λ_1 and λ_2. The experiments are conducted on *Digits*, *PACS*, *Yearbook*, and *FMoW* datasets.

TABLE V
AVERAGE TEST ACCURACIES (%) UNDER THREE ABLATION SETTINGS, NAMELY: (I) W/O THE TWO-STEP BUFFER UPDATE STRATEGY, (II) W/O THE SELECTIVE TRANSFER MECHANISM, AND (III) W/O THE DENSITY RATIO REWEIGHTING. THE BEST RECORD ON EACH DATASET IS HIGHLIGHTED IN **BOLD**.

Setting	*Digits*	*PACS*	*Yearbook*	*FMoW*
(I)	57.14	57.09	71.10	44.11
(II)	50.85	54.88	72.09	49.60
(III)	48.29	51.68	70.80	46.18
CNLDD	**61.69**	**60.85**	**73.44**	**50.45**

the commonly employed reservoir sampling strategy [5], [40], [43]. Moreover, regarding the factor of distribution shift, we eliminate the discrepancy-based selective transfer mechanism (Section V-D) by setting the weight vector $\mathbf{q}$ to a uniform vector. Additionally, to evaluate the robustness of CNLDD against label noise, we remove the density ratios from the objective function, i.e., $\beta(\cdot,\cdot)$ in (17). The three ablation settings are respectively referred to as (I), (II), and (III) in Table V. Consistent with the experiments in Section VII-C, the noise setting utilized here is the most challenging case, namely "inst. 40% ".

The experimental results for three ablation settings are presented in Table V, which indicate that the absence of any of buffer update strategy, discrepancy-based selective transfer, and label noise handling will lead to performance degradation. Therefore, all components in our CNLDD method are indispensable, as they play crucial roles in alleviating catastrophic forgetting in continual noisy label learning.

VIII. CONCLUSION AND FUTURE WORK

In this paper, we theoretically analyze the problem of learning from streaming noisy data with distribution shift, and we derive an upper bound for the cumulative generalization error. This bound highlights critical factors that lead to catastrophic forgetting and affect the overall continual learning performance, namely buffered data selection bias, distribution shift, and label noise. Our theoretical findings directly induce the proposed method of Continual Noisy Label Learning on Drifting Data Streams (termed "CNLDD"). To address the above challenges, CNLDD contains a two-step buffer update strategy to reduce buffered data selection bias, a selective knowledge transfer technique to mitigate distribution shift, and a density ratio reweighting approach to handle instance-dependent label noise. Thanks to the unified modeling, CNLDD demonstrates superior performance when compared with various state-of-the-art label noise learning and continual learning approaches on standard benchmark and real-world datasets.

For future work, we plan to further develop our theoretical framework to investigate the upper bound of cumulative generalization error in class-incremental continual learning.

REFERENCES

[1] D. Arpit et al., "A closer look at memorization in deep networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 233–242.
[2] P. Awasthi, C. Cortes, and C. Mohri, "Theory and algorithm for batch distribution drift problems," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2023, pp. 9826–9851.
[3] J. Bang, H. Koh, S. Park, H. Song, J.-W. Ha, and J. Choi, "Online continual learning on a contaminated data stream with blurry task boundaries," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 9275–9284.
[4] A. Beck, *First-Order Methods in Optimization*. Philadelphia, PA, USA: SIAM, 2017.
[5] A. Chaudhry et al., "Continual learning with tiny episodic memories," in *Proc. ICML Workshop Multi-Task Lifelong Reinforcement Learn.*, 2019.
[6] H. Cheng, Z. Zhu, X. Li, Y. Gong, X. Sun, and Y. Liu, "Learning with instance-dependent label noise: A sample sieve approach," in *Proc. Int. Conf. Learn. Representations*, 2021.
[7] W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, and A. Schrijver, *Combinatorial Optimization*. Berlin, Germany: Springer, 1998.
[8] F. R. Cordeiro and G. Carneiro, "A survey on deep learning with noisy labels: How to train your model when you cannot trust on the annotations?," in *Proc. SIBGRAPI Conf. Graph. Patterns Images*, 2020, pp. 9–16.
[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.
[10] M. Eskandar, T. Imtiaz, D. Hill, Z. Wang, and J. Dy, "STAR: Stability-inducing weight perturbation for continual learning," in *Proc. Int. Conf. Learn. Representations*, 2025.
[11] M. Farajtabar, N. Azizan, A. Mott, and A. Li, "Orthogonal gradient descent for continual learning," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2020, pp. 3762–3773.
[12] G. M. van de Ven, T. Tuytelaars, and A. S. Tolias, "Three types of incremental learning," *Nature Mach. Intell.*, vol. 4, pp. 1185–1197, 2022.
[13] C. Gong et al., "Class-wise denoising for robust learning under label noise," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 2835–2848, Mar. 2023.
[14] S. L. Hakimi, "Optimum locations of switching centers and the absolute centers and medians of a graph," *Operations Res.*, vol. 12, no. 3, pp. 450–459, 1964.
[15] O. B. Halima, B. I. Adebimpe, and N. A. Maxwell, "Adaptive machine learning models: Concepts for real-time financial fraud prevention in dynamic environments," *World J. Adv. Eng. Technol. Sci.*, vol. 12, no. 2, pp. 21–34, 2024.

[16] B. Han et al., "Co-teaching: Robust training of deep neural networks with extremely noisy labels," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, vol. 31, pp. 8527–8537.
[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[18] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan, "AugMix: A simple data processing method to improve robustness and uncertainty," in *Proc. Int. Conf. Learn. Representations*, 2020.
[19] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural Computation*, vol. 3, no. 1, pp. 79–87, 1991.
[20] N. Karim, U. Khalid, A. Esmaeili, and N. Rahnavard, "CNLL: A semi-supervised approach for continual noisy label learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2022, pp. 3877–3887.
[21] C. D. Kim, J. Jeong, S. Moon, and G. Kim, "Continual learning on noisy data streams via self-purified replay," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 537–547.
[22] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2014.
[23] J. Kirkpatrick et al., "Overcoming catastrophic forgetting in neural networks," *Proc. Nat. Acad. Sci. USA*, vol. 114, no. 13, pp. 3521–3526, 2017.
[24] P. Kumari, J. Chauhan, A. Bozorgpour, B. Huang, R. Azad, and D. Merhof, "Continual learning in medical image analysis: A comprehensive review of recent advancements and future prospects," *Med. Image Anal.*, vol. 106, 2025, Art. no. 103730.
[25] D. Li, Y. Yang, Y.-Z. Song, and T. Hospedales, "Learning to generalize: Meta-learning for domain generalization," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 3490–3497.
[26] J. Li, M. Zhang, K. Xu, J. Dickerson, and J. Ba, "How does a neural network's architecture impact its robustness to noisy labels?" *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 9788–9803, 2021.
[27] J. Li, R. Socher, and S. C. Hoi, "DivideMix: Learning with noisy labels as semi-supervised learning," in *Proc. Int. Conf. Learn. Representations*, 2020.
[28] T. Liu and D. Tao, "Classification with noisy labels by importance reweighting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 3, pp. 447–461, Mar. 2016.
[29] X. Liu et al., "Generative feature replay for class-incremental learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 226–227.
[30] W. Luo et al., "Estimating per-class statistics for label noise learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 47, no. 1, pp. 305–322, Jan. 2025.
[31] Y. Mansour, M. Mohri, and A. Rostamizadeh, "Domain adaptation with multiple sources," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 1–8.
[32] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*. Cambridge, MA, USA: MIT Press, 2018.
[33] E. Mucllari, A. Raghavan, and Z. A. Daniels, "Noise-tolerant coreset-based class incremental continual learning," 2025, *arXiv:2504.16763*.
[34] P. Pan, S. Swaroop, A. Immer, R. Eschenhagen, R. Turner, and M. E. E. Khan, "Continual deep learning by functional regularisation of memorable past," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 4453–4464.
[35] O. Sener and S. Savarese, "Active learning for convolutional neural networks: A core-set approach," in *Proc. Int. Conf. Learn. Representations*, 2018.
[36] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2014.
[37] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 2994–3003.
[38] D. B. Shmoys, "Computing near-optimal solutions to combinatorial optimization problems," in *Combinatorial Optimization* (DIMACS Series in Discrete Mathematics and Theoretical Computer Science), vol. 20. Providence, RI, USA: Amer. Math. Soc., 1995, pp. 355–397.
[39] J. Tang et al., "Direct distillation between different domains," in *Proc. Eur. Conf. Comput. Vis.*, 2025, pp. 154–172.
[40] J. S. Vitter, "Random sampling with a reservoir," *ACM Trans. Math. Softw.*, vol. 11, no. 1, pp. 37–57, 1985.
[41] R. Volpi, D. Larlus, and G. Rogez, "Continual adaptation of visual representations via domain randomization and meta-learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 4443–4453.
[42] J. Wang, X. Zhou, D. Zhai, J. Jiang, X. Ji, and X. Liu, "ϵ-Softmax: Approximating one-hot vectors for mitigating label noise," in *Proc. Adv. Neural Inf. Process. Syst.*, 2024.
[43] L. Wang, X. Zhang, H. Su, and J. Zhu, "A comprehensive survey of continual learning: Theory, method and application," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 8, pp. 5362–5383, Aug. 2024.
[44] S. Wang, X. Li, J. Sun, and Z. Xu, "Training networks in null space of feature covariance for continual learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 184–193.
[45] X. Xia et al., "Combating noisy labels with sample selection by mining high-discrepancy examples," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2023, pp. 1833–1843.
[46] X. Xia et al., "Regularly truncated m-estimators for learning with noisy labels," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 5, pp. 3522–3536, May 2024.
[47] H. Yao, C. Choi, B. Cao, Y. Lee, P. W. W. Koh, and C. Finn, "Wild-time: A benchmark of in-the-wild distribution shift over time," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 10309–10324.
[48] J. Yoo, Y. Liu, F. Wood, and G. Pleiss, "Layerwise proximal replay: A proximal point method for online continual learning," in *Proc. Int. Conf. Mach. Learn.*, 2024.
[49] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3987–3995.
[50] Z.-Y. Zhang, Y.-Y. Qian, Y.-J. Zhang, Y. Jiang, and Z.-H. Zhou, "Adaptive learning for weakly labeled streams," in *Proc. Proc. ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 2022, pp. 2556–2564.
[51] Z.-H. Zhou, J.-J. Wang, T. Wei, and M.-L. Zhang, "Weakly-supervised contrastive learning for imprecise class labels," in *Proc. Int. Conf. Mach. Learn.*, 2025.
[52] Z. Zhu, Y. Song, and Y. Liu, "Clusterability as an alternative to anchor points when learning with noisy labels," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 12912–12923.

Wenshui Luo received the master's degree from the Nanjing University of Science and Technology, in 2025. He is currently working toward the PhD degree with Shanghai Jiao Tong University, under the supervision of Prof. Chen Gong. His research interests mainly include continual learning and weakly-supervised learning.

Shuo Chen received the Doctoral degree from the Nanjing University of Science and Technology, in 2020. From 2018 to 2019, he was a CSC visiting student with the University of Pittsburgh, USA. From 2020 to 2024, he was a postdoctoral researcher and research scientist with RIKEN-AIP. He is currently an associate professor with the School of Intelligence Science and Technology, Nanjing University China. He is also a visiting scientist with the RIKEN Center for Advanced Intelligence Project (RIKEN-AIP) Japan. He has authored or coauthored more than 50 technical papers with top-tier conferences such as NeurIPS, ICML, ICLR, and CVPR, and prominent journals such as *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *IEEE Transactions on Image Processing*, and *IEEE Transactions on Neural Networks and Learning Systems*. His research interests mainly include machine learning and pattern recognition, in particular, self-supervised learning and metric learning. He was the (senior) area chair of NeurIPS, ICML, ICLR, CVPR, ECCV, AAAI, and IJCAI more than 20 times, and was also the action editor for *Neural Network*. He was the recipient of the "Excellent Doctoral Dissertation Award" of Chinese Institute of Electronics (CIE) and the "Excellent Doctoral Dissertation Nomination" of Chinese Association for Artificial Intelligence (CAAI).

Tao Zhou (Senior Member, IEEE) received the PhD degree from Shanghai Jiao Tong University, in 2016.He was a postdoctoral fellow with UNC-CH and a research scientist with IIAI. He is currently a professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology, China. He is also an associate editor for *IEEE Transactions on Neural Networks and Learning Systems*, *IEEE Transactions on Image Processing*, *IEEE Transactions on Medical Imaging*, and *Pattern Recognition*. His research interests include machine learning, computer vision, AI in healthcare, and medical image analysis.

Chen Gong (Senior Member, IEEE) received the dual Doctoral degrees from Shanghai Jiao Tong University (SJTU) and from the University of Technology Sydney (UTS), respectively. He is currently a full professor with the School of Automation and Intelligent Sensing, Shanghai Jiao Tong University. He has authored or coauthored more than 130 technical papers at prominent journals and conferences such as JMLR, IJCV, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *IEEE Transactions on Neural Networks and Learning Systems*, *IEEE Transactions on Image Processing*, ICML, NeurIPS, ICLR, CVPR, ICCV, ECCV, AAAI, and IJCAI. His research interests mainly include machine learning, data mining, and learning-based vision problems. He is also the associate editor for *IEEE Transactions on Circuits and Systems for Video Technology*, *Neural Networks*, and NePL, and also the area chair or senior PC member of several top-tier conferences such as ICML, ICLR, AAAI, IJCAI, ECML-PKDD, AISTATS, ICDM, and ACM MM. He was the recipient of the Excellent Doctorial Dissertation Award of Chinese Association for Artificial Intelligence, Young Elite Scientists Sponsorship Program of China Association for Science and Technology, and Wu Wen-Jun AI Excellent Youth Scholar Award. He was also selected as the Global Top Chinese Young Scholars in AI released by Baidu, and "World's Top 2% Scientists" released by Stanford University.