An Adaptive Method for Weak Supervision with Drifting Data

Alessio Mazzetto **Reza Esfandiarpoor** **Akash Singirikonda** **Eli Upfal** **Stephen Bach**

Brown University

Abstract

We introduce an adaptive method with formal quality guarantees for weak supervision in a non-stationary setting. Our goal is to infer the unknown labels of a sequence of data by using weak supervision sources that provide independent noisy signals of the correct classification for each data point. This setting includes crowdsourcing and programmatic weak supervision. We focus on the non-stationary case, where the accuracy of the weak supervision sources can drift over time, e.g., because of changes in the underlying data distribution. Due to the drift, older data could provide misleading information to infer the label of the current data point. Previous work relied on a priori assumptions on the magnitude of the drift to decide how much data to use from the past. In contrast, our algorithm does not require any assumptions on the drift, and it adapts based on the input by dynamically varying its window size. In particular, at each step, our algorithm estimates the current accuracies of the weak supervision sources by identifying a window of past observations that guarantees a near-optimal minimization of the trade-off between the error due to the variance of the estimation and the error due to the drift. Experiments on synthetic and real-world labelers show that our approach adapts to the drift.

1 Introduction

In order to efficiently create training data for machine learning, programmatic weak supervision (Ratner et al., 2016, 2017; Zhang et al., 2022) estimates the accuracy of multiple noisy sources of labels without access to ground truth. Given a set of *labeling functions* that vote on the true label for each unlabeled example, the goal is to infer the latent ground truth. Once inferred, these labels can be used as training data. In this paper, we study the *non-stationary* setting, in which the accuracy of each labeling function can drift over time because of changes in the underlying data. For example, in an image classification task, latent subclasses that make up each class might shift over time. If the task is to classify animals into categories like "mammal" and "bird," the accuracy of a weak labeler that looks for attributes like wings might change in accuracy if animals like bats become more or less prevalent. We ask the question, "Under what conditions can we detect changes in the accuracies of weak labelers over time and bound their error without access to ground truth?"

Programmatic weak supervision is important for creating training data sets when resources are limited. It can be used for natural language processing (Safranchik et al., 2020; Yu et al., 2021; Zhang et al., 2021), computer vision (Varma et al., 2017; Chen et al., 2019a; Fu et al., 2020), tabular data (Chatterjee et al., 2020; Arachie and Huang, 2021), and other modalities (Sala et al., 2019; Shin et al., 2022). It has also enabled machine learning applications in industry (Bach et al., 2019; Bringer et al., 2019; Suri et al., 2020) and academia (Callahan et al., 2019; Fries et al., 2021). Even when prompting or fine-tuning large pre-trained models, weak supervision can unlock improved quality and enable adaptation to new tasks (Smith et al., 2022; Arora et al., 2022; Yu and Bach, 2023).

The central modeling challenge in programmatic weak supervision is estimating the probabilistic relationships among the votes of the weak labelers and the latent

Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258. Copyright 2025 by the author(s).

ground truth. It is hard because, without access to ground truth labels, the observed votes can be explained in many different ways. Perhaps the votes tend to agree because they are all accurate labelers. Or perhaps they are all inaccurate. Perhaps there are correlations among the votes caused by relying on similar decision processes. If one assumes that the votes are conditionally independent given the true label and that the examples are *independent*, and *identically distributed* (i.i.d.), this is equivalent to the Dawid-Skene model (Dawid and Skene, 1979) that is the basis for many related works in crowdsourcing (Raykar et al., 2010; Liu et al., 2012; Parisi et al., 2014; Joglekar et al., 2015; Zhang et al., 2016). Many works on crowdsourcing and weak supervision have relaxed the conditional independence assumption in various ways to account for a wide range of weak labelers (Balsubramani and Freund, 2015a,b, 2016; Bach et al., 2017; Varma et al., 2019; Arachie and Huang, 2019; Mazzetto et al., 2021b,a; Arachie and Huang, 2021).

With two exceptions discussed below, all these aforementioned works assume that the examples are drawn independently from a fixed distribution. This is a restrictive assumption when data is collected over time, and it is natural to observe a change, or *drift*, in the distribution of the examples. In our work, we relax the identically distributed assumption, and assume only that the examples are independent. This introduces a trade-off: to obtain a good estimate at the current time, using more past examples provides more data, which might result in a better estimate if that data is similarly distributed, but might harm the estimate if the window includes a significant distribution drift.

Much prior work has addressed the problem of drifting data in the supervised learning setting (Gama et al., 2014; Lu et al., 2018). These methods generally rely on labeled data that is unavailable in the weakly supervised setting. Another broad line of work has viewed drift detection as an unsupervised problem, looking for non-stationarity in arbitrary distributions (Barry and Hartigan, 1993; Killick et al., 2012; Truong et al., 2020). These methods generally assume a prior distribution on the locations in time of drift. That prior can be either defined explicitly in a Bayesian framework or implicitly via a heuristic cost function that penalizes the trade-off between better fitting the data and finding more drift points. In a similar vein, previous works on relaxing the i.i.d. assumption with multiple noisy labelers have placed assumptions on how much their accuracies can drift (Bonald and Combes, 2016; Fu et al., 2020). In contrast, our goal is to estimate the labelers' accuracies, without prior assumption on the drift. The lack of any assumptions means that each individual sample can come from its own distribution. This is a very challenging problem, as the drift is unknown and we cannot estimate the drift from the data, since we have access to only a single sample from each distribution.

Our Contributions. We introduce the first *adaptive* algorithm for programmatic weak supervision in the presence of drift with formal guarantees on the quality of its parameter estimates. The advantage of an adaptive algorithm is that it can react in a rigorously principled way to changes in the accuracies of the weak labelers as they occur (as opposed to having to make an assumption on how much drift will occur). When the underlying process is stationary, it can accumulate as much data as possible by using a large window of time in order to best estimate the accuracies of the labelers. When drift does occur, it can minimize the drift error by using a smaller window with the most recent (and most relevant) data to estimate the accuracies.

Our method selects the amount of data to use based on differences in the rates of agreement among the labelers. We derive a principled decision rule for this selection and provide a rigorous analysis that bounds the resulting error of the estimated accuracies of the labelers. Our novel bound separates the statistical error of estimating the parameters from the error caused by possible drift. This analysis enables the algorithm to select a close-to-optimal trade-off to minimize the worst-case error.

The conceptual difference between our approach and all previous work on weak supervision with drifting data is that we do not rely on prior information about the drift, or try to learn the drift from the data (both unrealistic in many applications). Instead, at each time step, our algorithm compares its estimation obtained using different window sizes and uses this information to detect drift and adjust the window size for the decision at that step. We analytically prove that this information is sufficient to allow the algorithm to efficiently adapt to drift in distribution, without explicitly estimating the magnitude of the drift.

We demonstrate the functionality and the advantage of our algorithm over fixed-window-size strategies in several experimental settings, including synthetic data, image recognition, and video classification. The results show that our algorithm adapts to the drift as it occurs, dynamically selecting the amount of data (window size) to use in an effective way. Unlike fixed-window-size strategies, our approach consistently maintains high accuracy as it adapts to the changing drift.

2 Problem Statement

Given a vector $\boldsymbol{v} \in \mathbb{R}^q$, let $\|\boldsymbol{v}\|_\infty = \max_{1\leq i\leq q} |v_i|$. Similarly, given a matrix $\boldsymbol{C} \in \mathbb{R}^{q\times q}$, we define $\|\boldsymbol{C}\|_\infty =$

$\max_{i,j} |C_{ij}|$. A binary classification task is specified by a function $y : \mathcal{X} \mapsto \mathcal{Y}$, where $\mathcal{X} = \mathbb{R}^d$ is the classification domain and $\mathcal{Y} = \{-1, 1\}$ is the label space. Given x, we would like to infer its label $y(x)$. We assume access to n weak labeling functions $\ell_1, \dots, \ell_n$, where each $\ell_i : \mathcal{X} \to \mathcal{Y}$ provides a tentative labeling of the item x. For example, each weak labeling function can be a classifier that was trained for a related task, or a decision rule based on a simple programmatic criterion. The weak labeling functions $\ell_1(x), \dots, \ell_n(x)$ are the only information sources for the labels in our classification task.

We receive a sequence of examples $X_1, X_2, \dots$ over time. For any given time t, our goal is to obtain an accurate estimate of the correct label $y(X_t)$ of X_t given the weak labelling functions $\ell_1, \dots, \ell_n$ and the input sequence up to time t, $X_1, \dots, X_t$.

We adapt the standard assumptions used in analyzing weak supervision, in particular crowdsourcing, with no drift (Dawid and Skene, 1979; Ratner et al., 2017). We first assume that the input sequence $(X_t)_{t \in \mathbb{N}}$ is an independent, but not identically distributed stochastic process. Any finite subset of its random variables are mutually independent, and each X_t is sampled from a distribution D_t over $\mathcal{X}$ that can drift over time. Formally, this is stated with the following assumption.

Assumption 2.1. For any finite $t \geq 1$, the input vector $(X_1, \dots, X_t)$ is distributed as $\prod_{i=1}^{t} D_i$.

The second assumption is that the weak labelers have independent errors conditioned on the true label.

Assumption 2.2. For any $t \geq 1$ and $i \neq j$, for $X_t \sim \mathcal{D}_t$, we have that the events $\{\ell_i(X_t) \neq y(X_t)\}$ and $\{\ell_j(X_t) \neq y(X_t)\}$ are independent given $y(X_t)$.

This conditional independence is widely adopted across domains such as programmatic weak supervision and crowdsourcing (Dawid and Skene, 1979; Liu et al., 2012; Parisi et al., 2014; Joglekar et al., 2015; Zhang et al., 2016), and methods leveraging this assumption consistently demonstrate robust performance in practical applications (Ratner et al., 2017). Recent research has explored alternative, weaker assumptions, such as access to a limited amount of labeled data (Arachie and Huang, 2019, 2021; Mazzetto et al., 2021b,a), or parametric assumptions on the joint distributions of the true labels and outputs from weak labelers (Fu et al., 2020; Ratner et al., 2019). In Appendix C, we detail how our method can be extended to incorporate one of these alternative assumptions.

We let the accuracy of the weak labeler i at time t be

$$p_i(t) \doteq \Pr_{X \sim D_t} \left(\ell_i(X) = y(X) \right) \quad . \tag{1}$$

The value $p_i(t) \in [0, 1]$ is the probability that the weak labeler ℓ_i is correct with a sample $X_t \sim D_t$. The accuracy probability $p_i(t)$ is a function of the input distribution D_t and therefore may drift in time. We let $\boldsymbol{p}(t) = (p_1(t), \dots, p_n(t))$.

Example. Assume that the classification task is to distinguish whether an input image contains a cat or a dog. Let ℓ_{tail} be a weak labeler that detects whether an animal has a tail or not. This weak labeler provides no signal if we only observe images of cats and dogs that both have tails, however, the relevance of this classifier can change over time: if the probability of observing a dog without a tail (e.g., a bulldog) grows over time, this weak labeler can provide a stronger signal towards the right classification. Our goal is to adapt dynamically to the change in accuracy of the weak labelers.

Sources of Drift. For concreteness, we analyze our algorithm with respect to a drift in the input distribution over $\mathcal{X}$ (also referred to as *covariate shift*). However, our analysis applies to a more general case, since it only relies on the variation of the accuracy of the weak labelers $p_i(t)$, and is agnostic to the underlying cause of this variation.

As an example, in addition to drift in the input distribution, we can also allow a change in the functionality of the labeling functions. For example, a human labeler can get tired and make more mistakes, or a sensor's accuracy can be affected by a change of light or temperature. Formally, instead of a labelling function $\ell_i(X)$ we have a family of labelling functions $\{\ell_{i,t}(X) \mid t \geq 1\}$. Equation (1) is replaced with $p_i(t) \doteq \Pr_{X \sim D_t} \left(\ell_{i,t}(X) = y(X) \right)$, and the algorithm and analysis are the same.

In a similar manner, our analysis can be extended to the setting where the binary classification task $y(x)$ also changes over time (referred to as *concept drift*), by replacing $y(x)$ with a time-dependent function $y_t(x)$. This extension requires modifying Assumption 2.2 and Equation (1) accordingly, but it does not change our algorithm or analysis.

3 Related Work

To our knowledge, only two prior works have considered relaxations of the identical distribution (no drift) assumption in learning from multiple noisy sources of labels. They both require assumptions on how much the accuracies of the labelers can change over time. The first (Bonald and Combes, 2016) assumes that the accuracy of the weak labelers can change at most by a constant at each step, i.e., there exists $\Delta > 0$, known a priori, such that $\|\boldsymbol{p}(t) - \boldsymbol{p}(t+1)\|_\infty \leq \Delta$ for all $t \geq 1$. The second (Fu et al., 2020) assumes that the

KL divergence between two consecutive distributions is upper bounded by a constant Δ. These are similar assumptions: an upper bound on the magnitude of the drift allows these methods to determine before execution how much information to use from the past.

These algorithms are impractical as the value Δ is unknown in practice, and they cannot adapt to changes in the rate of drift over time. If the algorithm overestimates the drift, then it will use a smaller amount of data than it should, resulting in a greater statistical error in its estimates of the labelers' accuracies. If it underestimates the drift, then the algorithm will use too much data and incur a large error due to the drift. In contrast, in this work, our goal is to dynamically choose the window size as a function of the observed votes without requiring any prior assumptions on the magnitude of the drift. In other words, our approach is to *adapt* to the drift as it occurs. The adaptivity to the drift is important to capture the changes in the data distributions. As we will see in Section 6.1, algorithms that rely on a fixed choice of Δ are severely limited when the rate of drift itself is changing.

There is a vast literature that addresses the challenges of coping with non-stationary data in numerous different settings. In our work, we focus on a drift setting where we observe data from a non-stationary distribution, and we have access to a single sample at each time step. Within this drift setting, a relevant sequence of works (Bartlett, 1992; Long, 1998; Mohri and Muñoz Medina, 2012; Hanneke and Yang, 2019)) has studied the supervised learning problem, assuming some known upper bound on the drift. The minimax error for density estimation with distribution drift was studied by Mazzetto and Upfal (2023b), again with some a priori assumption on the drift rate. Recent work provides an adaptive algorithm for agnostic learning of a family of functions with distribution drift (Mazzetto and Upfal, 2023a), and analogous results were proven for discrete distribution estimation (Mazzetto and Upfal, 2023b; Mazzetto, 2024), model selection Han et al. (2024), and vector quantization (Mazzetto et al., 2025). While these works study similar drift settings, their results do not directly apply to our weak supervision setting. Our work is the first to provide an adaptive algorithm for weak supervision and crowdsourcing in a non-stationary setting and without any prior assumptions on the drift.

A process that reveals just one sample per step is more natural but harder to handle than processes that reveal a set of inputs in each step (e.g., Bai et al., 2022; Zhang et al., 2022). In the latter scenario, an algorithm has multiple samples from each distribution to learn the drift error (Mohri and Muñoz Medina, 2012; Awasthi et al., 2023). We also note that the non-stationary setting was also extensively studied in reinforcement learning (e.g., Auer et al., 2019; Chen et al., 2019b; Wei and Luo, 2021). That setting significantly differs from ours, as the goal is to minimize the regret, and the distribution of the samples is also affected by the decisions taken by a policy on the environment.

Finally, we would like to remark that there is a vast literature on the drift detection problem in both learning and data mining (e.g., Gama et al., 2014; Lu et al., 2018; Agrahari and Singh, 2022; Yu et al., 2022; Li et al., 2022; Yu et al., 2023). Unlike these works, our goal is to provably decide over a window of data depending on the drift within that window.

4 Preliminary Results

Our work builds on the following results that study the problem in settings where the accuracy probabilities are known, or there is no drift in the input distribution.

Assume first that the accuracy $\boldsymbol{p}(t) = (p_1(t), \ldots, p_n(t))$ of the weak labelers at any time $t \geq 1$ are known. With Assumption 2.2, it is known that the optimal aggregation rule for classifying X_t is a weighted majority vote of $\ell_1(X_t), \ldots, \ell_n(X_t)$, where the weights are a function of $\boldsymbol{p}(t)$ (Nitzan and Paroush, 1982). In particular, consider the family of weighted majority classifiers $f_{\boldsymbol{w}} : \mathcal{X} \mapsto \mathcal{Y}$ with weights $\boldsymbol{w} = (w_1, \ldots, w_n)$, i.e., $f_{\boldsymbol{w}}(x) = \text{sign}\left(\sum_{i=1}^{n} w_i \ell_i(x)\right)$. The optimal aggregation of $\ell_1(X_t), \ldots, \ell_n(X_t)$ is given by $f_{\boldsymbol{w}^*(t)}$ where

$$\boldsymbol{w}^*(t) = \left(\ln\left(\frac{p_1(t)}{1-p_1(t)}\right), \ldots, \ln\left(\frac{p_n(t)}{1-p_n(t)}\right)\right). \quad (2)$$

The above result implies that under Assumption 2.2, the knowledge of the weak labelers' accuracies is sufficient to obtain the optimal aggregation rule. In weak supervision and crowdsourcing applications, the accuracy probabilities of the weak labelers are unknown. Several methods for estimating $\boldsymbol{p}(t)$ using previous samples, have been proposed in the literature in a setting without distribution drift (Dawid and Skene, 1979; Ghosh et al., 2011; Zhang et al., 2016). It is known that under mild assumptions, if we have access to enough identically distributed samples, it is possible to accurately estimate the accuracies of the weak labelers, and different minimax optimal methods have been proposed in this setting (Zhang et al., 2016; Bonald and Combes, 2017). Our contribution is an adaptive method that allows for this estimation in a non-stationary setting without any prior assumption on the drift.

Our estimation method is based on the technique developed by Bonald and Combes (2017) that uses the weak labelers' *correlation matrix* to estimate the expertise of each weak labeler in a no-drift setting. In particular, for each $t \geq 1$, we let the correlation matrix $\boldsymbol{C}(t) \in [-1,1]^{n\times n}$ be defined as $C_{ij}(t) = \mathbb{E}_{X\sim D_t}\left[\ell_i(X)\ell_j(X)\right]$ for all $(i,j) \in \{1,\dots,n\}^2$ When there is no distribution drift and under mild assumptions on the bias of the estimates of the weak supervision sources, it is possible to show that a good estimation of the correlation matrix $\boldsymbol{C}(t)$ implies a good estimation of the accuracies $\boldsymbol{p}(t)$. The assumption on the bias is formalized as follows.

Assumption 4.1. There exists $\tau > 0$ such that $p_i(t) \geq \frac{1}{2} + \tau$ for all $t \geq 1$ and $i \in \{1,\dots,n\}$.

With this assumption, the following result holds.

Proposition 4.2 (Lemma 9 of Bonald and Combes (2017))**.** *Let $\boldsymbol{C} \in [-1,1]^{n\times n}$ be a matrix such that $\|\boldsymbol{C} - \boldsymbol{C}(t)\|_\infty \leq \epsilon$, and assume $n \geq 3$. Let Assumptions 2.1, 2.2 and 4.1 hold. Then, there exists an estimation procedure that given in input $\boldsymbol{C}$, it outputs $\hat{\boldsymbol{p}} = (\hat{p}_1,\dots,\hat{p}_n)$ such that $\|\boldsymbol{p}(t) - \hat{\boldsymbol{p}}\|_\infty \leq (5/2)\epsilon/\tau^2$.*

The intuition behind the result of Proposition 4.2 is that for a time t, for $i \neq j$, the entry $C_{ij}(t)$ is proportional to how much the weak labelers ℓ_i and ℓ_j agree, and by using Assumption 2.2 on the conditional independence of the error of the weak labelers, it can be written as a function of $\boldsymbol{p}(t)$ since $C_{ij}(t) = (2p_i(t)-1)(2p_j(t)-1)$. The proposition demonstrates that it is feasible to retrieve $\boldsymbol{p}(t)$ by using an estimate of the values $C_{ij}(t)$ for $i \neq j$. Note that both vectors $\boldsymbol{p}(t)$ and $\boldsymbol{1} - \boldsymbol{p}(t)$ would satisfy the constraints given by $C_{ij}(t)$ for $i \neq j$. Assumption 4.1 is used to differentiate between those two symmetrical scenarios where two weak labelers are more inclined to agree when they are both likely to be correct or both likely to be incorrect. The algorithm for the non-drift case presented by Bonald and Combes (2017), and the algorithm presented here for the drift case are oblivious to the value of τ.

5 Algorithm

As explained in the previous section, our method revolves around the estimation of the correlation matrix $\boldsymbol{C}(t)$ at the current time t in order to use Proposition 4.2 and obtain an estimate of $\boldsymbol{p}(t)$. All proofs are deferred to the supplementary material. We define $\hat{\boldsymbol{C}}^{[r]}(t) \in [-1,1]^{n\times n}$ as $\hat{\boldsymbol{C}}^{[r]}(t) \doteq \frac{1}{r}\sum_{k=t-r+1}^{t}(\ell_1(X_k),\dots,\ell_n(X_k))^T(\ell_1(X_k),\dots,\ell_n(X_k))$ The matrix $\hat{\boldsymbol{C}}^{[r]}(t) \in [-1,1]^{n\times n}$ is the empirical correlation matrix computed using the latest r samples $X_{t-r+1},\dots,X_t$. This matrix provides the following guarantee on the estimation of $\boldsymbol{C}(t)$.

Lemma 5.1. *Let $t \geq 1$, let $\delta \in (0,1)$, and let Assumption 2.2 hold. The following inequality holds with probability at least $1-\delta$:*

$$\|\boldsymbol{C}(t) - \hat{\boldsymbol{C}}^{[r]}(t)\|_\infty \leq \sqrt{\frac{2\ln(n(n-1)/\delta)}{r}} + 12\sum_{k=t-r+1}^{t-1}\|\boldsymbol{p}(k) - \boldsymbol{p}(k+1)\|_\infty \qquad (3)$$

Lemma 5.1 shows that the error of estimating $\boldsymbol{C}(t)$ by using the previous r samples can be upper bounded with the sum of two error terms: a *statistical error* and a *drift error*. The statistical error is related to the *sample complexity* of the estimation: it is due to the variance of the estimator $\hat{C}^{[r]}(t)$, and it decays with rate $O(1/\sqrt{r})$. The drift error is unknown and it quantifies the error introduced due to the distribution shift, and it is measured as the sum of the maximum variation of the accuracy of the weak labelers at each step. The drift error is non-decreasing with respect to r. There is a trade-off: we want to choose r to minimize the sum of the statistical error and the drift error.

We remark that adapting to drifting data adds an additional constraint to the sample complexity of the solution. When the samples are drawn from a fixed distribution, a larger sample size reduces the statistical error of the estimates. When the samples are drawn from a drifting distribution, a larger sample size still reduces the statistical error. However, a larger sample also increases the drift error since it uses older samples. Thus, even if the algorithm receives a very long input sequence, the drift poses an upper bound on the sample size that can be used efficiently at any given step.

Our main contribution is an algorithm that *without* any assumption on the drift can provide a close-to-optimal solution of the above trade-off (3). This is a challenging problem, as it is not possible to estimate the drift error, since we only have a single sample from each distribution. Nonetheless, our algorithm guarantees an estimation error of the matrix $\boldsymbol{C}(t)$ that is essentially up to constant as tight as the value of r that minimizes the right-hand side of (3). This yields a guarantee on the estimation of $\boldsymbol{p}(t)$ by using Proposition 4.2.

The pseudocode of our method is reported in Algorithm 1. Our algorithm has the following parameters (a more detailed discussion is deferred to Appendix A.3):

- $\delta \in (0,1)$ is the failure probability of the algorithm for the estimation at a given time t.
- $m \in \mathbb{N}$ is the maximum number of window sizes evaluated by our algorithm.
- The parameter $\beta > 0$ affects the threshold used in our algorithm, and it controls the sensitivity of our algorithm to changes in drift. We empirically show that any value $\beta \in (0, 0.1)$ provides a good estimation (Appendix A.3).

The next theorem gives our algorithm's error guarantee.

Theorem 5.2 (Main Result)**.** *Let Assumptions 2.1, 2.2 and 4.1 hold. Let $\delta \in (0,1)$, $\beta > 0$, and $m \in \mathbb{N}$. Assume the number of weak labelers is $n \geq 3$. Fix a time $t \geq 1$. There exist an algorithm that outputs $\hat{\boldsymbol{p}} = (\hat{p}_1, \dots, \hat{p}_n)$ such that with probability at least $1-\delta$:*

$$\|\boldsymbol{p}(t) - \hat{\boldsymbol{p}}\|_\infty = O\left(\frac{\beta + \beta^{-1}}{\tau^2} \min_{1 \leq r \leq \min(t, 2^m)} \left(\sqrt{\frac{\ln(nm/\delta)}{r}} + \sum_{k=t-r+1}^{r-1} \|\boldsymbol{p}(k) - \boldsymbol{p}(k+1)\|_\infty\right)\right),$$

where τ is defined as in Assumption 4.1.

A statement of the theorem with the exact constants is provided in Appendix B. The algorithm evaluates window sizes $\mathcal{R} = \{r_1 = 2^0, \dots, r_m = 2^{m-1}\}$. Our method increases the window size ending at time t as long as it does not include significant drift in distribution. As a reference for making this decision we observe that if the samples are identically distributed, then the estimated correlation matrices $\hat{\boldsymbol{C}}^{[r_{k+1}]}$ and $\hat{\boldsymbol{C}}^{[r_k]}$ should be similar up to the statistical error that is proportional to $O(1/\sqrt{r_k})$. The strategy of our algorithm is based on this intuition. Starting with $k = 1$, we iteratively compare the empirical covariance matrix computed respectively with r_k and r_{k+1} samples. If there is minimal drift, the empirical quantity $\|\hat{\boldsymbol{C}}^{[r_{k+1}]} - \hat{\boldsymbol{C}}^{[r_k]}_{i,j}\|\|_\infty$ should be comparable to the statistical error due to using r_k samples. If that is the case, we increase the value of k. If this empirical quantity is larger, then a significant drift must have occurred. In this case, we can stop and show that using r_k samples is provably close to optimal. This strategy is implemented in lines 2–7. The *threshold* used as a terminating condition for the iteration of the algorithm is the right-hand side of line 5. The lines 8–12 implement the method that maps a correlation matrix to the accuracies of the weak supervision sources and attains the guarantees of Proposition 4.2 (Bonald and Combes, 2017).

Computation and Memory. We optimize the memory and computational efficiency of the algorithm in an online setting by storing only the most recent $r_m = 2^{m-1}$ data points at each step where m is the maximum number of window sizes considered by our algorithm. The value of m provides a trade-off between the quality of estimation and the use of memory and computational resources. A larger maximum window size can yield a better estimation when the data is stationary. However, it is sufficient to use a small value of m: for any $\epsilon > 0$, if we set $m = O(\log(1/\epsilon))$, then we incur an additional additive estimation error ϵ compared to an algorithm that has access to all the past samples. Also, the empirical covariance matrices can be maintained efficiently: for any window size $r \in \mathcal{R}$, the matrix $\hat{C}^{[r]}(t)$ differs from $\hat{C}^{[r]}(t-1)$ by only two samples, thus it can be updated in time $O(n^2)$.

Algorithm 1 Non-Stationary Accuracy Estimation

```
1: Input: $(X_i)_{i=1}^t, (\ell_i)_{i=1}^n, \mathcal{R} = \{r_1, \dots, r_m\}, \beta, \delta$.
2: $A_{n,m,\delta} \leftarrow \sqrt{2\ln[(2m-1) \cdot n(n-1)/\delta]}$
3: $k \leftarrow 1$
4: while $(k \leq m-1)$ and $(r_{k+1} \leq t)$ do
5:     if $\|\hat{\boldsymbol{C}}^{[r_{k+1}]}(t) - \hat{\boldsymbol{C}}^{[r_k]}(t)\|_\infty \leq A_{n,m,\delta}\left[\frac{2\beta}{\sqrt{r_k}} + \sqrt{\frac{1-\frac{r_k}{r_{k+1}}}{r_k}}\right]$ then $k \leftarrow k+1$
6:     else break
7: $\hat{\boldsymbol{C}} \leftarrow \hat{\boldsymbol{C}}^{[r_k]}(t)$
8: for all $h \in \{1, \dots, n\}$ do
9:     $(i,j) \leftarrow \text{argmax}_{i \neq j, j \neq h, i \neq h} \left|\hat{C}_{ij}\right|$
10:    if $\hat{C}_{ij} = 0$ then $\hat{p}_h \leftarrow 1/2$
11:    else $\hat{p}_h \leftarrow \left(1 + \sqrt{\left|\frac{\hat{C}_{ih}\hat{C}_{hj}}{\hat{C}_{ij}}\right|}\right)/2$
12: return $\hat{p} = (\hat{p}_1, \dots, \hat{p}_n)$
```

Failure Probability. The theorem guarantees that the estimation is correct (i.e., it satisfies the upper bound of the statement) at any fixed time step t with probability at least $1-\delta$, where $\delta > 0$ can be set arbitrarily small. Thus, over a horizon $T \geq 0$, the algorithm is expected to fail in at most $T\delta$ steps. It is important to note that the algorithm is guaranteed to recover from failures. In particular, since the maximum window size is r_m, the event of giving a wrong estimation at time $t + r_m$ is independent of the outcomes of all events before time $t+1$.

6 Empirical Evaluation

We demonstrate the functionality and the advantage of our algorithm over fixed-window-size strategies in several experimental settings, including synthetic data, image recognition, and video classification. Due to space constraints, we present only part of the experimental results in this section. Additional experiments and details are reported in the supplementary material.

Setup. At each time step, we receive an unlabeled example which must be labeled based on the available weak labelers. We use Algorithm 1 to estimate the accuracies of the weak labelers. We then make a prediction for the current time step's example by weighting the vote of each labeler proportionally to its estimated accuracy using the weighting $\boldsymbol{w}^*$ described in Equation (2). For all experiments, we run our algorithm with $m = 20$, $\mathcal{R} = \{2^0, 2^1, \dots, 2^{19}\}$, and $\beta = \delta = 0.1$ (see Appendix A.3). We compare with the following

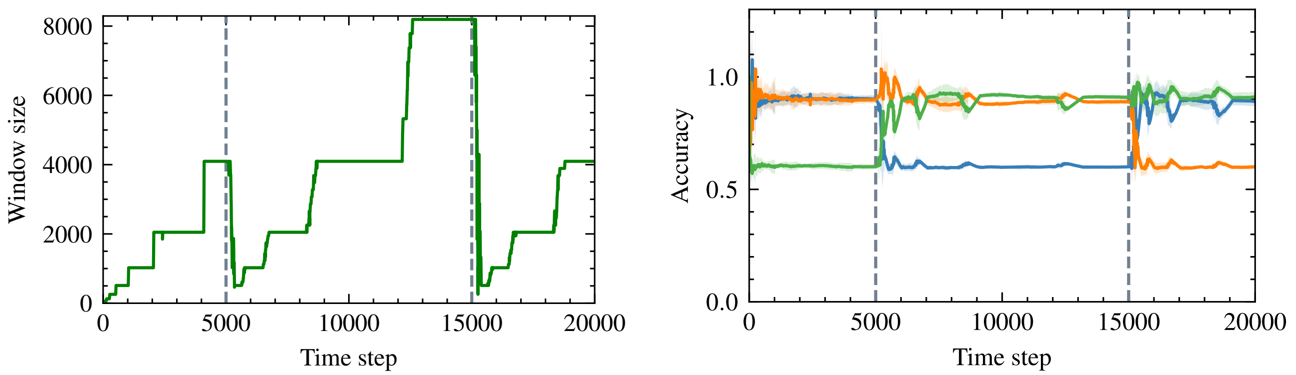

Figure 1: We report the window size chosen by the algorithm (left) and the estimated accuracies of each weak labeler over time (right). The vertical lines represent when a change in distribution occurs.

baselines algorithms: (1) a majority vote of the weak labelers at this step; and (2) non-adaptive fixed-window-size algorithms, one for each size in $\mathcal{R}$. The latter algorithms are the same as Algorithm 1, except that we use a fixed size window to estimate $\hat{\boldsymbol{C}}$ (line 7). Since the triplet method for estimating accuracies (lines 8-12) is not constrained to return a probability between 0 and 1, we clip the estimated accuracies to the interval $[0.1, 0.9]$. The fixed-window-size algorithms are analogous to previous work on crowdsourcing with drift that also determines a priori how much information to use from the past, depending on an assumption that constraints the magnitude of the drift. In practice, it is not possible to run those algorithms from the previous work, since we cannot estimate the drift as we have access to only a single sample from each distribution. The code for the experiments is available online [1].

6.1 Synthetic Data

We first show how our algorithm adapts to changing input distributions with an artificial experiment on synthetic data that satisfies all of our assumptions. The algorithm receives input from three weak labelers ($n = 3$), and the input stream has $4 \cdot T$ data points with $T = 5000$. The data is partitioned into three contiguous blocks of size T, $2T$, and T. The accuracies of the weak labelers do not change within the same block, but do change between blocks. In particular, for each block, two weak labelers have a high accuracy equal to 0.9, and the other one has a low accuracy equal to 0.6. The weak labeler with low accuracy is different in each block. We remark that our algorithm is oblivious to this partitioning of the data.

In Figure 1, we plot the window size used by the adaptive algorithm and its estimates of the accuracies of each weak labeler in each time t, $1 \leq t \leq 4T$. The reported results are an average over 10 independent generations of this synthetic data. The main observation is that our algorithm correctly identifies a change in distribution, and reduces the window size whenever it transitions to the next block. This allows for a very good estimation of the weak labeler accuracies, as the algorithm uses data mostly from the current block.

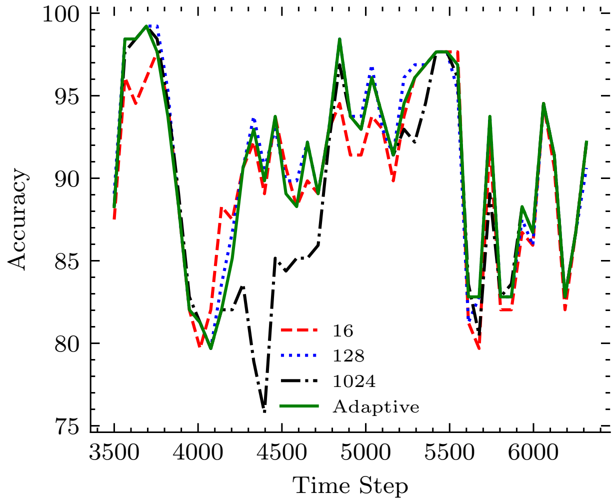

Figure 2: We report the accuracy of different fixed-window-size strategies and our adaptive algorithm for the Tennis Rally dataset over time (single run, *permute*). For each time step t, the reported accuracy is an average over the next 128 time steps. The plot shows that no fixed-window-size strategy is consistently good, while the adaptive strategy consistently matches the best strategy at any given time.

Clearly, there is a delay before our algorithm can correctly identify the distribution change since it needs to collect enough data from the new block to assess that a significant change happened. As a result, the estimation of the weak labelers' accuracy is worse for the data right after a block change. The variation in

[1] `https://github.com/BatsResearch/mazzetto-aistats2025-code`

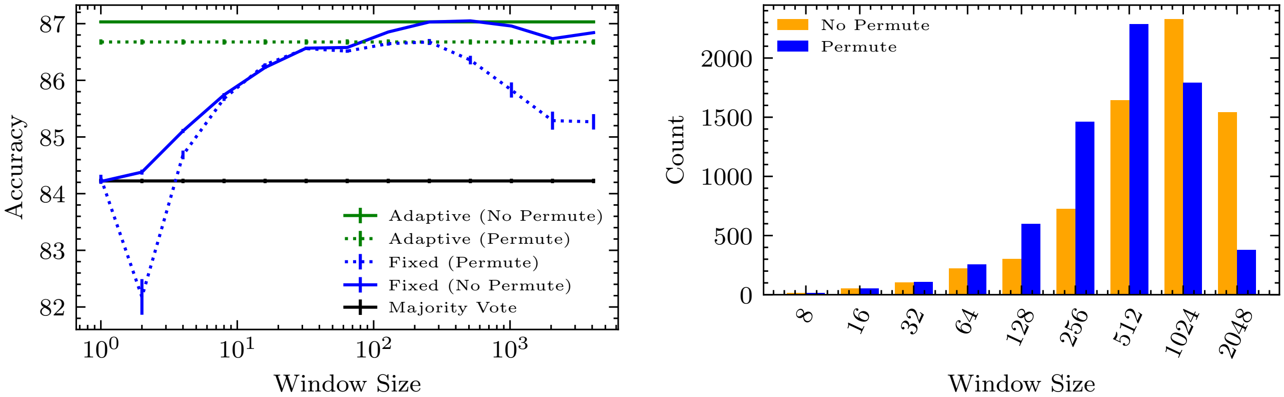

Figure 3: In the left plot, we report the average accuracy of the dynamically selected window sizes (our algorithm) and different fixed-window-size strategies for the Tennis Rally dataset. In the right plot, we report the histogram of the window sizes chosen by our algorithm. The reported results are for both experimental setups *permute* and *no permute*. The average and standard deviations are over 30 random runs, where the randomness of each run is due to the abstensions, and the shuffling of the weak labelers for the *permute* setup.

the accuracy estimates in the middle of a block is due to the window size selection strategy of the algorithm: whenever the algorithm increases the window size, the larger window includes in the following few steps a small number of samples from the previous block, resulting in a small additional error in the estimation. Previous approaches, which predetermine the amount of past information to use based on a parameter Δ that measures the magnitude of the drift, cannot address the drift scenario presented in this synthetic experiment. In this case, the duration of each stationary period varies, altering the number of samples we wish to extract from each period.

6.2 Video Analysis

We evaluate our algorithm on three video analysis tasks: **Basketball** (Sala et al., 2019), **Commercial**, and **Tennis Rally** (Fu et al., 2020). For each of those datasets, we are given a sequence of frames of a video, and a set of weak supervision sources (Zhang et al., 2021). We report here the results for the Tennis Rally dataset, where the goal is to identify tennis rallies during a broadcast match. Additional details for this dataset, and the results for the other video analysis tasks are presented in the supplementary material.

The Tennis Rally dataset has 6 weak supervision sources that can abstain on some of the input frames. Since our method requires the weak labelers to provide an output at each step, we map each abstention to a random label ± 1. For each of the video analysis tasks, we use two experimental setups. In the first experiment (*no permute*), we simply use the original dataset. In the second experiment (*permute*), we introduce an additional source of drift for the weak labelers. In particular, at each time step, with probability 10^{-3} we randomly shuffle the names of the weak labelers.

The first observation is that the performance of a given fixed-window-size strategy can change over time, as displayed in Figure 2. In particular, the best choice of window size also changes over time. Our algorithm obtains an overall good performance by consistently adapting its window size based on the input sequence. In the left plot of Figure 3, we compare the average accuracy of our algorithm with the majority vote and the fixed-window-size strategies for both experimental setups (*permute* and *no permute*). Our algorithm achieves an accuracy that is competitive with respect to the optimal fixed-window-size strategy. We remark that the accuracy values in these figures are computed using labeled data that is *not* available to the algorithm. Thus, an algorithm cannot evaluate the accuracy of the fixed-window strategies to choose the best one among them. Additionally, no fixed-window-size strategy can perform optimally in both drift settings as its accuracy is inherently linked to the data's drift patterns. On the other hand, our algorithm can automatically adjust to the drift and do at least as well as the best fixed-window-size strategy.

To further illustrate the adaptivity of our algorithm, we report in the right plot of Figure 3 a histogram of the window sizes chosen by our algorithm during its execution. This plot demonstrates that our algorithm varies its chosen window size throughout its execution. In particular, our algorithm indeed selects smaller window sizes in the *permute* setup, since it adapts to the additional drift due to the random shuffling of the identities of the weak labelers. We can also observe that the accuracies of the fixed-window-size strategies follow the phenomenon outlined in our theory: small window sizes and larger window sizes are less competitive than a proper selection of the window size depending on the drift. Using a large fixed-window-size strategy still provides a good solution for the *no permute* setup. This is most likely because there are a few weak labelers that are consistently accurate throughout the whole input sequence. On the other hand, for the *permute* setup, large fixed-window-size strategies exhibit a lower accuracy as weak labelers can have different performances throughout the window due to the random shuffling. Nevertheless, our algorithm can still adaptively recognize the identities of the accurate weak supervision sources. We also observe that the fixed-window strategy with size 1 is equivalent to the majority vote. This is most likely the reason why this strategy can be more competitive than other small fixed-window-size strategies that learn noisy weights based on little data.

7 Conclusion

This paper presents the first method with rigorous guarantees for learning from multiple noisy sources of labels in non-stationary settings *without* any prior assumptions about the nature of the changes over time.

Limitations and Future Work. The method presented in this paper does not extend to multi-class classification. One can follow the heuristic proposed by Fu et al. (2020), and execute multiple one-versus-all classifiers. However, this heuristic does not provide any formal analysis on the obtained result, and the outcome of different one-versus-all classifiers may not be consistent. Thus, a provable multi-class classification under drift is an interesting open problem.

Acknowledgements

We sincerely thank the anonymous reviewers for their valuable feedback and constructive suggestions, which have helped improve the quality of this work. Alessio Mazzetto was supported by a Kanellakis Fellowship at Brown University. We gratefully acknowledge support from Cisco. This material is based on research sponsored by Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory (AFRL) under agreement number FA8750-19-2-1006 and by the National Science Foundation (NSF) under award IIS-1813444. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory (AFRL) or the U.S. Government. Disclosure: Stephen Bach is an advisor to Snorkel AI, a company that provides software and services for data-centric artificial intelligence.

References

Agrahari, S. and Singh, A. K. (2022). Concept drift detection in data stream mining: A literature review. *Journal of King Saud University-Computer and Information Sciences*, 34(10):9523–9540.

Arachie, C. and Huang, B. (2019). Adversarial label learning. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Arachie, C. and Huang, B. (2021). A general framework for adversarial label learning. *Journal of Machine Learning Research*, 22(118):1–33.

Arora, S., Narayan, A., Chen, M. F., Orr, L. J., Guha, N., Bhatia, K., Chami, I., Sala, F., and Ré, C. (2022). Ask me anything: A simple strategy for prompting language models. *arXiv preprint arXiv:2210.02441*.

Auer, P., Gajane, P., and Ortner, R. (2019). Adaptively tracking the best bandit arm with an unknown number of distribution changes. In *Conference on Learning Theory*, pages 138–158. PMLR.

Awasthi, P., Cortes, C., and Mohri, C. (2023). Theory and algorithm for batch distribution drift problems. In *International Conference on Artificial Intelligence and Statistics*, pages 9826–9851. PMLR.

Bach, S. H., He, B., Ratner, A., and Ré, C. (2017). Learning the structure of generative models without labeled data. In *International Conference on Machine Learning (ICML)*.

Bach, S. H., Rodriguez, D., Liu, Y., Luo, C., Shao, H., Xia, C., Sen, S., Ratner, A., Hancock, B., Alborzi, H., Kuchhal, R., Ré, C., and Malkin, R. (2019). Snorkel DryBell: A case study in deploying weak supervision at industrial scale. In *ACM SIGMOD Conference on Management of Data (SIGMOD) Industry Track*.

Bai, Y., Zhang, Y.-J., Zhao, P., Sugiyama, M., and Zhou, Z.-H. (2022). Adapting to online label shift with provable guarantees. *Advances in Neural Information Processing Systems*, 35:29960–29974.

Balsubramani, A. and Freund, Y. (2015a). Optimally combining classifiers using unlabeled data. In *Conference on Learning Theory (COLT)*, pages 211–225.

Balsubramani, A. and Freund, Y. (2015b). Scalable semi-supervised aggregation of classifiers. In *Neural Information Processing Systems (NeurIPS)*.

Balsubramani, A. and Freund, Y. (2016). Optimal binary classifier aggregation for general losses. In *Neural Information Processing Systems (NeurIPS)*.

Barry, D. and Hartigan, J. A. (1993). A Bayesian analysis for change point problems. *Journal of the American Statistical Association*, 88(421):309–319.

Bartlett, P. L. (1992). Learning with a slowly changing distribution. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 243–252.

Bonald, T. and Combes, R. (2016). A streaming algorithm for crowdsourced data classification. *arXiv preprint arXiv:1602.07107*.

Bonald, T. and Combes, R. (2017). A minimax optimal algorithm for crowdsourcing. *Advances in Neural Information Processing Systems*, 30.

Bringer, E., Israeli, A., Shoham, Y., Ratner, A., and Ré, C. (2019). Osprey: Weak supervision of imbalanced extraction problems without code. In *International Workshop on Data Management for End-to-End Machine Learning (DEEM)*.

Caba Heilbron, F., Escorcia, V., Ghanem, B., and Carlos Niebles, J. (2015). Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 961–970.

Callahan, A., Fries, J. A., Ré, C., Huddleston, J. I., Giori, N. J., Delp, S., and Shah, N. H. (2019). Medical device surveillance with electronic health records. *NPJ Digital Medicine*, 2(1):1–10.

Chatterjee, O., Ramakrishnan, G., and Sarawagi, S. (2020). Robust data programming with precision-guided labeling functions. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Chen, V. S., Varma, P., Krishna, R., Bernstein, M., Ré, C., and Fei-Fei, L. (2019a). Scene graph prediction with limited labels. In *IEEE/CVF International Conference on Computer Vision (ICCV)*.

Chen, Y., Lee, C.-W., Luo, H., and Wei, C.-Y. (2019b). A new algorithm for non-stationary contextual bandits: Efficient, optimal and parameter-free. In *Conference on Learning Theory*, pages 696–726. PMLR.

Dawid, A. P. and Skene, A. M. (1979). Maximum likelihood estimation of observer error-rates using the em algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 28(1):20–28.

Fries, J. A., Steinberg, E., Khattar, S., Fleming, S. L., Posada, J., Callahan, A., and Shah, N. H. (2021). Ontology-driven weak supervision for clinical entity classification in electronic health records. *Nature Communications*, 12(1):1–11.

Fu, D., Chen, M., Sala, F., Hooper, S., Fatahalian, K., and Ré, C. (2020). Fast and three-rious: Speeding up weak supervision with triplet methods. In *International Conference on Machine Learning (ICML)*, pages 3280–3291. PMLR.

Fu, D. Y., Crichton, W., Hong, J., Yao, X., Zhang, H., Truong, A., Narayan, A., Agrawala, M., Ré, C., and Fatahalian, K. (2019). Rekall: Specifying video events using compositions of spatiotemporal labels. *arXiv preprint arXiv:1910.02993*.

Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., and Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4):1–37.

Ghosh, A., Kale, S., and McAfee, P. (2011). Who moderates the moderators? crowdsourcing abuse detection in user-generated content. In *Proceedings of the 12th ACM conference on Electronic commerce*, pages 167–176.

Han, E., Huang, C., and Wang, K. (2024). Model assessment and selection under temporal distribution shift. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*.

Hanneke, S. and Yang, L. (2019). Statistical learning under nonstationary mixing processes. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1678–1686. PMLR.

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

Joglekar, M., Garcia-Molina, H., and Parameswaran, A. (2015). Comprehensive and reliable crowd assessment algorithms. In *International Conference on Data Engineering (ICDE)*.

Killick, R., Fearnhead, P., and Eckley, I. A. (2012). Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500):1590–1598.

Li, W., Yang, X., Liu, W., Xia, Y., and Bian, J. (2022). Ddg-da: Data distribution generation for predictable concept drift adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 36, pages 4092–4100.

Liu, Q., Peng, J., and Ihler, A. T. (2012). Variational inference for crowdsourcing. In *Neural Information Processing Systems (NeurIPS)*.

Long, P. M. (1998). The complexity of learning according to two models of a drifting environment. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 116–125.

Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., and Zhang, G. (2018). Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering*, 31(12):2346–2363.

Mazzetto, A. (2024). An improved algorithm for learning drifting discrete distributions. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Mazzetto, A., Ceccarello, M., Pietracaprina, A., Pucci, G., and Upfal, E. (2025). Center-based approximation of a drifting distribution. In *36th International Conference on Algorithmic Learning Theory (ALT)*.

Mazzetto, A., Cousins, C., Sam, D., Bach, S. H., and Upfal, E. (2021a). Adversarial multiclass learning under weak supervision with performance guarantees. In *International Conference on Machine Learning (ICML)*.

Mazzetto, A., Sam, D., Park, A., Upfal, E., and Bach, S. H. (2021b). Semi-supervised aggregation of dependent weak supervision sources with performance guarantees. In *Artificial Intelligence and Statistics (AISTATS)*.

Mazzetto, A. and Upfal, E. (2023a). An adaptive algorithm for learning with unknown distribution drift. In *Neural Information Processing Systems (NeurIPS)*.

Mazzetto, A. and Upfal, E. (2023b). Nonparametric density estimation under distribution drift. In *International Conference on Machine Learning (ICML)*.

Mohri, M. and Muñoz Medina, A. (2012). New analysis and algorithm for learning with drifting distributions. In *International Conference on Algorithmic Learning Theory*, pages 124–138. Springer.

Nitzan, S. and Paroush, J. (1982). Optimal decision rules in uncertain dichotomous choice situations. *International Economic Review*, pages 289–297.

Parisi, F., Strino, F., Nadler, B., and Kluger, Y. (2014). Ranking and combining multiple predictors without labeled data. *Proceedings of the National Academy of Sciences of the USA*, 111(4):1253–1258.

Ratner, A., Bach, S. H., Ehrenberg, H., Fries, J., Wu, S., and Ré, C. (2017). Snorkel: Rapid training data creation with weak supervision. In *Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases*, volume 11, page 269. NIH Public Access.

Ratner, A., Hancock, B., Dunnmon, J., Sala, F., Pandey, S., and Ré, C. (2019). Training complex models with multi-task weak supervision. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4763–4771.

Ratner, A. J., De Sa, C. M., Wu, S., Selsam, D., and Ré, C. (2016). Data programming: Creating large training sets, quickly. In *Neural Information Processing Systems (NeurIPS)*.

Raykar, V. C., Yu, S., Zhao, L. H., Valadez, G. H., Florin, C., Bogoni, L., and Moy, L. (2010). Learning from crowds. *Journal of Machine Learning Research*, 11(4).

Safranchik, E., Luo, S., and Bach, S. H. (2020). Weakly supervised sequence tagging from noisy rules. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Sala, F., Varma, P., Sagawa, S., Fries, J., Fu, D., Khattar, S., Ramamoorthy, A., Xiao, K., Fatahalian, K., Priest, J., et al. (2019). Multi-resolution weak supervision for sequential data. In *Neural Information Processing Systems (NeurIPS)*.

Shin, C., Li, W., Vishwakarma, H., Roberts, N., and Sala, F. (2022). Universalizing weak supervision. In *International Conference on Learning Representations (ICLR)*.

Smith, R., Fries, J. A., Hancock, B., and Bach, S. H. (2022). Language models in the loop: Incorporating prompting into weak supervision. *arXiv:2205.02318 [cs.LG]*.

Suri, S., Chanda, R., Bulut, N., Narayana, P., Zeng, Y., Bailis, P., Basu, S., Narlikar, G., Ré, C., and Sethi, A. (2020). Leveraging organizational resources to adapt models to new data modalities. *Proc. VLDB Endow.*, 13(12):3396–3410.

Truong, C., Oudre, L., and Vayatis, N. (2020). Selective review of offline change point detection methods. *Signal Processing*, 167.

Varma, P., He, B. D., Bajaj, P., Khandwala, N., Banerjee, I., Rubin, D., and Ré, C. (2017). Inferring generative model structure with static analysis. *Neural Information Processing Systems (NeurIPS)*.

Varma, P., Sala, F., He, A., Ratner, A., and Ré, C. (2019). Learning dependency structures for weak supervision models. In *International Conference on Machine Learning (ICML)*.

Wei, C.-Y. and Luo, H. (2021). Non-stationary reinforcement learning without prior knowledge: An optimal black-box approach. In *Conference on Learning Theory*, pages 4300–4354. PMLR.

Xian, Y., Lampert, C. H., Schiele, B., and Akata, Z. (2018). Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2251–2265.

Yu, H., Li, J., Lu, J., Song, Y., Xie, S., and Zhang, G. (2023). Type-ldd: A type-driven lite concept drift detector for data streams. *IEEE Transactions on Knowledge and Data Engineering*, 36(12):9476–9489.

Yu, H., Zhang, Q., Liu, T., Lu, J., Wen, Y., and Zhang, G. (2022). Meta-add: A meta-learning based pre-trained model for concept drift active detection. *Information Sciences*, 608:996–1009.

Yu, P. and Bach, S. H. (2023). Alfred: A system for prompted weak supervision. In *Meeting of the Association for Computational Linguistics (ACL) Demonstration*.

Yu, Y., Zuo, S., Jiang, H., Ren, W., Zhao, T., and Zhang, C. (2021). Fine-tuning pre-trained language model with weak supervision: A contrastive-regularized self-training approach. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*.

Zhang, J., Hsieh, C.-Y., Yu, Y., Zhang, C., and Ratner, A. (2022). A survey on programmatic weak supervision. *arXiv preprint arXiv:2202.05433*.

Zhang, J., Yu, Y., Li, Y., Wang, Y., Yang, Y., Yang, M., and Ratner, A. (2021). WRENCH: A comprehensive benchmark for weak supervision. In *Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*.

Zhang, Y., Chen, X., Zhou, D., and Jordan, M. I. (2016). Spectral methods meet EM: A provably optimal algorithm for crowdsourcing. *The J. of Mach. Learn. Research*, 17(1):3537–3580.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]
   (b) The license information of the assets, if applicable. [Yes]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Not Applicable]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

A Additional Experiments

We provide further details about our experimental setup and also show additional results on other tasks. All experiments were conducted on a M2 Max Macbook Pro with 32GB RAM. This section is organized as follows:

1. In Appendix A.1, we report additional details for our experiments on the video classification tasks (Section 6.2). In particular, we report the experimental results for the Basketball and Commercial datasets, and also provide additional details on the Tennis dataset.

2. In Appendix A.2, we provide experimental results for an image classification task with artificial drift that uses the Animals with Attributes 2 (AwA2) dataset.

3. In Appendix A.3, we discuss the hyper-parameters of our algorithm.

A.1 Video Classification

In this subsection, we provide additional details and results for the experiments on the video classification tasks (Section 6.2).

In this set of experiments, we use three datasets: **Commercial**, **Tennis Rally**, and **Basketball**.

Commercial (Fu et al., 2019): the goal is to identify commercial segments from a TV news broadcast.

Basketball (Fu et al., 2020): the goal is to identify basketball videos in a subset of ActivityNet (Caba Heilbron et al., 2015).

Tennis Rally (Fu et al., 2020): the goal is to identify tennis rallies during a tennis match.

We use the version of those datasets provided by the weak supervision benchmark platform Wrench (Zhang et al., 2021). For each of those datasets, we only use the training data, since it contains the largest number of data points. We use the weak supervision sources provided by Wrench for each of those tasks. Table 1 provides additional information on the number of data points and weak supervision sources for each task.

Table 1: Number of weak labelers and data points for the three datasets.

Dataset	Number of Weak Labelers	Number of Data Points
Basketball	4	6959
Commercial	4	64130
Tennis	6	17970

The sequence of data points described the sequence of frames of the video. The weak supervision sources provide a vote for each frame, and the goal is to combine their votes to provide the correct classification of the frame. In the original dataset, the weak supervision sources are allowed to abstain on some of the data points. To fit those datasets to our theoretical framework, we map each abstension to a random vote in $\{-1, 1\}$.

As described in Section 6.2, in the "no permute" case we use the dataset as is. In the "permute" case, we randomly shuffle the names of the weak labelers with probability 10^{-3} at each step.

For each of those datasets, we run an experimental evaluation across 30 runs. In each run, there is randomness introduced in the mapping of the abstentions to random vote, and also in the random shuffling of the name of the weak labelers for the "permute" case.

A.1.1 Additional Results

In Figure 4 and Figure 5, we report experimental results for respectively Commercial and Basketball. Those results corroborate the findings obtained with the Tennis Rally dataset (Section 6.2). In particular, we show again that smaller window sizes have higher accuracy in the "permute" case due to the additional drift. Accordingly, the right plot of Figure 4 clearly shows that the adaptive algorithm favors smaller window size in the “permute” case for the Commercial dataset.

We highlight that although the Basketball dataset contains a lot of noise due to having the highest percentage of abstension and the smallest number of data points, the shape of the plot of the left plot of Figure 5 still resembles what we would expect from our theoretical framework: smaller and larger window sizes are less competitive (accuracy-wise) than a proper selection of the window size depending on the drift.

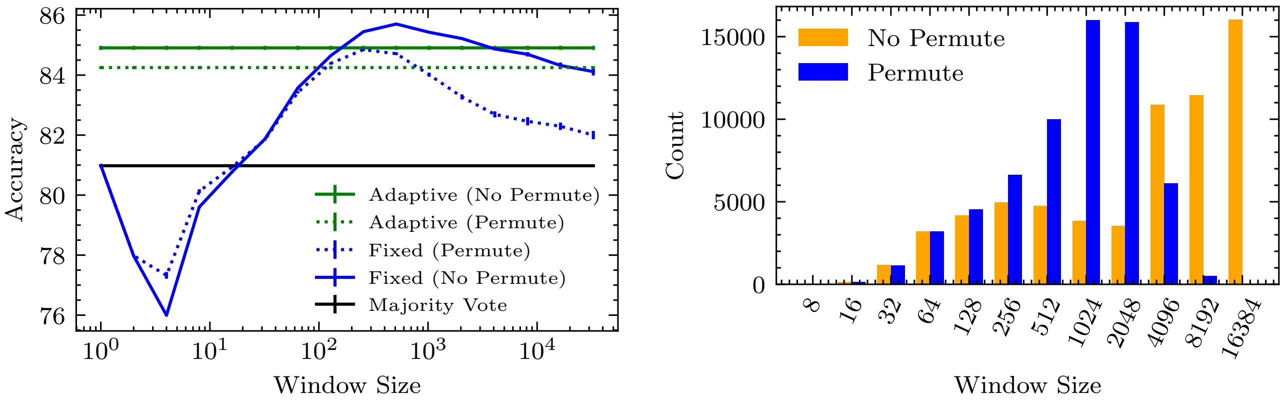

Figure 4: (**Commercial**). In the left plot, we report the average accuracy of the dynamically selected window sizes (our algorithm) and different fixed-window-size strategies for the Commercial Dataset. In the right plot, we report the histogram of the window sizes chosen by our algorithm.

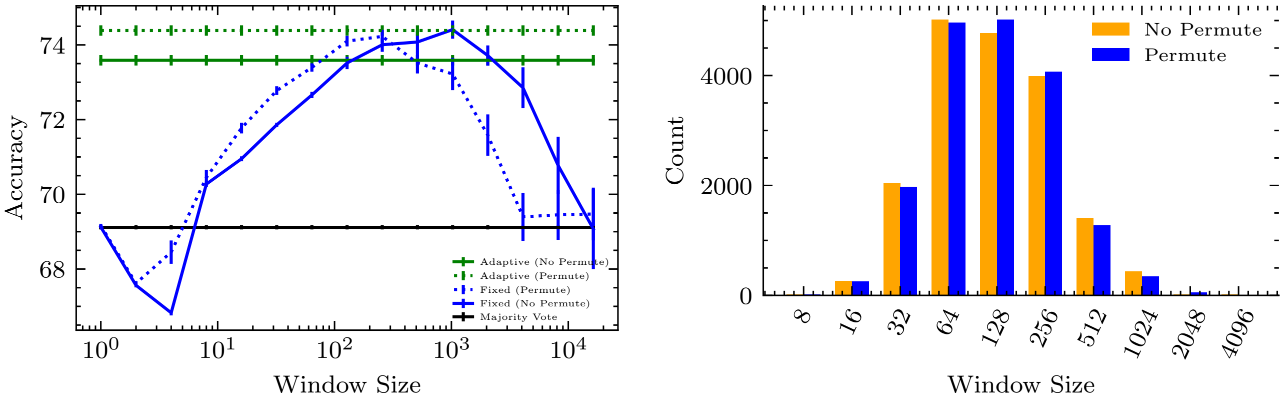

Figure 5: (**Basketball**). We report the same results of Figure 4 but for the Basketball dataset.

For completeness, in Figures 6 and 7, we also report the F1 Score (the harmonic mean of precision and recall) obtained by our adaptive algorithm and the fixed-window-size strategies across all three tasks. We observe that our observations regarding the accuracy numbers also applies to the F1 score.

We also plot the weights computed by our algorithm for the weak supervision sources over a single run for both the settings “permute” and “no permute” (Figure 8 and 9). We remind that the weights of the algorithm are computed according to (2) given the adaptive algorithm’s estimated accuracies.

In the no permute case, there are no sudden changes in the weights of the learners. This also suggests that the original dataset does not exhibit a significant drift on which weak labelers are accurate. However, in the “permute” case there are sharp changes in the weights of the learner, which correspond to when a shuffle of the names of the weak labelers occurs in the dataset. This shows that our algorithm can adapt to the drift introduced over the weak labelers.

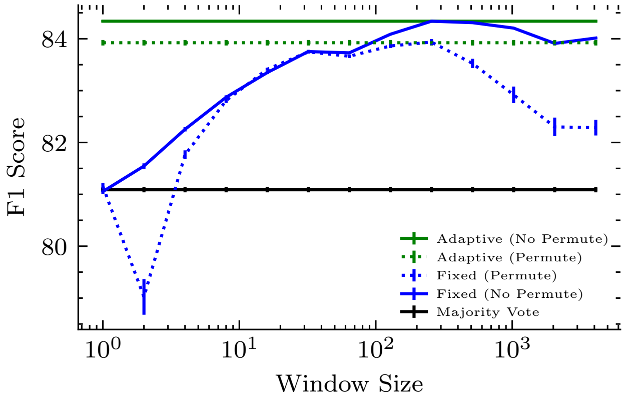

Figure 6: F1 Score on the Tennis dataset.

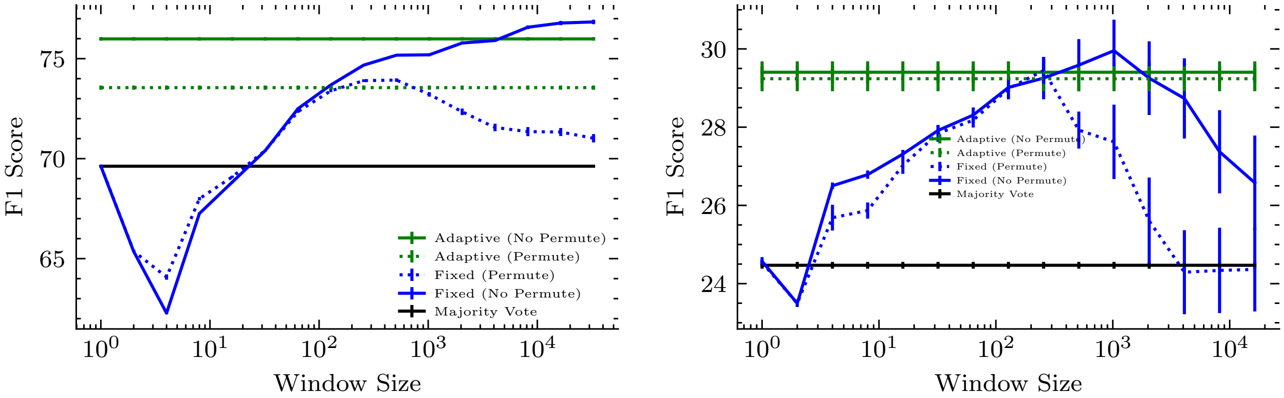

Figure 7: F1 Score on the Commercial dataset (left) and on the Basketball dataset (right)

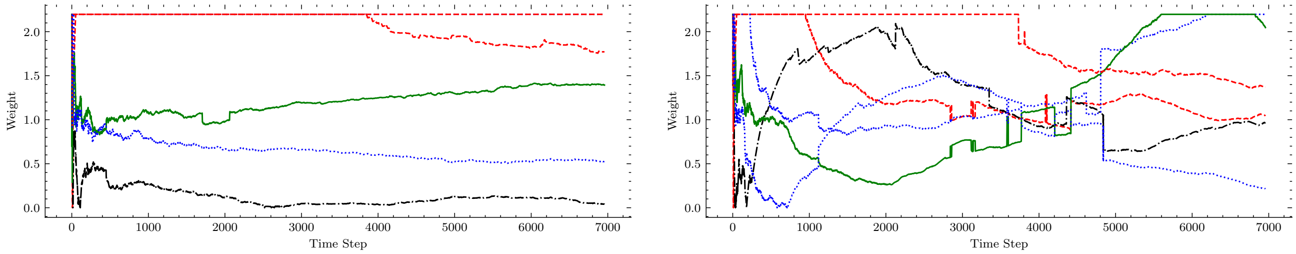

Figure 8: **Tennis Rally.** The weights given to each weak labeler over time by a single run of our adaptive algorithm. The left is the ”no permute” case and the right is the ”permute” case.

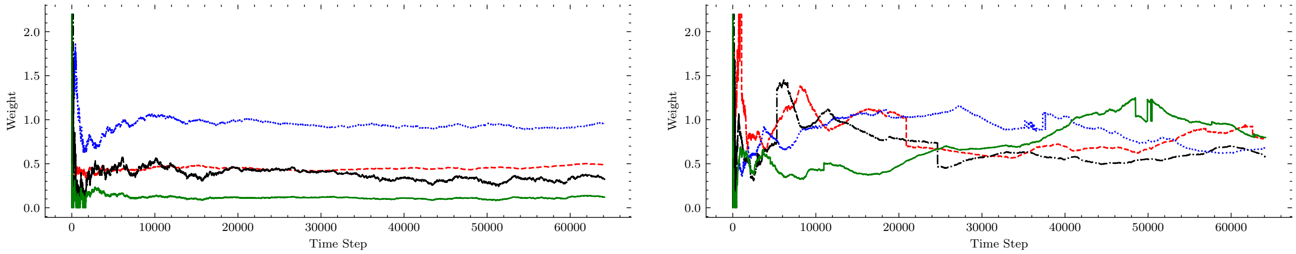

Figure 9: **Commercial.** The weights given to each weak labeler over time by a single run of our adaptive algorithm. The left is the ”no permute” case and the right is the ”permute” case.

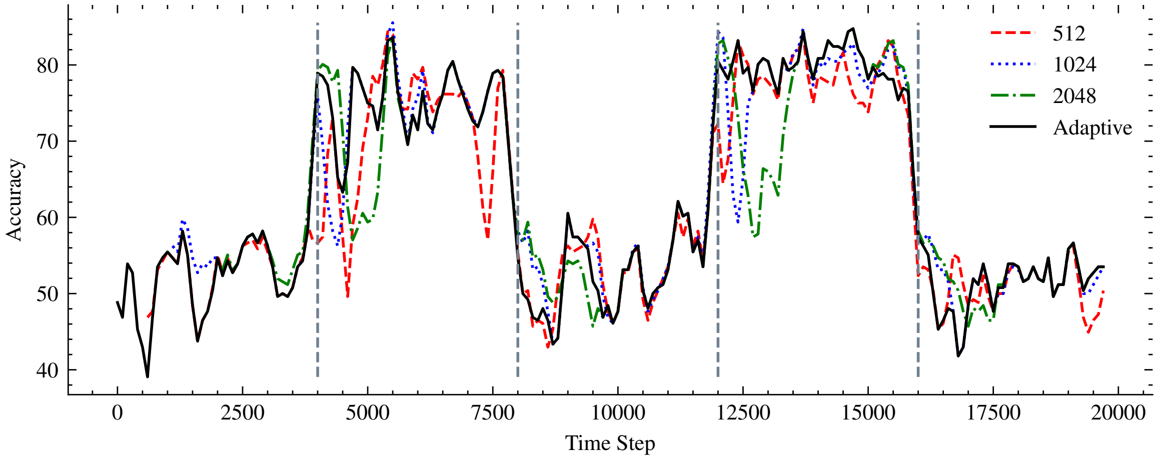

Figure 10: Accuracy of different window sizes for each time step for AwA2 dataset. The vertical lines mark a distribution shift. The accuracies are obtained from one random execution. Accuracies for window size r are only reported for time steps $t \geq r$. For each time step t, the reported accuracy is an average over the next 256 time steps, $X_t, \ldots, X_{t+256-1}$.

A.2 Image Classification

In this subsection, we report the experimental results for an image classification task with artificial drift built from the *Animals with Attributes2* (AwA2) dataset (Xian et al., 2018).

The Animals with Attributes2 dataset consists of images of animals from 50 disjoint classes, that are split into 40 seen classes, used for training, and 10 unseen classes, used for testing. The dataset also provides the relations among 85 attributes (e.g., “patches”) and classes through a binary class-attribute matrix, where each entry indicates whether animals from a certain class exhibit an attribute or not. Following previous work (Mazzetto et al., 2021b,a), we obtain weak supervision sources by fine-tuning ResNet-18 models (He et al., 2016) on the seen classes to detect each of the attributes.

We use this dataset to construct a binary classification task with artificial drift. We define two target classes over the unseen test classes. The first target class contains images from the classes “horse” and “sheep”; the second target class contains images from classes “giraffe” and “bobcat.” We use the class-attribute matrix to identify attributes that are helpful to distinguish between those two target classes. An attribute is helpful if 1) it appears only in one of the target classes and 2) it consistently appears or does not appear in both classes of each target class. Using this criteria, we choose the attribute detectors for “black”, “white”, “orange”, “yellow”, “spots”, and “domestic” attributes as weak supervision sources.

To create a dataset with drift, we sample $5T$ images with repetition from the selected classes with $T = 4000$. We partition the data into five contiguous blocks of size T. In the first block, we sample from “sheep” and “bobcat” classes with a probability of 0.1 and from “horse” and “giraffe” classes with a probability of 0.9. To create drift, we alternate the probability of sampling from each of the subclasses between 0.1 and 0.9 for consecutive blocks.

In Figure 10, we visualize the accuracy of the adaptively selected window sizes and multiple fixed window sizes over time. As expected, the accuracy of fixed window sizes changes over time. For example, small window sizes achieve better accuracy shortly after a distribution shift occurs by limiting the number of out-of-distribution samples, and large window sizes achieve better accuracy toward the end of each block by using more samples from the same distribution. On the other hand, our algorithm successfully detects the drift and selects the best window size for each time step accordingly. As a result, our algorithm maintains a close-to-optimal performance for most of the time steps. These results emphasize that the optimal window size itself can change over time.

We report the window sizes selected by our algorithm at each time step in Figure 11. Consistent with previous results on synthetic data, our algorithm successfully detects the drift and selects small window sizes to limit out-of-distribution samples. At the same time, for stationary periods, our algorithm selects large window sizes to include more samples from the same distribution.

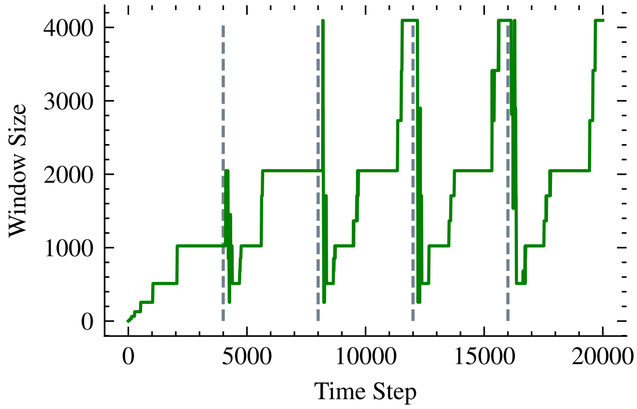

Figure 11: Adaptively selected window sizes for each time step for AwA2 dataset. Vertical lines mark a distribution shift. Each point is an average of three random executions.

In Figure 12 and Table 2, we plot the average accuracy of the dynamically selected window sizes and multiple fixed window sizes for the AwA2 dataset. Using a proper window size is crucial for achieving good performance. Using too large or too small window sizes decreases the accuracy by up to 7 percentage points. On the other hand, our algorithm adapts to each time step and selects a close-to-optimal window size without any prior knowledge about the drift. As a result, adaptive window sizes achieve better or comparable accuracy to any fixed strategy. Although some fixed strategies have a competitive accuracy on average, their accuracy changes significantly for different time steps.

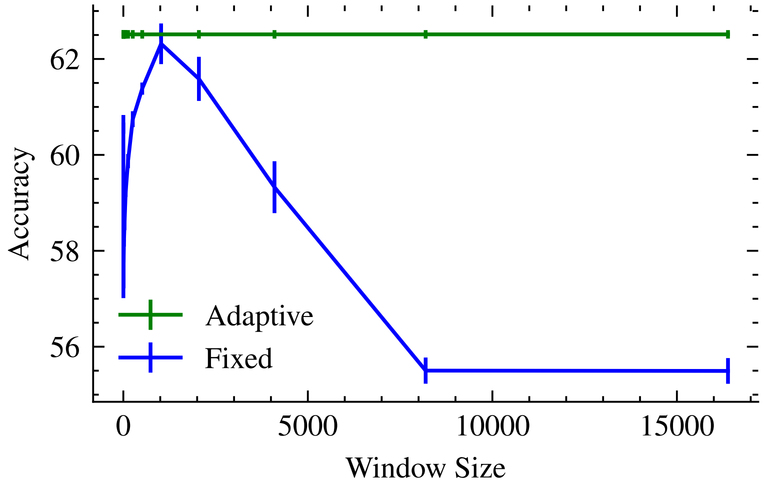

Figure 12: Accuracy of the dynamically selected window sizes as well as different fixed-window-size strategies for the AwA2 dataset. For each window size, the average accuracy and standard error of the mean are reported over three random executions.

Table 2: Accuracy of the dynamically selected window sizes as well as different fixed-window-size strategies for the AwA2 dataset. For each window size, the average accuracy and standard error of the mean are reported over three random executions.

Window Size	Accuracy
Adaptive	62.51 ± 0.09
All Past	55.49 ± 0.27
Majority Vote	60.64 ± 0.20
2	57.22 ± 0.21
4	58.72 ± 0.19
8	57.75 ± 0.12
16	58.16 ± 0.07
32	58.56 ± 0.13
64	59.23 ± 0.10
128	59.87 ± 0.15
256	60.75 ± 0.17
512	61.38 ± 0.13
1024	62.32 ± 0.42
2048	61.59 ± 0.46
4096	59.33 ± 0.54
8192	55.50 ± 0.27
16384	55.49 ± 0.27

A.3 Hyper-parameters

In this section, we discuss the hyper-parameters of the adaptive algorithm. The discussion of this section refers to the more general statement of our main result (Theorem B.1). Our algorithm has the following hyper-parameters:

- The value $\delta \in (0, 1)$ represents the failure probability of the algorithm.
- The sequence $\mathcal{R} = \{r_1, \ldots, r_k\}$ represents the possible window sizes that the algorithm considers. In order to obtain better guarantees in Theorem B.1, we look for a sequence $\mathcal{R}$ such that: (i) the minimum ratio between consecutive elements γ_m is large, as this avoids comparing window sizes that are very similar with one another and for which it is very hard to detect if drift occurred; (ii) the maximum ratio between consecutive elements γ_M is small, as this prevents a situation in which $\mathcal{R}$ is sparse, and there is no value in $\mathcal{R}$ that is close to the optimal window size. A natural choice for $\mathcal{R}$ is to use a sequence of powers where $r_i = \gamma^i$ for some $\gamma > 1$, then $\gamma_m = \gamma_M = 1/\gamma$. With our analysis, the best guarantees of the algorithm are achieved by using a sequence of powers of $1/(\sqrt{2} - 1)^2$ as $\mathcal{R}$.
- The value of β affects the threshold used in our algorithm. Intuitively, the value of β is proportional to how much drift the algorithm must observe before stopping, and it affects the sensitivity of our algorithm to detect drift. The optimal value of β that minimizes the upper bound of our algorithm is $\beta = \sqrt{2} - 1$.

In our experiments, we let $\delta = 0.1$ be an arbitrarily small failure probability. We let $\mathcal{R} = \{2^0, 2^1, \ldots, 2^{19}\}$. We use powers of 2 rather than powers of $1/(\sqrt{2} - 1)^2$ to define $\mathcal{R}$ (see discussion above) for ease as all the resulting window sizes are integers.

We run an additional experiment to see the effect of the value β on the results of our adaptive algorithm. We report the accuracy and the F1 score of our adaptive algorithm for the Basketball dataset (Figure 13) and for the Tennis Rally dataset (Figure 14). We observe that small values of β provide similar results in both datasets, and the value $\beta = \sqrt{2} - 1$ is also a good choice. Interestingly, we can notice that for the "permute" setup, higher values of β lead to a decrease in performance for both datasets. This is most likely due to the fact that for higher values of β, the algorithm needs to observe a larger magnitude of drift in order to choose a smaller window size. For the "permute", it is necessary to react to the introduced drift to obtain better performance, hence smaller values of β are more competitive. In our experiments, we choose $\beta = 0.1$ as an arbitrarily small value.

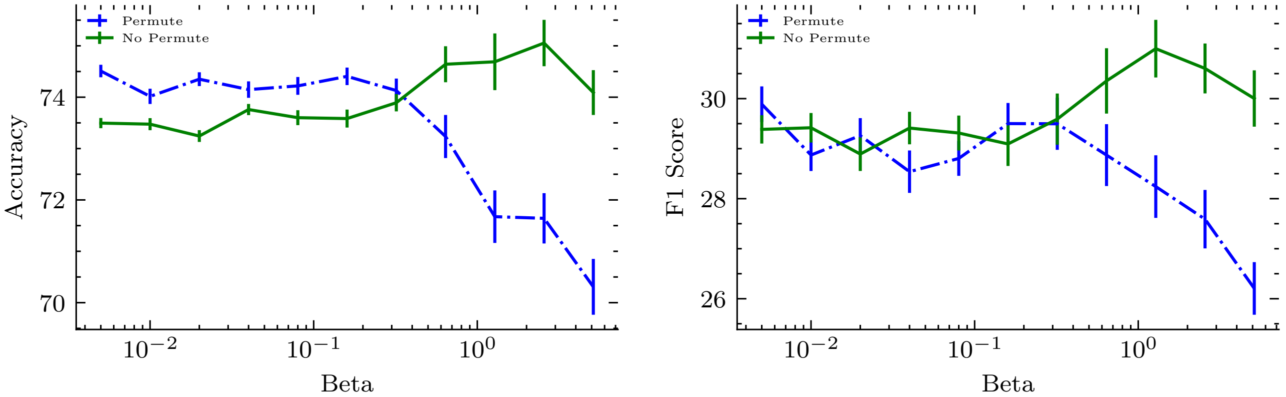

Figure 13: Accuracy (left) and F1 score (right) of our adaptive algorithm for the Basketball dataset varying the value of β. The reported results are an average over 30 runs.

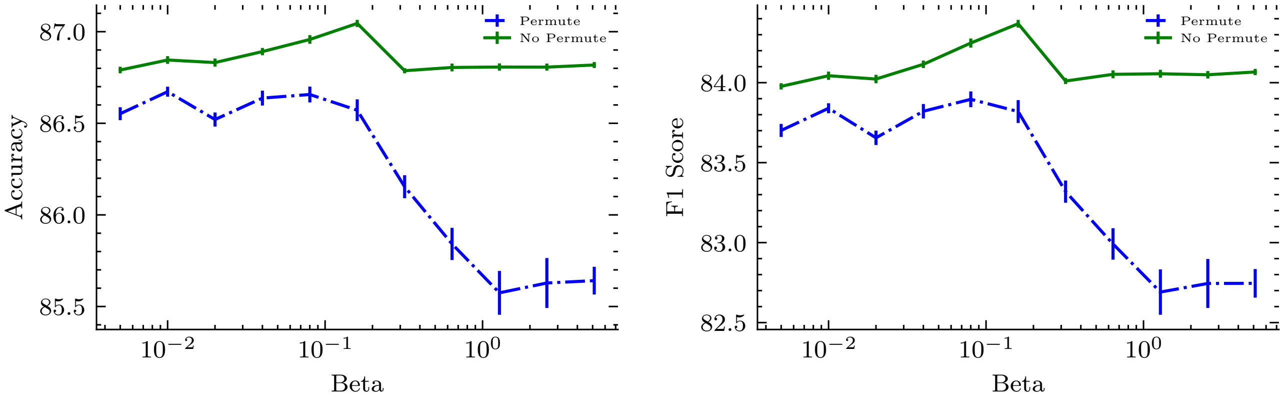

Figure 14: Accuracy (left) and F1 score (right) of our adaptive algorithm for the Tennis Rally dataset varying the value of β. The reported results are an average over 30 runs.

B Deferred Proofs

In this section, we prove our paper's main theorem (Theorem 5.2). We establish the following broader result, from which Theorem 5.2 follows directly as a corollary. Compared to Theorem 5.2, the following theorem explicits the constants in the upper bound, and it allows to consider an arbitrary choice of window sizes $\mathcal{R} = \{r_1, \ldots, r_m\}$ such that $r_1 < \ldots < r_m$.

Theorem B.1. *Let Assumptions 2.1, 2.2 and 4.1 hold. Let $\delta \in (0,1)$ and $\beta > 0$. Let $\mathcal{R} = \{r_1, \ldots, r_m\}$. Assume $n \geq 3$. If we run Algorithm 1 at time $t \geq r_1$, then with probability at least $1-\delta$ it provides an estimate $\hat{\boldsymbol{p}} = (\hat{p}_1, \ldots, \hat{p}_n)$ such that*

$$\|\boldsymbol{p}(t) - \hat{\boldsymbol{p}}\|_\infty \leq \frac{5\Phi_{\mathcal{R},\beta}}{2\tau^2} \min_{r \in [r_1, \min(t, r_m)]} \left(\frac{A_{\delta,n,m}}{\sqrt{r}} + 12 \sum_{k=t-r+1}^{r-1} \|\boldsymbol{p}(k) - \boldsymbol{p}(k+1)\|_\infty \right)$$

where $A_{\delta,n,m} \doteq \sqrt{2\ln[(2m-1)\cdot n(n-1)/\delta]}$, *and* $\Phi_{\mathcal{R},\beta} = 1 + \max\left\{\frac{2\beta+2}{\gamma_m(1-\gamma_M)}, \frac{2\beta+2}{\beta(1-\gamma_M)}\right\}$, *with* $\gamma_M = \max\sqrt{r_k/r_{k+1}}$ *and* $\gamma_m = \min\sqrt{r_k/r_{k+1}}$.

A more detailed discussion on the hyper-parameters β and $\mathcal{R}$ is provided in Appendix A.3.

We outline the proof:

1. We show that the estimation of the correlation matrix at time t by using the previous r samples can be decomposed into the sum of two error components: *a statistical error* and a *drift error* (Appendix B.1).

2. In order to bound the statistical error, we use standard concentration inequalities to show that the estimation of the correlation matrix obtained by using the r previous samples is close to its expected value with error $O(1/\sqrt{r})$ with high-probability (Appendix B.2).
3. In order to upper bound the drift error, we show an inequality that relates the drift in correlation matrices over time with the drift of the accuracies of the weak labelers (Appendix B.3).
4. We use the previous results to show the trade-off between statistical error and drift error depicted in Lemma 5.1 (Appendix B.4).
5. We prove Theorem B.1: we show how to dynamically select the window size in order to optimize the above trade-off (Appendix B.5).

B.1 Error Decomposition

We define the average correlation matrix over the previous r samples as

$$\boldsymbol{C}^{[r]}(t) = \frac{1}{r} \sum_{k=t-r+1}^{t} \boldsymbol{C}(k) \ .$$

We show the following error decomposition in the upper bound of the error of estimating the matrix $\boldsymbol{C}(t)$ by using the empirical matrix $\hat{\boldsymbol{C}}^{[r]}(t)$ induced by the previous r samples.

Proposition B.2. *For any $1 \leq r \leq t$, we have that*

$$\left\| \boldsymbol{C}(t) - \hat{\boldsymbol{C}}^{[r]}(t) \right\|_\infty \leq \underbrace{\left\| \boldsymbol{C}^{[r]}(t) - \hat{\boldsymbol{C}}^{[r]}(t) \right\|_\infty}_{\textit{statistical error}} + \underbrace{\sum_{i=t-r+1}^{t-1} \left\| \boldsymbol{C}(i) - \boldsymbol{C}(i+1) \right\|_\infty}_{\textit{drift error}} \ . \tag{4}$$

Proof. We use the triangle inequality, and obtain that:

$$\begin{aligned} \left\| \boldsymbol{C}(t) - \hat{\boldsymbol{C}}^{[r]}(t) \right\|_\infty &= \left\| \boldsymbol{C}(t) - \boldsymbol{C}^{[r]}(t) + \boldsymbol{C}^{[r]}(t) - \hat{\boldsymbol{C}}^{[r]}(t) \right\|_\infty \\ &\leq \left\| \boldsymbol{C}^{[r]}(t) - \hat{\boldsymbol{C}}^{[r]}(t) \right\|_\infty + \left\| \boldsymbol{C}(t) - \boldsymbol{C}^{[r]}(t) \right\|_\infty \ . \end{aligned} \tag{5}$$

Again, by using the triangle inequality, we obtain the following chain of inequalities

$$\begin{aligned} \left\| \boldsymbol{C}(t) - \boldsymbol{C}^{[r]}(t) \right\|_\infty \leq \frac{1}{r} \sum_{i=t-r+1}^{t} \left\| \boldsymbol{C}(t) - \boldsymbol{C}(i) \right\|_\infty &\leq \sup_{t-r+1 \leq i \leq t} \left\| \boldsymbol{C}(t) - \boldsymbol{C}(i) \right\|_\infty \\ &\leq \sum_{i=t-r+1}^{t-1} \left\| \boldsymbol{C}(i+1) - \boldsymbol{C}(i) \right\|_\infty \ . \end{aligned} \tag{6}$$

By plugging the above inequality into (5), we obtain the statement.

□

Observe that by definition, we have the following relation $\mathbb{E}\, \hat{\boldsymbol{C}}^{[r]}(t) = \boldsymbol{C}^{[r]}(t)$. The statistical error term describes how much the empirical estimation deviate to its expectation, i.e., it is equal to

$$\left\| \hat{\boldsymbol{C}}^{[r]}(t) - \mathbb{E}\, \hat{\boldsymbol{C}}^{[r]}(t) \right\|_\infty \ .$$

This error is related to the variance of $\hat{\boldsymbol{C}}^{[r]}$, and we will use a concentration inequality to provide an upper bound to this term (Appendix B.2).

The drift error term describes the estimation error due to a change in the accuracy of the weak labelers, and indeed it is equal to 0 if no change occurs. In Appendix B.3, we will show how to analytically relate this error to the drift of the weak labelers' accuracies.

B.2 Upper Bound to the Statistical Error

In this subsection, our main goal is to provide an upper bound to the statistical error term

$$\left\|\hat{\boldsymbol{C}}^{[r]}(t) - \mathbb{E}\,\hat{\boldsymbol{C}}^{[r]}(t)\right\|_{\infty} = \left\|\boldsymbol{C}^{[r]}(t) - \hat{\boldsymbol{C}}^{[r]}(t)\right\|_{\infty} \tag{7}$$

by using a concentration inequality. The following result immediately follows by using McDiarmid's inequality.

Proposition B.3. *Consider a pair of indexes $(i,j) \in \{1,\dots,n\}^2$. Let $\delta > 0$. With probability at least $1-\delta$, it holds*

$$\left|C_{ij}^{[r]}(t) - \hat{C}_{ij}^{[r]}(t)\right| \leq \sqrt{\frac{2\ln(2/\delta)}{r}} \ .$$

Proof. Let $f(X_{t-r+1},\dots,X_t) = \hat{C}_{ij}^{[r]}(t)$. By definition of $\boldsymbol{C}(\cdot)$, it is easy to verify that

$$\mathbb{E}\,f(X_{t-r+1},\dots,X_t) = \frac{1}{r}\sum_{k=t-r+1}^{t} C_{ij}(k) = C_{ij}^{[r]}(t) \ .$$

Since each change of a single variable can change the value of f by at most $2/r$, we can use McDiarmid's inequality, and obtain that with probability at least $1-\delta$, it holds that

$$|f - \mathbb{E}\,f| \leq \left|C_{ij}^{[r]}(t) - \hat{C}_{ij}^{[r]}(t)\right| \leq \sqrt{\frac{2\ln(2/\delta)}{r}} \ . \tag{8}$$

□

An upper bound to the statistical error term (7) immediately follows by using the above proposition and taking an union bound over all possible indexes i,j. Since the matrices are symmetric, and the diagonal is always equal to 1, it is sufficient to take an union bound over only $n(n-1)/2$ choices of those indexes. Thus, with probability at least $1-\delta$, it holds that

$$\left\|\boldsymbol{C}^{[r]}(t) - \hat{\boldsymbol{C}}^{[r]}(t)\right\|_{\infty} \leq \sqrt{\frac{2\ln(n(n-1)/\delta)}{r}} \ . \tag{9}$$

For our algorithm, we will also need to show an upper bound to the error of estimating the difference between two correlation matrices with different window sizes. Since this result follows with a similar argument of Proposition B.3, we report it here.

Proposition B.4. *Consider a pair of indexes $(i,j) \in \{1,\dots,n\}^2$. Let $\delta > 0$, and let r, r' be two integers such that $1 \leq r < r' \leq t$. With probability at least $1-\delta$, it holds*

$$\left|\hat{C}_{ij}^{[r]}(t) - \hat{C}_{ij}^{[r']}(t) - C_{ij}^{[r]} + C_{ij}^{[r']}\right| \leq \sqrt{\frac{2\ln(2/\delta)\,(1 - r/r')}{r}} \ .$$

Proof. Let $f(X_{t-r'+1},\dots,X_t) = \hat{C}_{ij}^{[r]}(t) - \hat{C}_{ij}^{[r']}(t)$, and observe that

$$|f - \mathbb{E}\,f| = \left|C_{ij}^{[r]}(t) - C_{ij}^{[r']}(t) - \hat{C}_{ij}^{[r]}(t) + \hat{C}_{ij}^{[r']}(t)\right| \ .$$

The function f is equivalent to

$$\begin{aligned} f(X_{t-r'+1},\dots,X_t) &= \hat{C}_{ij}^{[r]}(t) - \hat{C}_{ij}^{[r']}(t) \\ &= \sum_{u=t-r+1}^{t} \left(\frac{1}{r} - \frac{1}{r'}\right)\ell_i(X_u)\ell_j(X_u) - \sum_{u=t-r'+1}^{t-r} \frac{1}{r'}\ell_i(X_u)\ell_j(X_u) \ . \end{aligned}$$

Thus, if we change the variable X_u with $t-r+1 \leq u \leq t$, the value of f can change by at most $2\left(\frac{1}{r}-\frac{1}{r'}\right)$, and if we change the variable X_u with $t-r'+1 \leq u \leq t-r$, the value of f can change by at most $2/r'$. We can use McDiarmid's inequality, and obtain that with probability at least $1-\delta$, it holds that:

$$\begin{aligned}
&\left|C_{ij}^{[r]}(t) - C_{ij}^{[r']}(t) - \hat{C}_{ij}^{[r]}(t) + \hat{C}_{ij}^{[r']}(t)\right| \\
&\leq \sqrt{\frac{\ln(2/\delta)}{2}}\sqrt{\sum_{u=t-r+1}^{t} 4\left(\frac{1}{r}-\frac{1}{r'}\right)^2 + \sum_{u=t-r'+1}^{t-r}\frac{4}{r'^2}} \\
&= \sqrt{2\ln(2/\delta)}\sqrt{r\left(\frac{1}{r}-\frac{1}{r'}\right)^2 + (r'-r)\frac{1}{r'^2}} \\
&= \sqrt{2\ln(2/\delta)}\sqrt{\frac{(r'-r)^2 + r(r'-r)}{rr'^2}} \\
&= \sqrt{2\ln(2/\delta)}\sqrt{\frac{r'(r'-r)}{rr'^2}} \\
&= \sqrt{\frac{2\ln(2/\delta)(1-r/r')}{r}}
\end{aligned}$$

□

We end this subsection by providing a result that we quote during the explanation of the algorithm. While this result is not necessary to prove Theorem 5.2, it has a similar flavour than the previous proposition, and we report its proof here.

Proposition B.5. *Let $1 \leq r \leq r' \leq t$. If $D_1 = \ldots = D_t$, then for any pair of indexes $i,j \in \{1,\ldots,\}^n$, it holds*

$$\mathbb{E}\,|\hat{C}_{ij}^{[r]}(t) - \hat{C}_{ij}^{[r']}(t)| \leq \sqrt{\frac{1}{r}-\frac{1}{r'}}\ .$$

Proof. For $i=j$ the statement is trivially true as the difference is 0. Let $i \neq j$. Consider the random variables $Z_k = \ell_i(X_{t-k+1})\ell_j(X_{t-k+1})$ for $1 \leq k \leq r'$. By assumption, the random variables are independent, and $Z_k \in [-1,1]$. By using the definition of $\hat{\boldsymbol{C}}^{[r]}$, we have:

$$\begin{aligned}
\mathbb{E}\,|\hat{C}_{ij}^{[r]}(t) - \hat{C}_{ij}^{[r']}(t)| &= \mathbb{E}\left|\frac{1}{r}\sum_{k=1}^{r} Z_k - \frac{1}{r'}\sum_{k=1}^{r'} Z_k\right| \\
&\leq \sqrt{\mathbb{V}\left(\frac{1}{r}\sum_{k=1}^{r} Z_k - \frac{1}{r'}\sum_{k=1}^{r'} Z_k\right)}\ ,
\end{aligned}$$

where in the last step we used Jensen's inequality. Now, we have that:

$$\begin{aligned}
\mathbb{V}\left(\frac{1}{r}\sum_{k=1}^{r} Z_k - \frac{1}{r'}\sum_{k=1}^{r'} Z_k\right) &= \mathbb{V}\left(\left(\frac{1}{r}-\frac{1}{r'}\right)\sum_{k=1}^{r} Z_k - \frac{1}{r'}\sum_{k=r+1}^{r'} Z_k\right) \\
&= \left[\left(\frac{1}{r}-\frac{1}{r'}\right)^2 r + \frac{1}{r'^2}(r'-r)\right]\mathbb{V}(Z_1) \\
&= \frac{r'-r}{rr'}\mathbb{V}(Z_1)\ .
\end{aligned}$$

Since $Z_1 \in [-1,1]$, by Popoviciu's inequality we have that $\mathbb{V}(Z_1) \leq 1$. Hence, we can conclude that

$$\mathbb{E}\,|\hat{C}_{ij}^{[r]}(t) - \hat{C}_{ij}^{[r']}(t)| \leq \sqrt{\frac{1}{r}-\frac{1}{r'}}\ .$$

□

B.3 Upper Bound to the Drift Error

In this subsection, we show how to provide an upper bound to the drift error term

$$\sum_{i=t-r+1}^{t-1} \|\boldsymbol{C}(i) - \boldsymbol{C}(i+1)\|_\infty$$

as a function of the variation in the weak labelers' accuracies $\boldsymbol{p}(t-r+1), \ldots, \boldsymbol{p}(t)$. Intuitively, the correlation matrix does not change if the weak labelers' accuracies are the same, and a bounded drift in the those accuracies also implies a small variation in the correlation matrix. This is formalized in the following proposition.

Proposition B.6. *For any* $1 \leq k \leq t-1$*, the following inequality holds*

$$\|\boldsymbol{C}(k) - \boldsymbol{C}(k+1)\|_\infty \leq 12\|\boldsymbol{p}(k) - \boldsymbol{p}(k+1)\|_\infty$$

Proof. Consider coordinates i, j such that $i \neq j$. By definition of $C_{i,j}(k)$, we have that

$$C_{i,j}(k) = \mathop{\mathbb{E}}_{X \sim D_k} [\ell_i(X) \cdot \ell_j(X)] \quad .$$

We have that $\ell_i(X) \cdot \ell_j(X)$ is equal to 1 if and only if ℓ_i and ℓ_j are both either correct or incorrect, and is equal to -1 otherwise. By using the definition of $p_i(k)$ and Assumption 2.2, we have that

$$\begin{aligned} C_{i,j}(k) = \mathop{\mathbb{E}}_{X \sim D_k} [\ell_i(X) \cdot \ell_j(X)] &= p_i(k)p_j(k) + (1-p_i(k))(1-p_j(k)) \\ &\quad - p_i(k)(1-p_j(k)) - p_j(k)(1-p_i(k)) \\ &= 4p_i(k)p_j(k) - 2p_i(k) - 2p_j(k) + 1 \quad . \end{aligned}$$

Hence, we have that

$$\begin{aligned} |C_{i,j}(k) - C_{i,j}(k+1)| &= \big|4p_i(k)p_j(k) - 4p_i(k+1)p_j(k+1) \\ &\qquad + 2p_i(k+1) + 2p_j(k+1) - 2p_i(k) - 2p_j(k)\big| \\ &\leq 4\big|p_i(k)p_j(k) - p_i(k+1)p_j(k+1)\big| \\ &\qquad + 2\big|p_i(k+1) - p_i(k)\big| + 2\big|p_j(k+1) - p_j(k)\big| \quad . \end{aligned} \tag{10}$$

where the first inequality follows from the triangle inequality. For ease of notation, let $\rho_k = \|\boldsymbol{p}(k+1) - \boldsymbol{p}(k)\|_\infty$. We have that

$$\begin{aligned} p_i(k)p_j(k) &= p_i(k)(p_j(k) + p_j(k+1) - p_j(k+1)) \\ &\leq p_i(k)p_j(k+1) + p_i(k)|p_j(k) - p_j(k+1)| \\ &\leq p_i(k)p_j(k+1) + \rho_k \\ &= (p_i(k) - p_i(k+1) + p_i(k+1))p_j(k+1) + \rho_k \\ &\leq p_i(k+1)p_j(k+1) + 2\rho_k \quad . \end{aligned}$$

which implies that $p_i(k)p_j(k) - p_i(k+1)p_j(k+1) \leq 2\rho_k$. Similarly, we can show that $p_i(k+1)p_j(k+1) - p_i(k)p_j(k) \leq 2\rho_k$, hence we have that $|p_i(k+1)p_j(k+1) - p_i(k)p_j(k)| \leq 2\rho_k$. By using this inequality in (10), we obtain $|C_{i,j}(k) - C_{i,j}(k+1)| \leq 12\rho_k$. The statement follows by substituting the definition of ρ_k. □

B.4 Proof of Lemma 5.1

Proof. With Proposition B.2, we have that:

$$\left\|\boldsymbol{C}(t) - \hat{\boldsymbol{C}}^{[r]}(t)\right\|_\infty \leq \left\|\boldsymbol{C}^{[r]}(t) - \hat{\boldsymbol{C}}^{[r]}(t)\right\|_\infty + \sum_{i=t-r+1}^{t-1} \|\boldsymbol{C}(i) - \boldsymbol{C}(i+1)\|_\infty \quad .$$

In order to conclude the result, we upper bound each term of the right-hand side of the above inequality individually. We use Proposition B.3 and take an union bound over $n(n-1)/2$ coordinates (see also (9)), and we obtain that with probability at least $1-\delta$, it holds

$$\left\| \boldsymbol{C}^{[r]}(t) - \hat{\boldsymbol{C}}^{[r]}(t) \right\|_\infty \leq \sqrt{\frac{2\ln(n(n-1)/\delta)}{r}}$$

Proposition B.6 yields the following upper bound:

$$\sum_{i=t-r+1}^{t-1} \| \boldsymbol{C}(i) - \boldsymbol{C}(i+1) \|_\infty \leq 12 \sum_{i=t-r+1}^{t-1} \| \boldsymbol{p}(i) - \boldsymbol{p}(i+1) \|_\infty$$

□

B.5 Dynamic Selection of the Window Size (Theorem B.1)

In this subsection, we show how to adaptively choose the number of past samples that minimizes a trade-off between the statistical error and the drift error. As a prerequisite, our algorithm requires that the used empirical quantities provide a good approximation of their estimated expectations. If this is not the case, we simply assume that our algorithm fails, and this happens with probability $\leq \delta$. The next corollary formalizes this required guarantee on the estimation. We remind the definition of the value $A_{\delta,n,m}$ in the statement of Theorem B.1:

$$A_{\delta,n,m} = \sqrt{2\ln[(2m-1)\cdot n(n-1)/\delta]}$$

Corollary B.7. *Let $\delta > 0$ Let $\mathcal{R} = \{r_1, \dots, r_m\}$. With probability at least $1-\delta$, it holds:*

$$\| \boldsymbol{C}(t) - \hat{\boldsymbol{C}}^{[r_k]}(t) \|_\infty \leq \frac{A_{\delta,n,m}}{\sqrt{r_k}} + \left\| \boldsymbol{C}(t) - \boldsymbol{C}^{[r]}(t) \right\|_\infty \qquad \forall k \leq m$$

$$\| \boldsymbol{C}^{[r_k]}(t) - \boldsymbol{C}^{[r_{k+1}]}(t) - \hat{\boldsymbol{C}}^{[r_k]}(t) + \hat{\boldsymbol{C}}^{[r_{k+1}]}(t) \|_\infty \leq A_{\delta,n,m} \sqrt{\frac{1 - r_k/r_{k+1}}{r_k}} \qquad \forall k \leq m-1$$

Proof. By using the triangle inequality, we have that

$$\| \boldsymbol{C}(t) - \hat{\boldsymbol{C}}^{[r_k]}(t) \|_\infty \leq \left\| \boldsymbol{C}^{[r_k]}(t) - \hat{\boldsymbol{C}}^{[r_k]}(t) \right\|_\infty + \left\| \boldsymbol{C}(t) - \boldsymbol{C}^{[r]}(t) \right\|_\infty$$

We use Proposition B.3 and Proposition B.4 to upper bound respectively $\| \boldsymbol{C}(t) - \hat{\boldsymbol{C}}^{[r_k]}(t) \|_\infty$ and $\| \boldsymbol{C}^{[r_k]}(t) - \boldsymbol{C}^{[r_{k+1}]}(t) - \hat{\boldsymbol{C}}^{[r_k]}(t) + \hat{\boldsymbol{C}}^{[r_{k+1}]}(t) \|_\infty$. Those propositions provide a guarantee for a single choice of window sizes and coordinates: we take an union bound over $n(n-1)/2$ choice of coordinates and $m + (m-1)$ different choice of window sizes, hence we take an union bound over $(2m-1)n(n-1)/2$ events. The statement immediately follows by an inspection of the value $A_{\delta,n,m}$. □

Throughout this subsection, we assume that the event of Corollary B.7 holds, otherwise our algorithm fails (with probability $\leq \delta$). Let β, γ_m and γ_M be defined as in Theorem B.1. We define the following function:

$$\mathcal{B}(r) = \frac{A_{\delta,n,m}}{\sqrt{r}} \frac{2\beta + 2}{1 - \gamma_M} + \left\| \boldsymbol{C}(t) - \boldsymbol{C}^{[r]}(t) \right\|_\infty$$

The value $\mathcal{B}(r)$ is the upper bound that our algorithm guarantees to $\|C(t) - \hat{C}^{[r]}(t)\|_\infty$ using any value $r \in \mathcal{R}$. In fact, we have that

$$\begin{aligned}
\mathcal{B}(r) &= \frac{A_{\delta,n,m}}{\sqrt{r}} \frac{2\beta + 2}{1 - \gamma_M} + \left\| \boldsymbol{C}(t) - \boldsymbol{C}^{[r]}(t) \right\|_\infty \\
&\geq \frac{A_{\delta,n,m}}{\sqrt{r}} + \left\| \boldsymbol{C}(t) - \boldsymbol{C}^{[r]}(t) \right\|_\infty \\
&\geq \| \boldsymbol{C}(t) - \hat{\boldsymbol{C}}^{[r]}(t) \|_\infty \qquad \forall r \in \mathcal{R} \ ,
\end{aligned}$$

where the last inequality follows from Corollary B.7. For any value $k \leq m-1$, also let

$$\mathcal{T}(k) \doteq 2\beta A_{\delta,n,m}\sqrt{\frac{1}{r_k}} + A_{\delta,n,m}\sqrt{\frac{1-r_k/r_{k+1}}{r_k}} \ ,$$

and observe that this is the quantity used as a threshold in Line 5 of the algorithm at iteration k.

The proof of Theorem B.1 revolves around the following two Propositions B.8 and B.9.

1. We guarantee that if $\|\hat{\boldsymbol{C}}^{[r_{k+1}]}(t) - \hat{\boldsymbol{C}}^{[r_k]}(t)\|_\infty$ is smaller than the threshold $\mathcal{T}(k)$, then a negligeable drift occured, and the upper bound $\mathcal{B}(r_{k+1})$ is smaller than $\mathcal{B}(r_k)$ (Proposition B.8) In this case, we can keep iterating.
2. On the other hand, if $\|\hat{\boldsymbol{C}}^{[r_{k+1}]}(t) - \hat{\boldsymbol{C}}^{[r_k]}(t)\|_\infty$ is greater than the threshold $\mathcal{T}(k)$, a sizeable drift occurred, and we can provide a lower bound on the drift error (Proposition B.9). In this case, we can stop iterating and return the current window size r_k.

We prove those two propositions.

Proposition B.8. *Let the event of Corollary B.7 hold. Then, for any* $1 \leq k \leq m-1$

$$\|\hat{\boldsymbol{C}}^{[r_k]} - \hat{\boldsymbol{C}}^{[r_{k+1}]}\|_\infty \leq \mathcal{T}(k) \Longrightarrow \mathcal{B}(r_{k+1}) \leq \mathcal{B}(r_k)$$

Proof. We have that

$$\mathcal{B}(r_{k+1}) - \mathcal{B}(r_k) = A_{\delta,n,m}\frac{2\beta+2}{1-\gamma_M}\left[\sqrt{\frac{1}{r_{k+1}}} - \sqrt{\frac{1}{r_k}}\right] + \|\boldsymbol{C}(t) - \boldsymbol{C}^{[r_{k+1}]}(t)\|_\infty - \|\boldsymbol{C}(t) - \boldsymbol{C}^{[r_k]}(t)\|_\infty \tag{11}$$

We can obtain the following upper

$$\begin{aligned}
& \|\boldsymbol{C}(t) - \boldsymbol{C}^{[r_{k+1}]}(t)\|_\infty - \|\boldsymbol{C}(t) - \boldsymbol{C}^{[r_k]}(t)\|_\infty \\
\leq\ & \|\boldsymbol{C}^{[r_{k+1}]}(t) - \boldsymbol{C}^{[r_k]}(t)\|_\infty \\
=\ & \|\hat{\boldsymbol{C}}^{[r_k]}(t) - \hat{\boldsymbol{C}}^{[r_{k+1}]}(t) - \hat{\boldsymbol{C}}^{[r_k]}(t) + \hat{\boldsymbol{C}}^{[r_{k+1}]}(t) + \boldsymbol{C}^{[r_k]}(t) - \boldsymbol{C}^{[r_{k+1}]}(t)\|_\infty \\
\leq\ & \|\hat{\boldsymbol{C}}^{[r_k]}(t) - \hat{\boldsymbol{C}}^{[r_{k+1}]}(t)\|_\infty + \|-\hat{\boldsymbol{C}}^{[r_k]}(t) + \hat{\boldsymbol{C}}^{[r_{k+1}]}(t) + \boldsymbol{C}^{[r_k]}(t) - \boldsymbol{C}^{[r_{k+1}]}(t)\|_\infty \\
\leq\ & \mathcal{T}(k) + A_{\delta,n,m}\sqrt{\frac{1-r_k/r_{k+1}}{r_k}}
\end{aligned}$$

where the first two inequalities are due to the triangle inequality, and the last inequality is due to the assumption of the proposition statement and Corollary B.7. By plugging the above inequality in (11) and using the definition of $\mathcal{T}(k)$, we obtain

$$\mathcal{B}(r_{k+1}) - \mathcal{B}(r_k) \leq \frac{A_{\delta,n,m}}{\sqrt{r_k}}\left[\frac{2\beta+2}{1-\gamma_M}\sqrt{\frac{r_k}{r_{k+1}}} - \frac{2\beta+2}{1-\gamma_M} + 2\beta + 2\sqrt{1-r_k/r_{k+1}}\right] \tag{12}$$

We have that

$$\begin{aligned}
& \left[\frac{2\beta+2}{1-\gamma_M}\sqrt{\frac{r_k}{r_{k+1}}} - \frac{2\beta+2}{1-\gamma_M} + 2\beta + 2\sqrt{1-r_k/r_{k+1}}\right] \\
\leq\ & \left[\frac{2\beta+2}{1-\gamma_M}\gamma_M - \frac{2\beta+2}{1-\gamma_M} + 2\beta + 2\right] \\
=\ & 0
\end{aligned}$$

By using this inequality in (12), we finally obtain that $\mathcal{B}(r_{k+1}) - \mathcal{B}(r_k) \leq 0$. □

This proposition guarantees that every time the If of Line 5 of the algorithm is true, then $\mathcal{B}(r_{k+1})$ is an upper bound at least as good as $\mathcal{B}(r_k)$. Conversely, the next proposition shows that when the If of Line 5 is false, a drift must have occurred.

Proposition B.9. *Let the event of Corollary B.7 hold. Then, for any* $1 \leq k \leq m-1$

$$\left\|\hat{\boldsymbol{C}}^{[r_k]} - \hat{\boldsymbol{C}}^{[r_{k+1}]}\right\|_\infty > \mathcal{T}(k) \Longrightarrow \sum_{u=t-r_{k+1}+1}^{t-1} \|\boldsymbol{C}(u) - \boldsymbol{C}(u+1)\|_\infty > A_{\delta,n,m} \cdot \beta/\sqrt{r_k}$$

Proof. We have that:

$$\begin{aligned}\|\hat{\boldsymbol{C}}^{[r_k]} - \hat{\boldsymbol{C}}^{[r_{k+1}]}\|_\infty &\leq A_{\delta,n,m}\sqrt{\frac{1-r_k/r_{k+1}}{r_k}} + \|\boldsymbol{C}^{[r_k]}(t) - \boldsymbol{C}^{[r_{k+1}]}(t)\|_\infty \\ &\leq A_{\delta,n,m}\sqrt{\frac{1-r_k/r_{k+1}}{r_k}} + \|C(t) - C^{[r_{k+1}]}(t)\|_\infty + \|C^{[r_k]}(t) - C(t)\|_\infty\end{aligned}$$

where the first inequality is due to Corollary B.7, and the second inequality is due to the triangle inequality. By using (6), we can show that

$$\|\boldsymbol{C}(t) - \boldsymbol{C}^{[r_{k+1}]}(t)\|_\infty + \|\boldsymbol{C}^{[r_k]}(t) - \boldsymbol{C}(t)\|_\infty \leq 2\sum_{u=t-r_{k+1}+1}^{t-1} \|\boldsymbol{C}(u) - \boldsymbol{C}(u+1)\|_\infty \ .$$

Hence, by using the assumption of the proposition and the definition of $\mathcal{T}(k)$, we obtain the following inequality:

$$2\sum_{u=t-r_{k+1}+1}^{t-1} \|\boldsymbol{C}(u) - \boldsymbol{C}(u+1)\|_\infty > 2\beta A_{\delta,n,m}/\sqrt{r_k} \ ,$$

and the statement immediately follows. □

In the following Lemma, we use Proposition B.8 and B.9 to show that the matrix $\hat{\boldsymbol{C}}$ of Line 7 of the algorithm provides a good approximation of $\boldsymbol{C}(t)$. Theorem B.1 immediately follows from this result by using Proposition 4.2.

Lemma B.10. *Consider the setting of Theorem B.1. Let* $\hat{C}$ *be the matrix defined at Line 7 of the algorithm. With probability at least* $1-\delta$*, it holds that*

$$\|\hat{\boldsymbol{C}} - \boldsymbol{C}(t)\|_\infty \leq \Phi_{\mathcal{R},\beta} \min_{r\in[r_1,\min(t,r_m)]}\left(\frac{A_{\delta,n,m}}{\sqrt{r}} + \sum_{u=t-r+1}^{t-1}\|C(u) - C(u+1)\|_\infty\right)$$

Proof. Assume that the event of Corollary B.7 holds (otherwise we say that our algorithm fails, with probability $\leq \delta$). For ease of notation, let $\nu = (2\beta+2)/(1-\gamma_M)$. Let $\hat{k} \leq m$ be the value such that $\hat{\boldsymbol{C}} = \hat{\boldsymbol{C}}^{[r_{\hat{k}}]}(t)$. We remind that the algorithm guarantees an upper bound $\mathcal{B}(r_{\hat{k}})$ to the estimation error $\|\hat{\boldsymbol{C}} - \boldsymbol{C}(t)\|_\infty$ Let r^* be the integer that minimizes

$$r^* = \text{argmin}_{r\in[r_1,\min(t,r_m)]}\left(\frac{A_{\delta,n,m}}{\sqrt{r}} + \sum_{u=t-r+1}^{t-1}\|C(u) - C(u+1)\|_\infty\right),$$

and let $\mathcal{B}^*$ be the minimum value of the above expression, i.e.

$$\mathcal{B}^* = \frac{A_{\delta,n,m}}{\sqrt{r^*}} + \sum_{u=t-r^*+1}^{t-1}\|C(u) - C(u+1)\|_\infty$$

In order to prove the lemma, it is sufficient to show that $\mathcal{B}(r_{\hat{k}})/\mathcal{B}^* \leq \Phi_{\mathcal{R},\beta}$.

We distinguish two cases: (a) $\hat{k} = m$ or $r^* < r_{\hat{k}+1}$ and (b) $r^* \geq r_{\hat{k}+1}$. Consider case (a). Let $\tilde{k}$ be the largest integer such that $r_{\tilde{k}} \leq r^*$. By construction, we can observe that $\tilde{k} \leq \hat{k}$. Since the algorithm did not interrupt in the iterations $1,\dots,\tilde{k},\dots,\hat{k}$, we can use Proposition B.8, to show that $\mathcal{B}(r_{\hat{k}}) \le \mathcal{B}(r_{\tilde{k}})$. Hence, we have that:

$$\begin{aligned}
\frac{\mathcal{B}(r_{\hat{k}})}{\mathcal{B}^*} \le \frac{\mathcal{B}(r_{\tilde{k}})}{\mathcal{B}^*} &= \frac{\nu \cdot A_{\delta,n,m}/\sqrt{r_{\tilde{k}}} + \left\|\boldsymbol{C}(t) - \boldsymbol{C}^{[r_{\tilde{k}}]}(t)\right\|_\infty}{A_{\delta,n,m}/\sqrt{r^*} + \sum_{u=t-r^*+1}^{t-1} \|\boldsymbol{C}(u) - \boldsymbol{C}(u+1)\|_\infty} \\
&\le \frac{\nu \cdot A_{\delta,n,m}/\sqrt{r_{\tilde{k}}} + \sum_{u=t-r_{\tilde{k}}+1}^{t-1} \|\boldsymbol{C}(u) - \boldsymbol{C}(u+1)\|_\infty}{A_{\delta,n,m}/\sqrt{r^*} + \sum_{u=t-r^*+1}^{t-1} \|\boldsymbol{C}(u) - \boldsymbol{C}(u+1)\|_\infty} \\
&\le \nu\sqrt{\frac{r^*}{r_{\tilde{k}}}} + \frac{\sum_{u=t-r_{\tilde{k}}+1}^{t-1} \|\boldsymbol{C}(u) - \boldsymbol{C}(u+1)\|_\infty}{\sum_{u=t-r^*+1}^{t-1} \|\boldsymbol{C}(u) - \boldsymbol{C}(u+1)\|_\infty} \\
&\le \nu/\gamma_m + 1
\end{aligned}$$

where the second inequality is due to (6), and the last inequality is due to the definition of $r_{\tilde{k}}$. We can observe that $1 + \nu/\gamma_m = 1 + \frac{2\beta+2}{\gamma_m(1-\gamma_M)} \le \Phi_{\mathcal{R},\beta}$ and this concludes the first part of the proof.

We consider case (b). Since the algorithm stopped at iteration $\hat{k} < m$, the If condition of Line 4 is false during this iteration, and due to Proposition B.9, we have that:

$$\sum_{u=t-r^*+1}^{t-1} \|\boldsymbol{C}(u) - \boldsymbol{C}(u+1)\|_\infty \ge \sum_{u=t-r_{\hat{k}+1}+1}^{t-1} \|\boldsymbol{C}(u) - \boldsymbol{C}(u+1)\|_\infty \ge A_{\delta,n,m}\beta/\sqrt{r_{\hat{k}}} \ . \tag{13}$$

We obtain:

$$\begin{aligned}
\frac{\mathcal{B}(r_{\hat{k}})}{\mathcal{B}^*} &= \frac{\nu \cdot A_{\delta,n,m}/\sqrt{r_{\hat{k}}} + \left\|\boldsymbol{C}(t) - \boldsymbol{C}^{[r_{\hat{k}}]}(t)\right\|_\infty}{A_{\delta,n,m}/\sqrt{r^*} + \sum_{u=t-r^*+1}^{t-1} \|\boldsymbol{C}(u) - \boldsymbol{C}(u+1)\|_\infty} \\
&\le \frac{\nu \cdot A_{\delta,n,m}/\sqrt{r_{\hat{k}}} + \sum_{u=t-r_{\hat{k}}+1}^{t-1} \|\boldsymbol{C}(u) - \boldsymbol{C}(u+1)\|_\infty}{A_{\delta,n,m}/\sqrt{r^*} + \sum_{u=t-r^*+1}^{t-1} \|\boldsymbol{C}(u) - \boldsymbol{C}(u+1)\|_\infty} \\
&\le \frac{\nu \cdot A_{\delta,n,m}/\sqrt{r_{\hat{k}}}}{\sum_{u=t-r^*+1}^{t-1} \|\boldsymbol{C}(u) - \boldsymbol{C}(u+1)\|_\infty} + \frac{\sum_{u=t-r_{\hat{k}}+1}^{t-1} \|\boldsymbol{C}(u) - \boldsymbol{C}(u+1)\|_\infty}{\sum_{u=t-r^*+1}^{t-1} \|\boldsymbol{C}(u) - \boldsymbol{C}(u+1)\|_\infty} \\
&\le \frac{\nu}{\beta} + 1 \ .
\end{aligned}$$

where in the last inequality we used (13) and the fact that $r_{\hat{k}} \le r^*$. We finally observe that $\nu/\beta + 1 = 1 + \frac{2\beta+2}{(1-\gamma_M)\beta} \le \Phi_{\mathcal{R},\beta}$, and this concludes the proof. □

We can finally prove Theorem B.1 as a simple corollary of Lemma B.10.

Proof of Theorem B.1. Let ϵ be the guarantee of Lemma B.10. If we let $\hat{C}$ be the matrix of Line 7 of the algorithm, we have that with probability at least $1-\delta$, it holds:

$$\left\|\hat{\boldsymbol{C}} - \boldsymbol{C}(t)\right\|_\infty \le \epsilon$$

Lines 8-12 of the algorithm implement the procedure of Bonald and Combes (2017) that attains the guarantee of Proposition 4.2. Hence, we have that:

$$\begin{aligned}
\|\hat{\boldsymbol{p}} - \boldsymbol{p}(t)\|_\infty &\le \frac{5\epsilon}{2\tau^2} \\
&\le \frac{5\Phi_{\mathcal{R},\beta}}{2\tau^2} \min_{r\in[r_1,\min(t,r_m)]} \left(\frac{A_{\delta,n,m}}{\sqrt{r}} + \sum_{u=t-r+1}^{t-1} \|\boldsymbol{C}(u) - \boldsymbol{C}(u+1)\|_\infty \right) \ .
\end{aligned}$$

The statement immediately follows by using Proposition B.6. □

C Relaxing the Conditional Independence Assumption

The analysis of our algorithm uses the conditional independence assumption for the error of the weak labelers (Assumption 2.2). In this section, we explore a possible relaxation of this assumption based on previous work that handles the general case where the weak labelers are arbitrarily correlated Mazzetto et al. (2021b). We propose a variant of our method that can be adopted in this setting.

At the time step t, consider the random vector

$$\boldsymbol{a}(t) = \left(\mathbf{1}_{\ell_1(x)=y(x)}, \ldots, \mathbf{1}_{\ell_n(x)=y(x)}\right) \qquad \text{where } x \sim D_t$$

Indeed, we have that $\mathbb{E}\,\boldsymbol{a}(t) = \boldsymbol{p}(t)$. The conditional independence assumption allows us to factorize the distribution of $\boldsymbol{a}(t)$, i.e., for any $\boldsymbol{a} \in \{0,1\}^n$, we have that:

$$\Pr(\boldsymbol{a}(t) = \boldsymbol{a}) = \prod_{i=1}^{n} \Pr(\boldsymbol{a}(t)_i = \boldsymbol{a}_i) \ . \tag{14}$$

The factorization (14) is the property that is exploited by the methods that are based on this conditional independence assumption, and it is also used to define the optimal aggregation rule given by (2).

If we relax Assumption 2.2, Equation (14) no longer holds, and in general, the optimal aggregation rule depends on the dependencies between the weak labelers' errors. In particular, the previous work Mazzetto et al. (2021b) shows that in the worst-case it is not possible to improve upon the most accurate weak labeler, without any further assumption on the errors' dependencies (a clear example is when $\ell_1 = \ldots = \ell_n$).

The previous work Mazzetto et al. (2021b) provides a method to determine a subset of weak labelers whose majority vote has provably bounded error, without introducing any assumption on the dependencies between the weak labelers' error. The idea is to leverage the information about the correlation to determine a subset of weak labelers who make errors in different parts of the distribution domain so that their majority vote can increase their aggregate performance. This method relies on the following quantities: (1) the correlation between the weak labelers outputs $\boldsymbol{C}(t)$, which can be estimated only using unlabeled data; and (2) the weak labelers' accuracies $\boldsymbol{p}(t)$. As we will see, we can use our adaptive method to maintain an estimate of the correlation matrix in a non-stationary setting.

With access to $\boldsymbol{p}(t)$ and $\boldsymbol{C}(t)$, the worst-case error of the majority vote for the weak labelers $\{\ell_1, \ldots, \ell_n\}$ at time t can be computed as a solution of the following linear program:

$$
\begin{aligned}
(*) \qquad & \max \sum_{\boldsymbol{a} \in \{0,1\}^n : \|\boldsymbol{a}\|_1 < n/2} p_{\boldsymbol{a}} & \\
& (a) \quad \sum_{\boldsymbol{a} \in \{0,1\}^n : a_i = 1} p_{\boldsymbol{a}} = \boldsymbol{p}(t)_i & \text{for } i = 1, \ldots, n \\
& (b) \quad \sum_{\boldsymbol{a} \in \{0,1\}^n : a_i = a_j} p_{\boldsymbol{a}} = (\boldsymbol{C}(t)_{ij} + 1)/2 & \text{for } i \neq j \\
& (c) \quad \sum_{\boldsymbol{a}} p_{\boldsymbol{a}} = 1 & \\
& (d) \quad 0 \leq p_{\boldsymbol{a}} \leq 1 & \forall \boldsymbol{a}
\end{aligned}
$$

As described in Mazzetto et al. (2021b), the optimization problem (*) can be used as a subroutine to find a subset of weak labelers which has provably a small worst-case error for their majority vote. However, the optimization problem (*) requires access to the unknown quantities $\boldsymbol{p}(t)$ and $\boldsymbol{C}(t)$.

In many practical applications, it is realistic to assume that the weak labelers provide a significantly stronger signal than noise, i.e. Assumption 4.1 holds for a value of $\tau \in (0, 1/2)$ that is far from zero. Then it is possible to replace constraint (a) with $\sum_{\boldsymbol{a} \in \{0,1\}^n : a_i = 1} p_{\boldsymbol{a}} \geq \frac{1}{2} + \tau$. In particular, for each weak labeler ℓ_i, we can use a different value τ_i to reflect our believe on how much this weak labeler is accurate Arachie and Huang (2019, 2021).

We leverage our adaptive method to keep track of the matrix $\boldsymbol{C}(t)$. Even in a non-stationary setting, where we only have access to a single sample from P_t, we can estimate $\boldsymbol{C}(t)$ from the samples $X_1, \ldots, X_t$ using Algorithm 1.

In fact, in our analysis we show that the matrix $\hat{\boldsymbol{C}}$ computed in Line 7 of our algorithm achieves a near-optimal trade-off between statistical error and drift error (Lemma B.10), providing an empirical estimate of $\boldsymbol{C}(t)$ in a drift setting.

Thus, we can obtain an algorithm for the case where the weak labelers' errors are not conditionally independent, and they can be arbitrarily correlated. At time t, we use Algorithm 1 to estimate $\boldsymbol{C}(t)$ with $\hat{\boldsymbol{C}}$. Then, we use the information on the correlation matrix to identify a subset of weak labelers whose majority vote is accurate. In particular, we use the algorithm described in Mazzetto et al. (2021b) which is based on solving multiple instances of the optimization problem $(*)$.